TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

Ying Chen

Advances on Coding and Transmission of Scalable Video and Multiview Video

Julkaisu 871 • Publication 871

Tampere 2010

Tampereen teknillinen yliopisto. Julkaisu 871
Tampere University of Technology. Publication 871

Ying Chen

Advances on Coding and Transmission of Scalable Video and Multiview Video

Thesis for the degree of Doctor of Technology to be presented with due permission for public examination and criticism in Tietotalo Building, Auditorium TB109, at Tampere University of Technology, on the 11th of February 2010, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2010

ISBN 978-952-15-2323-6 (printed)
ISBN 978-952-15-2339-7 (PDF)
ISSN 1459-2045

To my mother: Ouling Du (1947~2007)

Abstract

The Advanced Video Coding (H.264/AVC) is the state-of-art video coding standard which has been developed by the Joint Video Team of ISO/IEC MPEG and ITU-T VCEG. It has been widely adopted in numerous products and services, such as TV broadcasting, video conferencing, mobile TV, and Blue-ray Disc. However, to support other application scenarios, for example, video delivery over heterogeneous networks, and enhanced user experiences in different aspects, advanced video representations of a scene are desired. From 2004 to 2008, JVT has developed video coding standards as the extensions of H.264/AVC : the scalable extension and the multiview extension, namely Scalable Video Coding (SVC) and Multiview Video Coding (MVC), respectively.

SVC is designed to provide adaptation capability for heterogeneous network structures and different receiving devices with the help of temporal, spatial, and quality scalabilities. In addition, other potential scalabilities are investigated during the development of SVC. When a video sequence coded with SVC is delivered over an error-prone environment, it is challenging to achieve graceful quality degradation. Therefore, error resilient coding and error concealment techniques have been introduced for SVC. Some of the techniques are inherited from those for H.264/AVC, whereas some take advantages of the SVC features.

The large amount of data needed to be processed by multiview applications is a heavy burden for both transmission and decoding. The MVC standard includes a number of new techniques for improved coding efficiency, reduced decoding complexity, and new functionalities. The system level integration of MVC is conceptually more challenging as the output of the decoder may contain any combination of the views with any temporal resolution level. Multiview video only enables rendering of a limited number of views. To achieve 3D rendering ability at any view angle or position, depth maps can be coded with texture video sequences.

In this thesis, techniques for scalable and multiview video coding applications are proposed in the end to end systems based on SVC or MVC, including the design of standards, coding tools, encoder algorithms for high efficiency or improved error resilience, and decoder side error concealment.

The contributions of this thesis are presented in Chapter 1 to Chapter 5. Each chapter includes a summary of a number of published papers by the author. These papers are

attached to the thesis. Chapter 1 introduces H.264/AVC, focusing on the terminologies and the main techniques for SVC and MVC. An H.264/AVC baseline compliant high efficient encoding algorithm, which was based on hierarchical inter P picture structure, is proposed here. Chapter 2 and Chapter 3 give overviews for SVC and MVC, respectively. In Chapter 2, a color bit-depth scalable coding algorithm is proposed. It is a coding technology targeting future enhancement for SVC, and enabling storage or carriage of typical eight-bit video and high-bit video simultaneously with less bandwidth consumption. In Chapter 3, the presented techniques cover a wide range of MVC design, e.g., MVC bitstream structure, MVC transport, and MVC decoder resource management. The technologies contributed by the author, have been part of the MVC standard. Error concealment and error resilient methods for SVC and MVC, for an improved reconstructed video quality in an error-prone environment, are proposed in Chapter 4. In Chapter 5, the following coding techniques in specific multiview video coding applications are proposed: single-loop decoding for MVC, advanced asymmetric stereoscopic coding, and coding of 3D video content with depth maps.

The above approaches summarize the recent advances in coding and transmission of SVC and MVC, contributed by the author. With the deployment of the SVC and MVC standards, the proposed techniques are expected to be widely used by industry, in this field, and are becoming important references for other relevant academic research.

Acknowledgments

The research presented in this thesis has been carried out at the Department of Signal Processing, Tampere University of Technology (TUT), Finland, in collaboration with Nokia Research Center, Tampere, during 2006-2009.

This thesis owes its existence to the help, support, and inspiration of many people. In the first place, I would like to express my sincere appreciation and gratitude to Prof. Moncef Gabbouj for his supervision, advice, and support from the very early stage of this research as well as the careful review of the thesis. Prof. Gabbouj has created a wonderful environment which enables me to focus on my research. Without this environment, it is impossible for me to accumulate industry experience as well as achieve academic progress.

I am indebted to Dr. Huifang Sun, for his continuous encouragement for my research career over many years and his precious time to serve as a pre-examiner of this thesis.

The discussions and cooperation with my colleagues in Nokia Research Center (NRC) have contributed substantially to this work. Many thanks go in particular to Dr. Ye-Kui Wang and Dr. Miska M. Hannuksela, both of whom were my supervisors in the collaborated project between NRC and TUT. I am graceful to Miska and Ye-Kui for their detailed instructions in the video standardization work as well as their valuable advices in publications. I have especially benefited by the guidance and friendship of Ye-Kui who generously granted me his time and efforts for my study, career, and life.

Moreover, it is a wonderful experience to collaborate with University of Science and Technology of China, specifically with Prof. Houqiang Li's group, which has resulted in numerous standard contributions and publications with Prof. Li and his students: Yi Guo, Hui Liu, Shujie Liu, Siping Tao, Weixing Wan, Maosheng Ji, and Ling Zhu.

In addition, I owe my thanks to my colleagues in TUT, especially to Dr. Mehdi Rezaei, Jin Li, Chenghao Liu, Vinod Malamal Vadakital, for helping me get used to the research/education facilities and bringing encouragement and joy to color my life in Finland.

I would like to thank Virve Larmila, Ulla Siltaloppi and Elina Orava, for their great help for some routine but important administration work.

Actually, my research in video coding and transmission began in 2004 when I was working in Thomson Corporate Research. I am very grateful to my colleagues in Thomson, for their collaboration, instruction and friendship. Many thanks go in particular to Dr. Peng Yin, Kai Xie, Purvin Pandit, Dr. Eduard Francois and Jill Boyce.

I also extend my appreciation to Joint Video Team (JVT) and the experts who have reviewed my technical contributions. More specifically, I am indebted to Dr. Gary Sullivan, Prof. Thomas Wiegand and Prof. Jens-Rainer Ohm. In addition, I am very grateful to Dr. Anthony Verto, and Dr. Aljoscha Smolic, for the stimulating technical discussions and

fruitful collaborations on Multiview Video Coding and 3D Video work in JVT and MPEG. Dr. Smolic has also kindly served as a pre-examiner of this thesis.

I convey special acknowledgement to Peking University, from which I got solid academic training, during my undergraduate and master studies, which has benefited me for life long. My special thanks go to Prof. Pengwei Hao, my Master supervisor, who guided me patiently on my early research activities and publications.

Financial support of Nokia Foundation is gratefully acknowledged.

I owe special gratitude to my family, for continuous and unconditional support of all my undertakings, scholastic and otherwise. Words fail me to express my appreciation to my wife Jing Xu, whose dedication, love and persistent confidence in me, have taken the load off my shoulder, during the past seven years, when I have been living in four different countries. I owe her for unselfishly letting her intelligence, passions, and ambitions collide with mine. I am indebted to my mother, Ouling Du, for her care and love. Her unselfish dedication to the family made it possible for me to enter the top university in China, from a normal family in a small town in China. She had never complained in spite of all the hardships in her life. I could only wish that she was able to witness my completion of the PhD study.

Finally, I would finally wish to thank everyone who contributed to the successful completion of the thesis.

Contents

List of Publications

The thesis is composed of a summary part and 11 publications listed below as appendices. The publications are referred in the thesis as [P1], [P2], etc.

[P1] W. Wan, Y. Chen, Y.-K. Wang, M. M. Hannuksela, H. Li, and M. Gabbouj, "Efficient Hierarchical Inter Picture Coding for H.264/AVC Baseline Profile," *Picture Coding Symposium, PCS*'09, Chicago, Illinois, USA, May 6-8, 2009.

[P2] Y. Gao, Y. Wu, and Y. Chen, "H.264/Advanced Video Coding (AVC) Backward-Compatible Bit-Depth Scalable Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 4, pp. 500–510, April 2009.

[P3] Y. Chen, Y. -K. Wang, K. Ugur, M. M. Hannuksela, J. Lainema, and M. Gabbouj, "The Emerging MVC Standard for 3D Video Services," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, Article ID 786015.

[P4] Y. Guo, Y. Chen, Y.-K. Wang, H. Li, M. M. Hannuksela, and M. Gabbouj, "Error Resilient Coding and Error Concealment in Scalable Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 6, pp. 781–795, June 2009.

[P5] S. Liu, Y. Chen, Y. -K. Wang,. M. Gabbouj, M.M. Hannuksela, and H. Li, "Frame Loss Error Concealment for Multiview Video Coding," *IEEE International Symposium on Circuits and Systems, ISCAS'*08, Seattle, Washington, USA, May 18-21, 2008, pp. 3470–3473.

[P6] Y. Chen, M. M. Hannuksela, L. Zhu, A. Hallapuro, M. Gabbouj and H. Li, "Coding Techniques in Multiview Video Coding and Joint Multiview video Model," *Picture Coding Symposium, PCS*'09, Chicago, Illinois, USA, May 6-8, 2009.

[P7] Y. Chen, Y.-K. Wang, M. M. Hannuksela, and M. Gabbouj, "Single-Loop Decoding for Multiview Video Coding," *IEEE International Conference on Multimedia and Expo, ICME*'08, Hannover, Germany, June 23-26, 2008, pp. 605–608.

[P8] Y. Chen, S. Liu, Y.-K. Wang, M. M. Hannuksela, H. Li, and M. Gabbouj, "Low-complexity Asymmetric Multiview Video Coding," *IEEE International Conference*

on Multimedia and Expo, ICME'08, Hannover, Germany, June 23-26, 2008, pp. 773–776.

[P9] Y. Chen, Y.-K. Wang, M. M. Hannuksela, and M. Gabbouj, "Picture-level Adaptive Filter for Asymmetric Stereoscopic Video," *IEEE International Conference on Image Processing, ICIP*'08, San Diego, CA, USA, October 12-25, 2008, pp. 1944–1947.

[P10] Y. Chen, Y.-K. Wang, M. M. Hannuksela, and M. Gabbouj, "Regionally Adaptive Filtering for Asymmetric Stereoscopic Video Coding," *IEEE International Symposium on Circuits and Systems, ISCAS*'09, Taipei, May 24-27, 2009, pp. 2585–2588.

[P11] S. Tao, Y. Chen, M. M. Hannuksela, Y.-K. Wang, M. Gabbouj and H. Li, "Joint Texture and Depth Map Video Coding Based on the Scalable Extension of H.264/AVC," *IEEE International Symposium on Circuits and Systems, ISCAS'09,* Taipei, May 24-27, 2009, pp. 2353–2356.

List of Supplementary Publications

The contents of this thesis are also closely related to the following publications by the author:

[S1] Y. Chen, Y. -K. Wang, and M. Gabbouj, "Buffer Requirement Analysis for Multiview Video Coding," *Picture Coding Symposium*, *PCS*'07, Lisbon, Portugal, November 2007.

[S2] Y. Chen, K. Xie, F. Zhang, P. Pandit, and J. Boyce, "Frame Loss Error Concealment for SVC," *Journal of Zhejiang University* SCIENCE *A, also in International Packet Video Workshop,* Hangzhou, China, April 2006.

List of Abbreviations

ABT	Adaptive Block-size Transform
ASO	Arbitrary Slice Ordering
AVC	Advanced Video Coding standard
BMA	Boundary Matching Algorithm
CGS	Coarse Granularity Scalability
CIR	Cyclic Intra Refresh
CRC	Cyclic Redundancy Check
CABAC	Context-Adaptive Binary Arithmetic Coding
CAVLC	Context-Adaptive Variable-Length Coding
DCT	Discrete Cosine Transform
DIBR	Depth-Image-Based Rendering
DPB	Decoded Picture Buffer
DPCM	Differential Pulse Code Modulation
DVBS	Digital Video Broadcasting System
EBCOT	Embedded Block Coding with Optimal Truncation
EZW	Embedded Zerotree Wavelet
FGS	Fine Granularity Scalability
FIR	Finite Impulse Response
FMO	Flexible Macroblock Ordering
FREXt	Fidelity Range Extensions
GDR	Gradual Decoding Refresh
GOP	Group Of Pictures
HDTV	High Definition TV
IDR	Instantaneous Decoding Refresh
IEC	International Electrotechnical Commission
ISO	International Standardization for Organization
ITU	International Telecommunication Union
ITU-T	ITU Telecommunication Standardization Sector
JMVM	Joint Multiview Video Model
JSVM	Joint Scalable Video Model
JVT	Joint Video Team
LAN	Local Area Network
MANE	Media Aware Network Element

MB	MacroBlock
MC	Motion Compensation
MCTF	Motion Compensated Temporal Filtering
MDC	Multiple Description Coding
MGS	Medium Granularity Scalability
MLD	multiple-loop decoding
MMCO	Memory Management Control Operation
MPEG	Moving Picture Experts Group
MV	Motion Vector
MVC	Multiview Video Coding
NAL	Network Abstraction Layer
POC	Picture Order Count
PSNR	Peak Signal-to-Noise Ratio
QoS	Quality of Service
QP	Quantization Parameter
QVGA	Quarter Video Graphics Array
RD	Rate-Distortion
RDO	Rate-Distortion Optimization
RIR	Random Intra Refresh
PLR	Packet Loss Rate
ROPE	Recursive Optimal Per-pixel Estimate
RPLM	Reference Picture List Modification/ Reference Picture List Reordering
RPMR	Reference Picture Marking Repetition
RTP	Real-time Transport Protocol
SAD	Sum of Absolute Difference
SDTV	Standard Definition TV
SEI	Supplemental Enhancement Information
SLD	Single-Loop Decoding
SLEP	Systematic Lossy Error Protection
SNR	Signal-to-Noise Ratio
SVC	Scalable Video Coding
UDP	User Datagram Protocol
UEP	Unequal Error Protection
VCEG	Video Coding Experts Group
VCL	Video Coding Layer
VGA	Video Graphic Array
WLAN	Wireless Local Area Network
3GPP	Generation Partnership Project

Chapter 1

Advanced Video Coding

The H.264/AVC video coding standard is developed by the Joint Video Team (JVT) of the Moving Picture Experts Group (MPEG) of the International Standardization for Organization (ISO)/International Electrotechnical Commission (IEC) and Video Coding Experts Group (VCEG) of the International Telecommunication Union (ITU) Telecommunication Standardization Sector (ITU-T). H.264/AVC is published by MPEG as MPEG-4 part 10 Advanced Video Coding (AVC) and by ITU-T as ITU-T Recommendation H.264. Several versions of the standards have been released, some versions include new amendments. In particular, the version published in November 2007 refers to the standard including the scalable video coding Amendment and the version published in March 2009 refers to the standard including the multiview video coding Amendment. In this thesis, if without further explanation, by default, SVC refers to the scalable extension of H.264/AVC and MVC refers to the multiview extension of the H.264/AVC. They are specified as Amendments in the Annex G and Annex H of the H.264/AVC specification, respectively.

To deliver video over a channel with a limited bandwidth, coding efficiency is important. High efficiency corresponds to a high video quality with a fixed bandwidth, or a lower bandwidth with the same video quality. As the coding efficiency of the video standards gets higher, the computational complexity required at the decoders becomes a concern for real-time decoding. In addition, video services are provided through lossy channels, so dealing with errors is necessary either at the encoder or at the decoder. When those issues are jointly considered, interactivities of the coding layer and the transmission layer are necessary and thus require a good design of the system level interface for video coding standards. In this thesis, the proposed work is motivated by improving coding efficiency, controlling decoder side resource, minimizing the degradation at the encoder and decoder caused by packet losses, and facilitating the system level interface of the video standards.

As the basis of the thesis work, in this chapter, concepts, techniques, and application scenarios that are necessary for understanding the contributions of the thesis, are firstly described in Section 1.1. More specifically error resilience and concealment for H.264/AVC

are introduced afterwards in Section 1.2. Contribution of the thesis is summarized in Section 1.3, followed by the outline of the thesis, as described in Section 1.4.

1.1. OVERVIEW OF ADVANCED VIDEO CODING

H.264/AVC includes the coding of the typical 4:2:0, 8-bit, one-representation, video sequences in the earlier profiles. One extension of H.264/AVC is for high fidelity video, such as high bit-depth per sample, 4:2:2 and 4:4:4 chroma sampling formats. Another two extensions are for the scalable video applications and multiview video applications, wherein the video conveyed has, for one scene, different representations either with different SNR (Signal-to-Noise Ratio) and/or spatial quality, or from different viewing perspectives. These two extensions are Scalable Video Coding (SVC) and Multiview Video Coding (MVC), both of which are part of the H.264/AVC standard, although SVC and MVC are not presented in detail in this chapter.

H.264/AVC provides a support for the traditional Discrete Cosine Transform (DCT) plus Differential Pulse Code Modulation (DPCM) codec. As other video coding standards, such as MPEG-2 [1], and MPEG-4 part 2 visual [2], a picture is coded as a series of maroblocks (MBs), each of which uses either intra prediction or inter prediction. When inter prediction is used, the previously decoded signal is employed to generate a predicted signal with the help of certain motion vectors. The difference between the intra/inter predicted signal and the original signal of an MB is DCT transformed, quantized and entropy coded.

The features of H.264/AVC are reviewed in [3][4]. A block diagram of H.264/AVC coding is shown in Fig. 1, wherein motion estimation and mode decision are performed as encoding processes, and the decoder performs motion compensation based on the signaled motion vectors to get the predicted signal. Q^{-1} and T^{-1} are the inverse quantization and inverse transform, respectively. Quantization, inverse quantization, transform and inverse transform of H.264/AVC are introduced in [5].

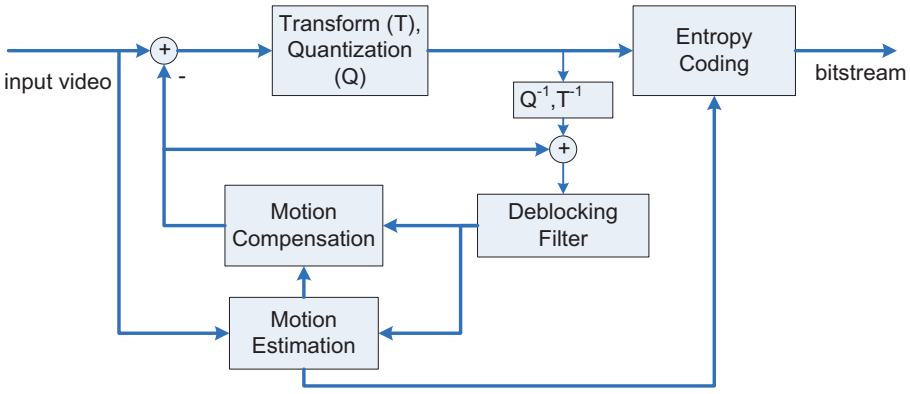


Fig. 1: Block diagram of H.264/AVC.

As most of video coding standards, H.264/AVC defines the syntax, semantics and decoding process for error-free bitstreams, any of which is conforming to a certain profile or level. The encoder is not specified but needs to guarantee that the generated bitstreams are

standard decoder compliant. For a video coding standard, the important design considerations are: coding efficiency, decoder complexity and the interactivities with the system.

In the context of video coding standard, a profile corresponds to a subset of algorithms, features or tools and constraints that apply to them; a level corresponds to the limitations of the decoder resource consumption, i.e., decoder memory and computation, which are related to the resolution of the pictures, bit rate and MB processing rate. A decoder conforming to a profile must support all the features defined in the profile. A decoder conforming to a level must be capable of decoding any bitstream that does not require resources beyond the limitations defined in the level. Profiles and levels are helpful for interoperability. For example, during video transmission, a pair of profile and level needs to be negotiated and agreed for a whole transmission session.

In H.264/AVC [3][4], the following major new coding tools are introduced, which are supported in all the profiles of H.264/AVC. They are:

- Advanced intra prediction
- Integer 4x4 Discrete Cosine Transform (DCT) transform [5]
- Flexible multiple reference pictures [6]
- High accuracy of motion vectors (1/4 sample)
- Hierarchical macroblock partitioning [6]
- Intra coded MB in Inter coded pictures
- Context-Adaptive Variable-Length Coding (CAVLC)
- In-loop deblocking filter [7]

Note that all profiles support 4:2:0 chroma sample format and 8 bit sample depth.

In the baseline profile and extended profile, the following extra tools are supported.

- Flexible Macroblock Ordering (FMO) [8]
- Arbitrary Slice Ordering (ASO)
- Redundant picture [8]

Compared with the baseline profile, the extended profile supports the following four tools:

- Support of B pictures
- Interlaced coding
- Weighted prediction [9]
- Data partition [8]
- SP/SI slices [10]

To reduce the decoder implementation cost, a constrained baseline profile was standardized for video conferencing and mobile applications. It corresponds to a subset of features that are shared by the baseline, the main and the extended profiles. In other words, this profile contains all the features of the baseline profile except the error resiliency tools, ASO, FMO and redundant pictures. The differences and relations of the feature sets of these profiles are illustrated in Fig. 2, wherein the constrained baseline corresponds to the intersection area of the following three: baseline, main, and extended profiles.

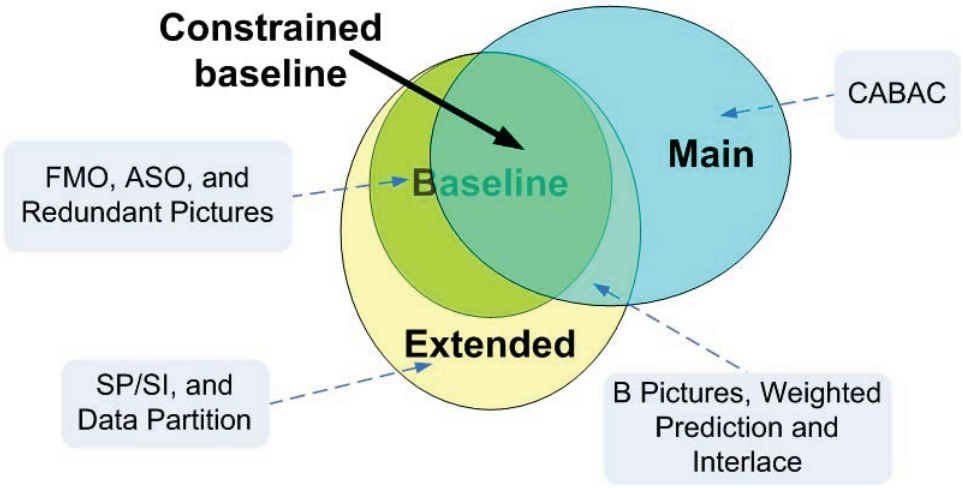


Fig. 2: Relations of feature sets of H.264/AVC profiles.

However, in the main profile, less error resilient tools are supported with high coding efficiency. Compared with the extended profile, error resilient tools, FMO, ASO, redundant picture, data partition, are not supported and SP/SI slices, which are targeting channel switching, are not supported. The only characteristic that is not supported in the extended profile but is supported in the main profile is:

- Context-Adaptive Binary Arithmetic Coding (CABAC) [11]

There are other profiles, namely Fidelity Range Extensions (FREext) profiles [3][12][13]: high profile, high 10 profile, high 4:2:2 profile and other advanced profiles. Note that the characteristics supported in main profile are a subset of those in high profile. It is the same characteristics relationship between high profile and high 10 profile or between high 10 profile and high 4:2:2 profile. The first three FREext profiles are oriented from the main profile with the following common characteristics:

- 4:0:0 chroma sample format
- 8x8 Integer DCT Transform [14]
- Adaptive Block-size Transform (ABT, between 4x4 DCT and 8x8 DCT) [14]
- Separate Quantization Parameter (QP) control for the two chroma components

In high 10 profile, up to 10 bit sample depth is supported. In high 4:2:2 profile, 4:2:2 chroma sample format is supported.

High 4:4:4 profile is a superset of high 4:2:2 and support the following new futures [15]:

- Up to 14 bit sample depth
- Separate color space coding (Each color component array is treated as a separate monochrome video picture and is independently coded)
- Lossless DPCM coding
- Improved intra prediction [16].

The relations of the future sets in the main profile and the FREext profiles are illustrated in Fig. 3.

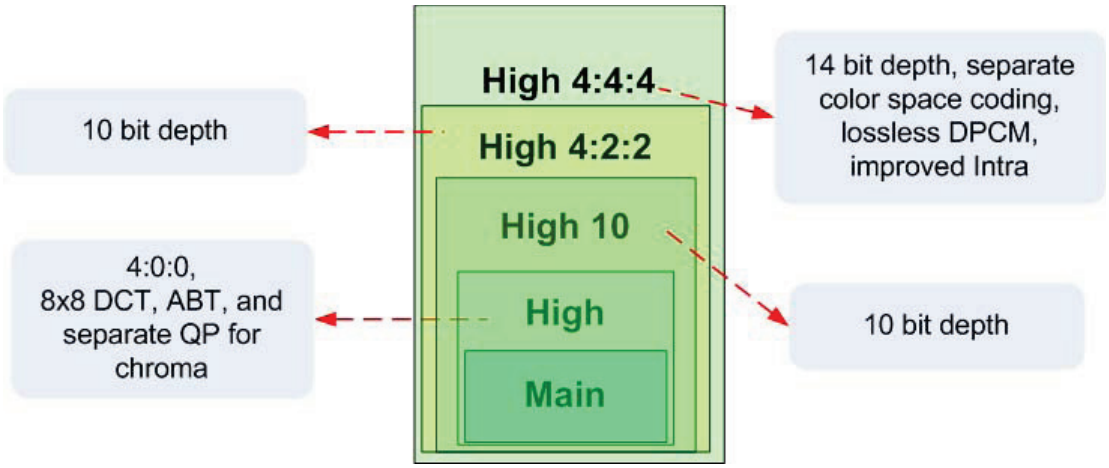


Fig. 3: Relations of feature sets of main profile and FREXt profiles.

Note that there are other profiles that are the subsets of the above high profiles. Those profiles are created with either constraints on predictive coding (Intra only) or CABAC coding (CAVLC only).

In the following part of this section, details of AVC techniques that are closely related to the research work of this thesis are reviewed. Those are helpful for understanding the mechanisms and algorithms proposed in this thesis.

1.1.1. Coded Pictures and Bitstream Structure

In H.264/AVC, the coded video bits are organized into Network Abstraction Layer (NAL) units, which provide a "network-friendly" video representation addressing the applications such as video telephony, storage, broadcast, or streaming. Network Abstraction Layer (NAL) units can be categorized to Video Coding Layer (VCL) NAL units and non-VCL NAL units. The supported VCL NAL unit types and non-VCL NAL units in H.264/AVC are defined in the H.264/AVC specification [3] and well categorized in [8] and [17]. The VCL units contain the core compression engine and comprise block, MB and slice levels. Other NAL units are non-VCL NAL units. A slice contains a series coded MBs which are coded using Intra or Inter mode.

In AVC, a coded picture, normally presented as a primary coded picture, is contained in an access unit, which consists of one or more NAL units. Multiple NAL units for a primary coded picture is only supported in the baseline and extended profiles. In those profiles, a picture can be coded into multiple slices. Each slice is contained in one NAL unit (when data partition is not used) and is independent from other slices for the parsing of the bits inside and can be decoded without prediction from any slice in the same picture. That is, the entropy decoding of a slice can be done without accessing the data in another slice.

In AVC, a coded picture can have different orders for display and decoding. In H.264/AVC, the order how NAL units are placed inside the bitstream is referred to as the decoding order. Picture Order Count (PicOrderCnt, POC) can be used to specify the display order of a coded picture. Frame number (frame_num) can be used to indicate the decoding

order, although a non-reference picture can have the same frame number as the closest reference picture in the decoding order.

1.1.2. Hierarchical Macroblock Partitioning

A VCL NAL unit typically contains the signal of a series of coded MBs. A MB can be coded with different modes. Each MB can be coded as Intra MB or Inter MB. H.264/AVC Inter mode codes an MB with signals predicted from the reconstructed texture of the already decoded pictures. When an MB is Inter coded, it may be further partitioned into MB partitions, which are of 16x16, 16x8, 8x16 or 8x8 sizes, as shown in the upper part of Fig. 4. An MB or MB partition uses the same reference picture, or pictures when it is bi-predicted [6]. An Intra slice contains only Intra MBs; A Inter-P (P) slice can contain predicted MBs and Intra MBs; A Inter-B (B) slices can contain bi-predicted MBs as well as predicted MBs and Intra MBs.

Furthermore, each MB or MB partition can be partitioned into 8x8, 8x4, 4x8 or 4x4 blocks (or sub-macroblock partitions), as shown in the bottom part of Fig. 4. The samples in each block share the same motion vector (or 2 motion vectors for bi-prediction: one motion vector for each direction).

The H.264/AVC based or compliant standards so far all follow this hierarchical macroblcok partitioning because it will make the hardware design module for the motion compensation part applicable to the extension standards of the H.264/AVC.

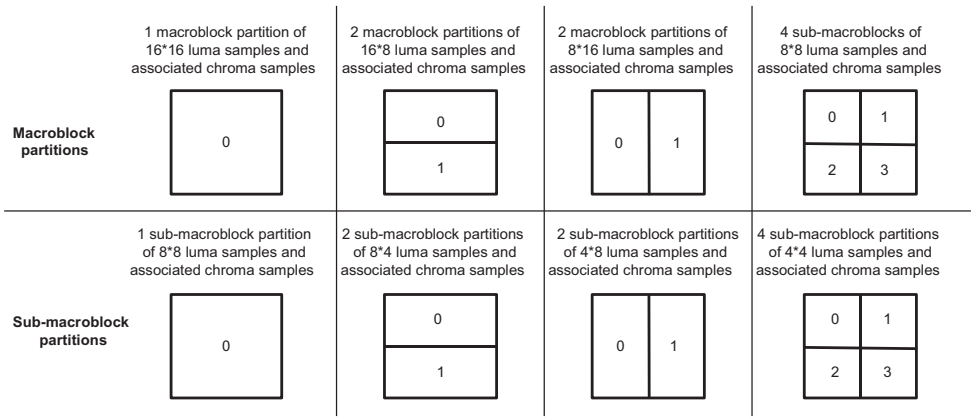


Fig. 4: **Hierarchical macroblock partitioning**

1.1.3. Decoded Pictures and their Buffer Management

Decoded pictures used for predicting subsequent coded pictures and for future output are buffered in the Decoded Picture Buffer (DPB). To efficiently utilize the buffer memory, the DPB management processes, including the storage process of decoded pictures into the DPB, the marking process of reference pictures, output and removal processes of decoded pictures

from the DPB, are specified. In AVC, any reference pictures in the DPB can be used as a reference picture and thus higher coding efficiency can be achieved. This is realized by the reference picture list construction [13].

Reference Picture Marking

The process for reference picture marking in AVC is summarized as follows [13].

The maximum number, referred to as M, of reference pictures used for inter prediction is indicated in the active sequence parameter set. When a reference picture is decoded, it is marked as "used for reference". If the decoding of the reference picture caused more than M pictures marked as "used for reference", at least one picture must be marked as "unused for reference". The DPB removal process then would remove pictures marked as "unused for reference" from the DPB if they are not needed for output as well.

When a picture is decoded, it is either a non-reference picture or a reference picture. A reference picture can be a long-term reference picture or short-term reference picture, and when it is marked as "unused for reference", it becomes a non-reference picture. In AVC, there are reference picture marking operations that change the status of the reference pictures.

There are two types of operations for the reference picture marking: sliding window and adaptive memory control. The operation mode for reference picture marking is selected on picture basis; whereas, sliding window operation works as a first-in-first-out queue with a fixed number of short-term reference pictures. In other words, short-term reference pictures with earliest decoding time is firstly to be removed (marked as picture not used for reference), in a implicit fashion. The adaptive memory control however removes short-term or long-term pictures explicitly. It also enables switching the status of the short-term and long-term pictures, etc.

Reference Picture Lists Construction

When decoding a coded slice, a reference picture list (list0) is constructed. If the coded slice is a bi-predicted slice, a second reference picture list (list1) is also constructed [13].

For simplicity, it is assumed herein that only one reference picture list is needed. To construct a reference picture list, first, an initial reference picture list is constructed with a certain order usually related to the decoding (if the current picture is P picture) or display order (if the current picture is a B picture). Then, Reference Picture List Modification (RPLM) is performed when the slice header contains RPLM commands.

1.1.4. Motion Compensation

In H.264/AVC, the accuracy of motion compensation is in the units of one quarter of the distance between luma samples [13][18].

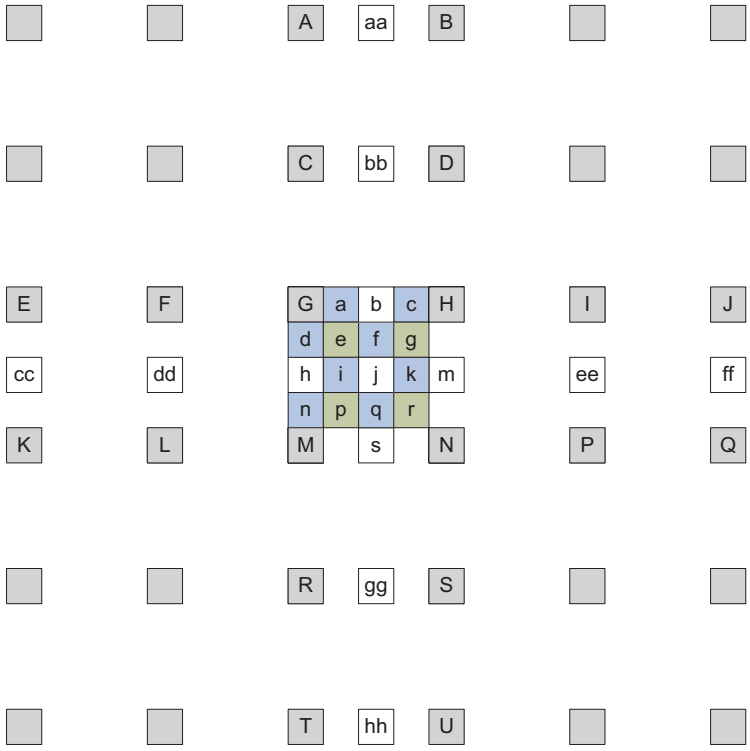


Fig. 5: Integer samples (shaded blocks with upper-case letters) and fractional sample positions (un-shaded blocks with lower-case letters) for quarter sample luma interpolation.

In case the motion vector points to an integer-sample position (as shown in Fig. 5 with upper-case letters), the prediction signal consists of the corresponding samples of the reference picture; otherwise, the corresponding sample is obtained using interpolation to generate non-integer positions.

In case the motion vector points to half-sample positions. The values in those positions are interpolated. If a half-sample position has one dimension which aligns to integer samples, e.g., those with double lower-case letters, the H.264/AVC 6-tap FIR (finite impulse response) is applied to interpolate the value half-sample position, otherwise, the prediction values at half-sample positions (e.g., position j in Fig. 5) are obtained by applying a one-dimensional H.264/AVC 6-tap filter first horizontally and then vertically. The 6-tap filter utilized in H.264/AVC half-sample interpolation is [1, -5, 20, 20, -5, 1]/32. This filter is an even-tap symmetric FIR filter.

In case the motion vector points to quarter-sample positions. The values in those half-sample positions are also interpolated and prediction values at quarter-sample positions are generated by averaging samples at integer-sample and half-sample positions.

In 4:2:0 sample format, a motion vector can point to integer sample, half sample, quarter sample or even 1/8-sample positions in the chroma components.

When the motion vector points to a non-integer sample position in the chroma component, the prediction values for the chroma component are obtained by the bilinear filter, as shown in Fig. 6, with the equation (1):

$$v = \left((s - d_x)(s - d_y)A + d_x(s - d_y)B + (s - d_x)d_y C + d_x d_y D + s^2/2\right)/s^2 \qquad (1)$$

wherein s is 8 and d_x and d_y are in the rage of 1 to 7 and are the horizontal or vertical distance to the top-left integer sample in a unit of 1/8-pel.

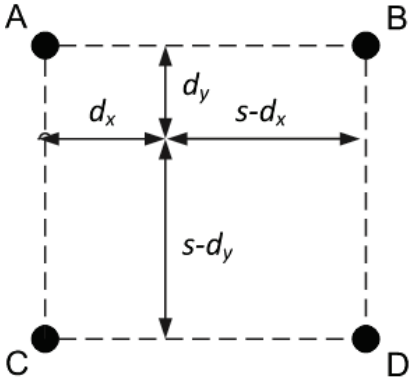


Fig. 6: Bilinear interpolation for chroma values in the non-integer positions.

1.1.5. Supplemental Enhancement Information

Supplemental Enhancement Information (SEI) contains information that is not necessary for decoding the coded pictures samples from VCL NAL units. SEI messages are also contained in non-VCL NAL units [17].

SEI messages are the normative part of the standard specification, while not mandatory for standard compliant decoder implementation. SEI messages assist in processes related to decoding, display, error resilience and other purposes.

1.2. ERROR RESILIENT CODING AND ERROR CONCEALMENT FOR H.264/AVC

In this section, common requirements for video transmission and error control tools are reviewed, and then the error resilient and concealment tools in H.264/AVC are reviewed.

1.2.1. Error Robust Requirement and Error Control Tools

The number of packet-based video transmission channels, such as the Internet and packet-oriented wireless networks, has been increasing rapidly. One inherent problem of video transmitted in packet-oriented connectionless protocol environments is channel errors. Packet loss may be caused due to an overloaded network node (switch, router, etc.) or because a packet reaches the destination with such a long latency that it was already considered useless or lost.

Another source of packet loss is bit errors caused by physical interference in any link of the transmission path. Many video communication systems apply the User Datagram Protocol (UDP) [19]. Any bit error occurred in a UDP packet will result in the loss of the packet, as UDP discards packets with bit errors. Packet loss can damage one whole picture or an area of it. More unfortunately, due to the predictive coding technique, a transmission

error (after error concealment) will propagate both temporally and spatially, and sometimes can bring substantial deterioration to the subjective and objective quality of the reproduced video sequence until an Instantaneous Decoding Refresh (IDR) picture is received and decoded. However, if the bitstream is protected by error control methods [20], the system may still maintain graceful degradation.

Various error control methods have been proposed. In [21], error control methods are classified into four types as follows: transport-level error control, source-level error resilient coding, interactive error control, and error concealment.

For error control, the contributions of this thesis mainly focus on source-level error resilient coding and error concealment.

1.2.2. Error Resilient Tools for H.264/AVC

Reference picture identification

Reference pictures are labeled by a fixed-length syntax element (i.e. frame_num in H.264/AVC), which is incremented by one (wrapped to zero after the maximum value) for each reference picture. For non-reference pictures, the value is incremented by one in relation to the value of the previous reference picture in decoding order. This frame number enables decoders to detect the loss of reference pictures. If there is no loss of reference pictures, the decoder can go on decoding; even if there is loss of non-reference pictures. Otherwise, a proper action should be taken, as temporal error propagation will occur. This concept was established firstly in H.263 (Annex U and subclause W.6.3.12 of Annex W), wherein term “picture number” was used [22]. The same idea was later also adopted in H.264/AVC as in the form of syntax element frame_num in the slice header.

Gradual Decoding Refresh (GDR)

GDR, which is enabled by the so-called isolated region technique [23], can be used for gradual random access, error resilience as well as other purposes. An isolated region in a picture can contain any MBs in a picture, whose locations can be indicated by the MB to slice group mapping included in the picture parameter set. A picture can contain zero or more isolated regions that do not overlap, and the rest of the picture is a leftover region. A coded isolated region can be decoded without the presence of any other isolated or leftover region of the same coded picture. Meanwhile, the isolated region can only be predicted from the corresponding isolated region in the reference pictures. Therefore, no error can be propagated from any other regions temporally or spatially. An isolated region evolving over time can completely stop error propagation resulted from packet losses occurred before the starting point of the isolated region in a gradual manner, i.e. after the isolated region covers the entire picture area. An example of GDR is shown in Fig. 7, wherein the errors of the previous pictures are not propagated and the picture quality has improved.

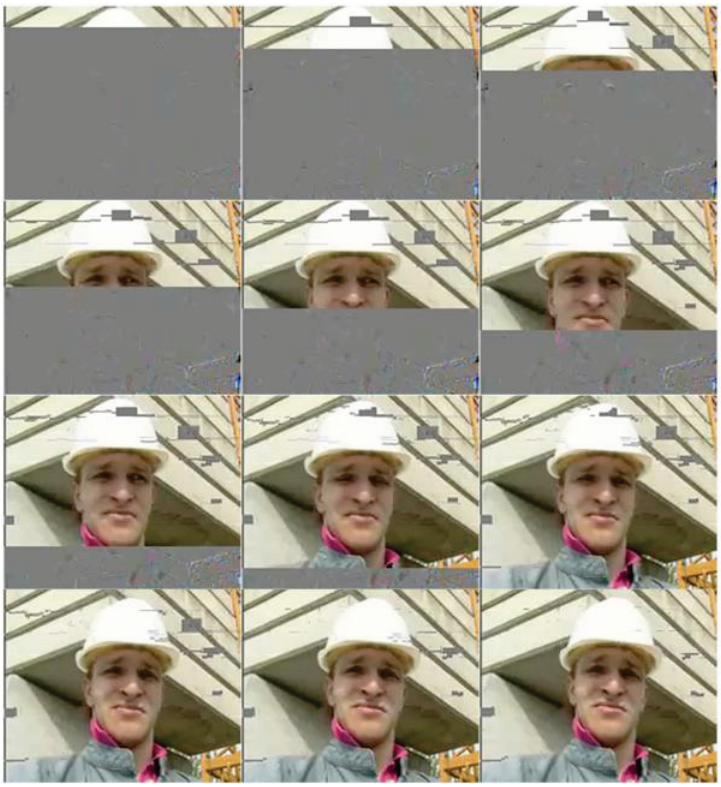


Fig. 7: An Example of Gradual Decoder Refresh.

Redundant slices/pictures

A redundant slice/picture is a coded representation of a primary picture or a part of a primary picture. The decoder should not decode redundant pictures when the corresponding primary picture is correctly received and can be correctly decoded. However, when the primary picture is lost or cannot be correctly decoded, a redundant picture can be utilized to improve the decoded video quality, if the redundant picture or part of it can be correctly decoded. A redundant picture can be coded as an exact copy of the primary picture, or with different coding parameters. Redundant pictures even do not have to cover the entire region represented by the primary pictures.

In [24], exact-copy redundant pictures were encoded for unequal protection of pictures that use relatively long-term reference pictures for inter prediction. Redundant pictures can also be encoded with some quality degradation using larger QPs than in the primary pictures, such that fewer bits will be used to represent redundant pictures. The method called Systematic Lossy Error Protection (SLEP) [25] belongs to this category. In [26] , Multiple Description Coding (MDC) was realized using redundant pictures in an H.264/AVC compatible manner. Another H.264/AVC redundant picture based MDC method was also reported in [27], wherein the coded slices of a primary picture and a redundant picture are interleaved into two descriptions. An H.264/AVC compatible redundant picture coding method, in combination with RPS, reference picture list reordering and adaptive redundant picture allocation was reported in [28].

Reference Picture Marking Repetition (RPMR)

RPMR, by use of the decoded reference picture marking repetition SEI message, can be used to repeat the decoded reference picture marking syntax structures in the earlier decoded pictures [13]. Consequently, even when earlier reference pictures were lost, the decoder can still maintain correct status of the reference picture buffer and reference picture lists. Incorrect status of the reference picture buffer and reference picture lists will always result in corrupted decoding even for pictures using correctly decoded reference pictures.

Spare picture signaling

The spare picture SEI message, signaling the similarity between a reference picture and other pictures, tells the decoder which picture can be used as a substituted reference picture or can be used to better conceal the lost reference picture [29]. Therefore, the reconstructed error in the reference picture can be minimized. Furthermore, unnecessary picture freezing, feedback and complex error concealment can be prevented.

Scene information signaling

The scene information SEI message provides a mechanism to select proper error concealment method for Intra pictures, scene-cut pictures and gradual scene transition pictures at the decoder [30]. For example, for a picture that is indicated as a scene-cut picture by a scene information SEI message, if it is entirely lost, the display in the decoder size can freeze the displayed video until a refresh picture is decoded; if it is partially lost, spatial error concealment method instead of temporal error concealment method should be applied to conceal the lost area.

Constrained intra prediction

Intra prediction utilizes available neighboring samples in the same picture for prediction of the coded samples to improve the efficiency of intra coding. In the constrained intra prediction mode, samples from inter coded blocks are not used for intra prediction. The use of this mode can improve error resilience. If errors occur in reference pictures, they may propagate to inter coded blocks of the current picture. Consequently, even if only intra prediction was used, the error may still propagate to the intra coded blocks due to intra prediction thus temporal error propagation cannot be efficiently stopped.

Intra MB/picture refresh

Intra refresh intentionally inserts Intra pictures or Intra MBs into the bitstream. Statistically, this method is not the optimal from a Rate-Distortion (RD) model point of view with error-free channel. However, it can still achieve a better RD performance in packet loss conditions. One essential reason is that errors from a reference picture can be mitigated without motion compensated temporal prediction.

The insertion of an Intra picture is a very simple and efficient technique, whereas it will lead to a large redundancy to the video. A lot of methods for insertion of Intra MBs have

been reported. Random Intra Refresh (RIR) [31] and Cyclic Intra Refresh (CIR) [32] are well known methods and used extensively. In RIR, the Intra-coded MBs are selected randomly from all the MBs of the picture, or from a finite sequence of the pictures. On the other hand, in CIR, each MB is intra updated at a fixed period, according to a fixed "update pattern". Neither algorithm takes the picture content or the bit stream properties into account. The verification model of MPEG-4 Visual has an algorithm which refreshes Intra MB adaptively according to the Sum of Absolute Difference (SAD) calculated between the spatially corresponding motion compensated MBs in the reference picture buffer. In [33], a Recursive Optimal Per-pixel Estimate (ROPE) algorithm was proposed, in which the first and second moment of each pixel under channel error are estimated. The moments can be used for the future picture that will refer it, and the end-to-end distortion propagated from this pixel will be calculated. This method can only be used for integer pixels. For sub-pixels, a modified algorithm was proposed later in [34]. LA-RDO mode decision in H.264/AVC [35] contains a high complexity MB selection method that places Intra MBs according to the RD characteristics of each MB. It needs to simulate a number of decoders at the encoder and each simulated decoder independently decodes the MB at a given Packet Loss Rate (PLR). For better performance, simulated decoders also apply error-concealment to lost MBs. The expected distortion of a MB is averaged over all the simulated decoders and this average distortion is used for mode selection. This LA-RDO method generally gives good performance, but it is not feasible for many implementations as the complexity of the encoder increases significantly due to simulating a potentially large number of decoders (30 was used in a lot of reported simulations). Thereafter, an error propagation map based algorithm was presented in [36], and it only needs to estimate the distortion based on 4x4 block map, so it has a much lower computational complexity; furthermore, a better performance than [35] was reported.

1.2.3. Error Concealment for H.264/AVC

Error concealment is a decoder-only technique. Typically, the spatial, temporal, and spectral redundancy can be used to mask the effect of channel errors in the decoder.

If the picture is partially corrupted, e.g., the picture is split into multiple slices, some of which are lost while others are received, the technique described in [37] can be used. In this technique, the lost area of the Intra picture is interpolated based on weighted pixel averaging of boundary pixels, and the weights used for averaging are inversely proportional to the distance between the source and destination pixels. The lost area of the Inter picture's Motion Vector (MV) is recovered by the well known Boundary Matching Algorithm (BMA). The chosen MV from the neighboring MVs is the one which can minimize the difference between the external boundary of the lost block from the current picture and the internal boundary of the reference picture block used for concealing the lost block. After finding the best MV, motion compensation can be used to reconstruct the block.

For low bitrate video transmission such as 3G wireless systems, usually, one picture is coded into only one packet, and loss of this packet implies that the entire picture must be recovered from the previously decoded pictures. The simplest way to solve this problem is by copying the previously decoded picture to replace the lost one. However, if the sequence contains a smooth motion, motion copy [38] can be used to improve the performance. The method first estimates the motion vectors, reference indices and partitioning modes for the blocks of the lost picture from the co-located blocks of the previously decoded picture, and then motion compensation is used to reconstruct the lost picture.

1.3. AUTHOR'S CONTRIBUTION TO THE PUBLICATIONS

The author's contributions to scalable video and multiview video are mostly reflected in the primary publications, denoted as [P1], [P2]…, [P11]. While all publications have resulted from team work, the author's contribution to each has been essential as described next.

Publication [P1] proposed an encoder algorithm for H.264/AVC compliant to the baseline profile with temporal scalability. The proposed algorithm achieves higher coding efficiency for H.264/AVC baseline profile. The author of this thesis proposed the methods and contributed to the implementation and simulations as well as the paper writing.

Publication [P2] extends the scalable extension of H.264/AVC into bit-depth scalability, although this scalability is not part of the scalable extension of H.264/AVC. The proposed solution provides substantial bandwidth reduction or PSNR (Peak Signal-to-Noise Ratio) gain compared to simulcast coding. Most of the author's efforts are for proposing and implementing the original idea. The paper also includes further extensions of the original idea which were proposed by the co-authors. The author has written part of the paper.

Publication [P3] reviews the MVC standard, as an extension of H.264/AVC, especially those essential standard features that were added on top of H.264/AVC into MVC, for different functionalities. As the first author of this paper, the author of this thesis is one of the key members of the team that proposed and implemented most of the reviewed techniques and solutions and he wrote most part of the paper. It is worth mentioning that the thesis author and some of the other co-authors of publication [P3] proposed them into the international standard committee JVT, and they were adopted as part of the MVC standard. The parallel decoding part of the paper, however, was not contributed by the thesis author.

Publication [P4] proposed error resiliency and concealment algorithms for SVC, those algorithms have been adopted in the standard software of SVC. The thesis author proposed, implemented and wrote the error concealment part of the paper. As this publication also reviews the existing AVC and SVC error resiliency and concealment mechanism, the author has also contributed in this review part of this publication.

Publication [P5] proposed an error concealment algorithm for MVC. The proposed algorithm outperforms other typical low-complexity error concealment algorithms. The thesis author proposed the idea and coordinated the implementation and simulations within the team and wrote parts of the paper.

Publication [P6] reviewed the coding techniques in multiview video coding and joint multiview video model. As the first author of this paper, the thesis author wrote this publication with discussions with other co-authors of the publication. The simulations in this paper were designed by the first author and carried out partly by the first author.

Publication [P7] proposed a single-loop decoding method for multiview video coding. It deduces the decoder computations and memory, without significant coding efficiency loss. As the first author of this paper, the thesis author proposed and implemented the algorithm and wrote the publication, which was reviewed and edited by other co-authors.

Publications [P8], [P9] and [P10] presented a series of coding tools which enables coding of the asymmetric stereoscopic video. Asymmetric stereoscopic video has one view with quarter resolution of the others. The algorithms can achieve complexity deduction and bitrate reduction. The thesis author proposed the ideas and carried out a large part of the implementation work. He also wrote major parts of the papers.

Publication [P11] took advantages of the motion vector correlation between the texture and depth videos of the same scene, in the context of 3D video which may use depth maps at the decoder for view synthesis. The proposed algorithm improves the efficiency for the coding of depth maps. The idea of this paper originated from a discussion between the thesis author and the third author. The thesis author was involved in the implementation, the simulations and the writing of the paper.

1.4. OUTLINE OF THE THESIS

In Chapter 2, the structure of SVC (the scalable extension of H.264/AVC) standard as well as the features of SVC is firstly reviewed. The features of SVC include hierarchical temporal scalability, inter-layer prediction, single-loop decoding and flexible transport interface. The contributions of this thesis in SVC include hierarchical P picture coding, which is also compatible with the baseline profile of H.264/AVC and color bit-depth scalability based on the SVC platform.

MVC (the multiview extension of H.264/AVC) standard is introduced in Chapter 3. The work of the thesis provides many features which have been adopted in the specification of MVC, to meet the requirements for multiview video content applications. The introduced MVC features cover bitstream adaptation capability, random access and view switching and decoding order and decoded picture management.

In Chapter 4, error resiliency and concealment algorithms for SVC are firstly introduced. This thesis contributes on error concealment algorithms for SVC and MVC. The proposed algorithms are with low complexity but outperform the methods with similar complexity.

In Chapter 5, the coding tools of MVC are reviewed. Then a series of tools for 3D video content coding are introduced. The contributions of this thesis contain single-loop decoding for MVC, asymmetric coding with lower complexity and high coding efficiency and joint coding of depth and texture.

Conclusions for this contribution of this thesis are given in Chapter 6.

Chapter 2

Scalable Video Coding (SVC): The Scalable Extension of H.264/AVC

Since 2004, the JVT has been working on a new standardized design for Scalable Video Coding (SVC). The SVC project is the scalable extension of the H.264/AVC standard, adding the temporal, spatial, and SNR (Signal-to-Noise Ratio) scalability features. The standard was finalized in 2007 [39].

As an extension of H.264/AVC, SVC offers a full backward compatibility with H.264/AVC, meaning that any SVC bitstream can be decoded by an H.264/AVC decoder, to get a possibly low spatial/temporal/SNR representation. In addition, the SVC design takes care of the trade-offs between coding efficiency and computational complexity.

In Section 2.1 and 2.2, SVC, as a standard is firstly introduced. The application scenarios of SVC are introduced in Section 2.3. Two algorithms are presented in Section 2.4 and 2..5. The fist algorithm is for the AVC baseline encoding with hierarchical P coding structure, the second one is for color bit-depth scalability. Although the first encoding algorithm is based on H.264/AVC, it is in the scope of temporal scalability, which is supported in SVC. Therefore, the encoding algorithm is presented in this SVC chapter. Other scalabilities are mentioned in Section 2.6.

2.1. SCALABLE VIDEO CODING - AN OVERVIEW

Scalable video coding usually refers to the coding of a high-quality video bitstream that contains one or more subset bitstreams. A subset bitstream can themselves be decoded with lower complexity and lower reconstruction quality than the complexity required by decoding the whole bitstream and the quality reconstructed from a whole bitstream. The subset bitstreams are layered together to form the whole bitstream and any of them can be derived

by dropping packets belonging to high layers. Scalable video concepts and the structure of an SVC bitstream are reviewed in this section.

2.1.1. Scalable Video Coding Concepts

Scalability refers to the feature of enabling lower representations of the video content while enhancing them in one or more dimensions. The scalability in the context of video coding specifically corresponds to bitstream scalability, wherein a sub-bitstream can have a relatively lower representation with e.g., lower spatial resolution (spatial scalability), lower frame rate (temporal scalability) or lower SNR (SNR/quality scalability). In other words, a scalable bitstream consists in a compressed video content hierarchically organized in successive layers, corresponding to different levels of image quality, frame rate, and picture size. Note that besides these three typical dimensions, other dimensions that have been taken into consideration include complexity, color bit-depth and chroma sampling format.

It is ideal to design a coding scheme with similar or better performances than H.264/AVC, but offering scalabilities with an equivalent or slightly higher complexity. One way widely explored to fulfill these objectives is 3D wavelet coding [40][41][42][43]. The principle is to apply the wavelet decomposition directly on the 3D spatio-temporal signal composed by a group of successive video frames. The 2D spatial decomposition is applied to realize spatial scalability. 1D decomposition, a.k.a. MCTF (Motion Compensated Temporal Filtering) is performed for each pixel of the reference frame of the group along its motion trajectory, to support temporal scalability. Wavelet based video coding use embedded entropy coding based on e.g., EZW [44] or EBCOT [45] algorithms, to get a fully SNR, spatial scalabilities for a picture and thus full SNR, spatial and temporal scalable stream.

However, wavelet coding techniques, including those mentioned above, have not been publicly standardized for video coding, although MPEG-4 part 2 has adopted wavelet coding for still texture [2].

The efforts on scalable video coding in the standards have been made since the early 1990's starting from MPEG-2 video coding. MPEG-2 [1] and MPEG-4 part-2: visual [2] both provide SNR, spatial and temporal scalabilities.

In MPEG-2, SNR scalability is achieved by simply re-quantize the quantization errors of the base layer. In MPEG-4 visual, SNR scalability is realized by Fine Granularity Scalability (FGS), wherein bit-plane coding is used to generate several enhancement layers. However, each FGS enhancement layer can be truncated into any number of bits within each frame to provide partial enhancement proportional to the number of bits decoded for each frame [46].

For spatial scalability, MPEG-2 upsamples the base layer reconstruction picture and makes it as one candidate for reference pictures; while in MPEG-4 visual, the residue between the original picture and the upsampled base layer reconstruction is further coded as if it was the residue between the original picture and the picture predicted from motion compensation.

Temporal scalability can be naturally supported by dropping the coded B pictures in MPEG-2 and MPEG-4 bitstreams. Besides, both of these standards provide the "base layer" and "enhancement layer" concepts to code two temporal layers, wherein the enhancement layer pictures can choose, for each prediction direction, a picture either from the base layer or the enhancement layer as a reference.

However, scalable extensions of these standards were not successful in industrial applications, mainly due to the coding efficiency loss compared with single layer coding. For example, although wavelet based video coding schemes have some inherent advantages for realizing scalability, H.264/AVC based scalable solutions can benefit from the high efficiency of the tools already available in the H.264/AVC specifications.

The scalable extension of H.264/AVC (namely H.264/SVC) provides a higher coding efficiency than the other scalable extension standards because of the following facts: 1. H.264/AVC, as a coding standard for single layer coding, provides a higher coding efficiency than the other standards; 2. H.264/SVC enables MB level adaptation between the H.264/AVC single layer coding modes (inter prediction, intra prediction) and the inter-layer prediction modes. So a RDO encoder is possible to improve the efficiency of the spatial and SNR enhancement layers by always selecting the most efficient modes; 3. H.264/AVC provides advanced reference picture management tools which enable hierarchical B picture coding of multiple temporal enhancement layers and efficient balancing of the bit-allocation among temporal layers. More details on the major features of H.264/SVC are described in Section 2.2.

At the early stage of the SVC development in MPEG, both 3-D wavelet codecs and H.264/AVC based codecs have been proposed and subjective tests in a variety of conditions have been performed. An H.264/AVC based solution was chosen to be the starting point of the SVC standard, mainly because that the wavelet base solutions are less mature and thus have less coding efficiency, especially for the low resolution operation point. Therefore, in the following part of this thesis, SVC refers only to the scalable extension of H.264/AVC, unless stated otherwise.

2.1.2. Structures of Scalable Video Coding based on H.264/AVC

An example of scalabilities in different dimensions is shown in Fig. 8. Scalabilities are enabled in three dimensions. In the time dimension, frame rates with 7.5 Hz, 15 Hz or 30 Hz can be supported by temporal scalability (T). When spatial scalability (S) is supported, resolutions of QCIF, CIF and 4CIF are enabled. In each specific spatial resolution and frame rate, the SNR (Q) layers can be added to improve the quality of the picture. Once the video content has been encoded in such a scalable way, an extractor tool should be used to adapt the actual delivered content according to application requirements, which are dependent e.g., on the clients or the transmission channel. In the example shown in Fig. 8, each cubic contains the pictures with the same frame rate (temporal level), spatial resolution and SNR. Better representation can be normally achieved by adding those cubes (pictures) in any

dimension. Combined scalability is supported when there are two, three or even more scalabilities enabled.

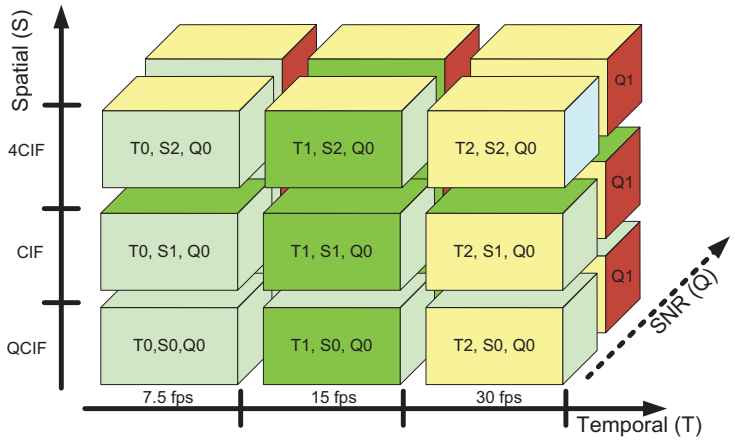


Fig. 8: Scalabilities in three different dimensions.

According to the SVC specifications, the pictures with the lowest spatial and quality layer are compatible with H.264/AVC, and their pictures of the lowest temporal level form the temporal base layer, which can be enhanced with pictures of higher temporal levels. In addition to the H.264/AVC compatible layer, several spatial and/or SNR enhancement layers can be added to provide spatial and/or quality scalabilities. SNR scalability is also referred to as quality scalability. Each spatial or SNR enhancement layer itself may be temporally scalable, with the same temporal scalability structure as the H.264/AVC compatible layer. For one spatial or SNR enhancement layer, the lower layer it depends on is also referred to as the base layer of that specific spatial or SNR enhancement layer.

An example of SVC coding structure is shown in Fig. 9. The pictures with the lowest spatial and quality layer (pictures in layer 0 and layer 1, with QCIF resolution) are compatible with H.264/AVC. Among them, those pictures of the lowest temporal level form the temporal base layer, as shown in layer 0 of Fig. 9. This temporal base layer (layer 0) can be enhanced with pictures of higher temporal levels (layer 1). In addition to the H.264/AVC compatible layer, several spatial and/or SNR enhancement layers can be added to provide spatial and/or quality scalabilities, for example the enhancement layer can be a CIF representation with the same resolution, as shown in Fig. 9, layer 2. In the example, layer 3 is the SNR enhancement layer, note SNR scalability is also referred to as quality scalability. As has shown in the example, each spatial or SNR enhancement layer itself may be temporally scalable, with the same temporal scalability structure as the H.264/AVC compatible layer. Also, an enhancement layer can enhance both the spatial resolution and the frame rate. For example, layer 4 provides a 4CIF enhancement layer, which further increases the frame rate from 15 Hz to 30 Hz.

As shown in Fig. 9, the coded slices in the same time instance are successive in the bitstream order and form one access unit in the context of SVC. Those SVC access units

then follow the decoding order, which is different from the display order and decided e.g., by the temporal prediction relationship.

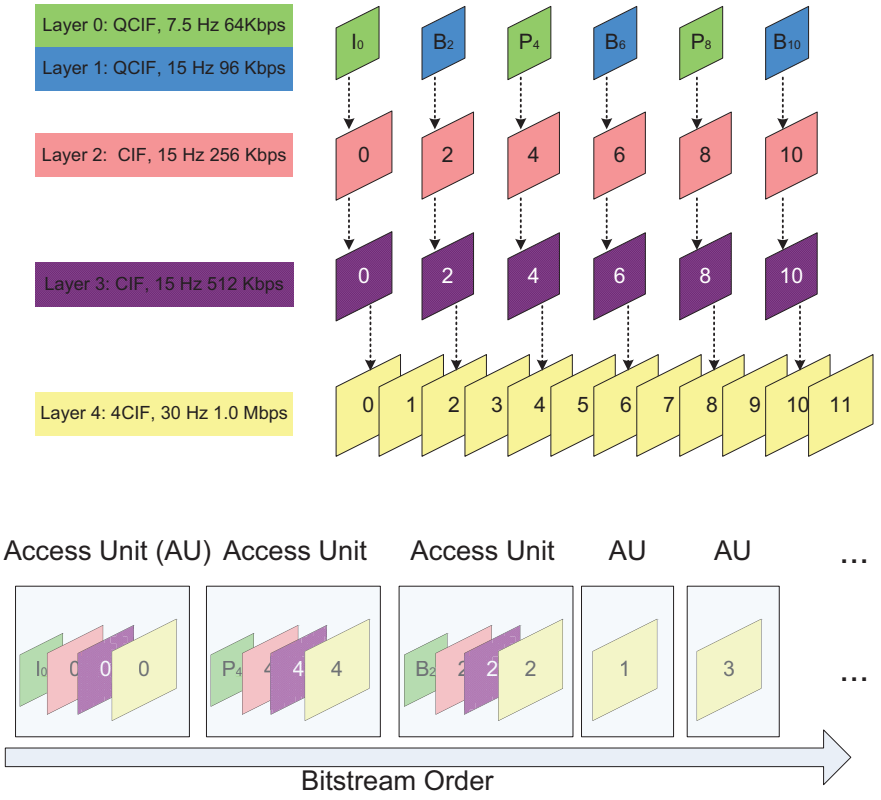


Fig. 9: Example structure of an SVC bitstream.

2.2. FEATURES OF SVC

Some functionalities of SVC are inherited from H.264/AVC. Compared with previous scalable standards, the most essential advantages, namely hierarchical temporal scalability, inter-layer prediction, single-loop decoding, and flexible transport interface are reviewed below. The reviewed features are necessary for understanding the mechanisms and algorithms proposed in this thesis.

2.2.1. Hierarchical Temporal Scalability

H.264/AVC provides flexible hierarchical B picture coding structure which enables advanced temporal scalability [47]. With this feature inherited from H.264/AVC, SVC supports temporal scalability for layers with different resolutions [48]. In SVC, a Group Of Pictures (GOP) consists of a so-called key picture, and all pictures which are located in the output/display order between this key picture and the previous key picture. A key picture is coded in regular or irregular intervals, which is either intra-coded or inter-coded using the previous key picture as a reference for motion compensated prediction. The non-key pictures

are hierarchically predicted from the pictures with lower temporal levels. The temporal level of a picture is indicated by the syntax element temporal_id in the NAL unit header SVC extension [39].

2.2.2. Inter-layer Prediction

SVC introduces inter-layer prediction for spatial and SNR scalabilities based on texture, residue and motion. The spatial scalability in SVC has been generalized to any resolution ratio between two layers [48]. The SNR scalability can be realized by Coarse Granularity Scalability (CGS) or Medium Granularity Scalability (MGS) [48]. In SVC, two spatial or CGS layers belong to different dependency layers (indicated by dependency_id in NAL unit header [39]), while two MGS layers can be in the same dependency layer. One dependency layer includes quality layers with quality_id [39] from 0 to higher values, corresponding to quality enhancement layers. In SVC, inter-layer prediction methods are utilized to reduce the inter-layer redundancy. They are briefly introduced in the following paragraphs.

Inter-layer texture prediction

The coding mode using inter-layer texture prediction is called “IntraBL” mode in SVC. To enable single-loop decoding [49], only the MBs, which have co-located MBs in the base layer coded as constrainedly intra modes, can use inter-layer textrure prediction mode. A constrainedly intra-coded MB is intra-coded without referring to any samples from the neighboring MBs that are inter-coded.

Inter-layer residual prediction

If an MB is indicated to use residual prediction, the co-located MB in the base layer for inter-layer prediction must be an inter MB and its residue may be upsampled according to the resolution ratio. The difference between the residue of the enhancement layer and that of the base layer is coded.

Inter-layer motion prediction

The co-located base layer motion vectors may be scaled to generate predictors for the motion vectors of MB or MB partition in the enhancement layer. In addition, there is one MB type named base mode, which sends one flag for each MB. If this flag is true and the corresponding base layer MB is not intra, then motion vectors, partitioning modes and reference indices are all derived from base layer.

2.2.3. Single-Loop Decoding

The single-loop decoding scheme in SVC is revolutionary compared with earlier scalable coding techniques. In the single-loop decoding scheme, only the target layer needs to be motion compensated and fully decoded [49]. Therefore, compared with the conventional multiple-loop decoding scheme, where motion compensation and full decoding are typically

required for every spatial or SNR scalable layer, decoding complexity as well as the DPB size can be greatly reduced.

2.2.4. Flexible Transport Interface

SVC provides flexible systems and transport interface designs that enable seamless integration of the codec to scalable multimedia application systems. Other than compression and scalability provisioning, systems and transport interface focuses on codec functionalities, such as, for video codec in general, interoperability and conformance, extensibility, random access, timing, buffer management, as well as error resilience, and for scalable coding in particular, backward compatibility, scalability information provisioning, and scalability adaptation. These mechanisms are augmented by the SVC file format extension to the ISO Base Media File Format [50] and Real-time Transport Protocol (RTP) payload formats [51]. Discussions of these SVC systems and transport interface designs can be found from [50], [51] and [52].

2.3. APPLICATION SCENARIOS FOR SVC

Typical application scenarios for scalable video coding are shown in Fig. 10. Note that, in this figure, only spatial and temporal scalabilities are shown. However, the scenarios for spatial scalability are also valid for SNR scalability. In practice, those scenarios may exist in different systems with different contents, network structures and receiving devices.

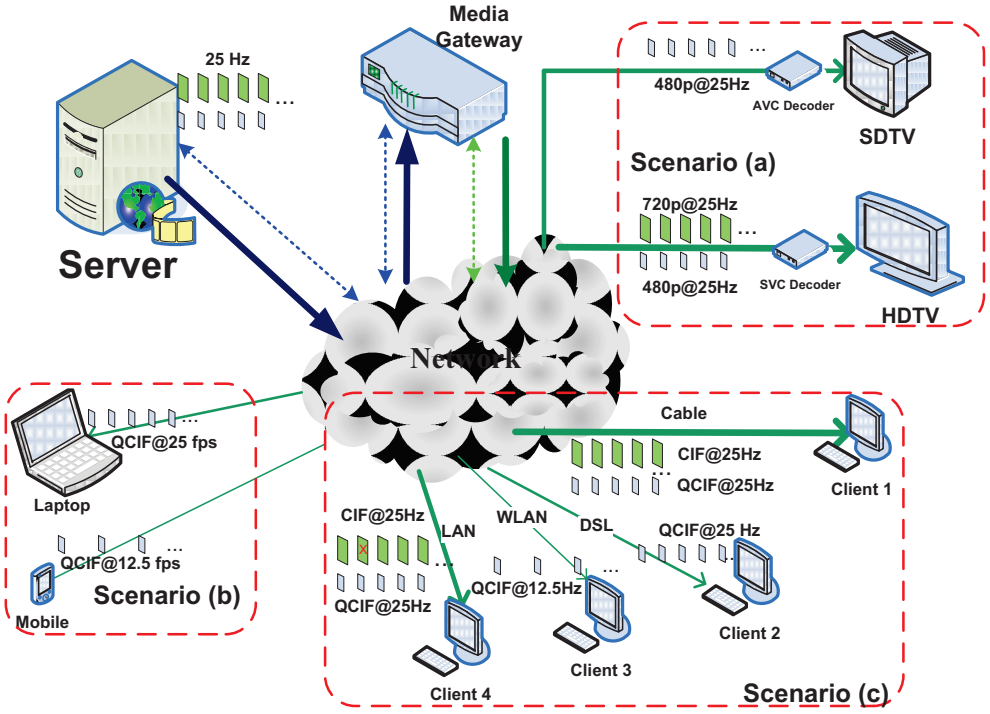


Fig. 10: Scalable video coding application scenarios.

Due to various levels of decoding capability, videos with different spatial resolutions, e.g. for a Standard Definition TV (SDTV) set and a HDTV set, can be decoded as shown in Scenario (a), or videos with different picture rates, e.g., for a mobile device and a laptop, can be decoded as shown in Scenario (b).

In scenario (a), a backward compliant solution for HDTV is illustrated, wherein an SVC bitstream can be delivered to different clients in e.g., Digital Video Broadcasting System (DVBS). The standard TV can receive and decode only the base layer, which e.g. is coded with H.264/AVC main profile from 480p (640x480 progressive) video content. HDTV, however, can receive both layers and output an e.g., 720p (1280x720 progressive) video. This spatial scalability shown here is also applicable for storage of the layered content, e.g., in DVDs.

In scenario (b), there are different devices with different decoding and rendering capabilities, in terms of computational capability, memory capacity and displayer size. So, even in an environment with good Quality of Service (QoS) and thus no packet loss, the mobile device will only process a sub-bitstream with a lower resolution e.g., QVGA and a lower frame rate, e.g. 12.5 fps, while the laptop can receive and decode a bitstream which contains a video representation of e.g. 25 fps VGA. As shown in this scenario, device adaptation in a broadcasting or a multicasting system is achieved by temporal and/or spatial scalability.

Clients can be the same but within different sub-networks or with different connections, e.g. in Scenario (c). Clients are connected with Cable, Local Area Network (LAN), Digital Subscriber Line (DSL), or Wireless LAN (WLAN). They can also be located in the same network but with different QoS, e.g. different congestion control methods applied by the intermediate nodes. Therefore, the expected bandwidth for each client may be different, which will lead to various received videos combined with different picture rates, spatial resolutions, and/or quality levels. Even for one client, due to bandwidth fluctuation, the received video may change at any moment in picture rate, spatial resolution, and quality level.

Some clients, such as client 4 in scenario (c), may try to receive more packets although the congestion will lead to frequent packet losses, especially in the enhancement layers. If no action is taken to protect against errors caused by packet losses, they will greatly exacerbate due to inherence of video coding e.g., predictive coding.

2.4. HIERARCHICAL P PICTURE CODING

Coding H.264/AVC compliant to the baseline profile usually leads to lower efficiency, not only because it does not support B pictures or CABAC, but also because IPPP… coding structure is typically used. As the baseline profile of H.264/AVC is important and the only supported profile in many application standards, such as mobile multimedia services specified by the Third Generation Partnership Project (3GPP), it is important to improve the baseline coding efficiency.

Therefore, in [P1], to investigate H.264/AVC coding when only intra (I) and inter (P) slices are supported, a content-adaptive QP cascading scheme for the hierarchical P coding method compatible with Baseline profile of H.264/AVC is proposed. Besides the high efficiency of the proposed method, temporal scalability is also provided with hierarchical B coding structure.

The proposed method is based on a picture-level QP optimization. The proposed method has a significantly better rate-distortion performance than the traditional IPPP… coding structure and outperforms hierarchical P coding methods using fixed delta QP settings between temporal levels noticeably with up to 0.53 dB gain in average luminance PSNR [P1].

The hierarchical B coding structure has been demonstrated as an effective tool for improving coding efficiency, e.g. compared with the traditional IBBP coding structure [53]. In this structure, the importance of pictures at each temporal level differs because of the hierarchically structured temporal prediction chain. Therefore, improved bitrate saving under the same quality constraint can be achieved by using higher QP values for higher temporal levels. In [53], Schwarz et al. proposed a QP setting method which fixed the difference of QP between each pair of temporal levels without considering the content characteristics change across sequences or pictures. The above method uses picture-level QP optimization, which selects one constant QP value for the whole slice.

One example of the hierarchical P coding structure (with 4 hierarchical/temporal levels) is shown in Fig. 11. The first picture of a video sequence is an IDR picture. A picture is called a key picture when all previously coded pictures also precede that picture in display order. As illustrated in Fig. 11, a key picture and all pictures that are temporally located between the current key picture and the previous key picture are considered as a GOP. In the hierarchical P coding structure, multiple references in inter prediction are supported. The prediction relationship is as follows. The key pictures are either intra-coded or inter-coded using previous key pictures as references. The remaining pictures of a GOP are hierarchically predicted and it is possible to use pictures from the past and/or from the future in display order as references. Referring to Fig. 11, picture 1 refers to pictures 0 and 2 which means that the inter prediction reference of each MB or MB partition is either from picture 0 or picture 2; however, any MB or MB partition cannot be simultaneously predicted from both pictures 0 and 2, as only one reference picture list (list 0) is constructed.

Let T be the total number of temporal levels. The QP value used for the coding of the pictures in the highest temporal level, denoted as the QP_{T-1}, is an input parameter, which can be set according to the desired target bitrate. The QP value for (coding) a lower temporal level picture is set according to a delta value to the QP_{T-1}. This delta value is decided by a scaling-factor, which depends on the prediction modes of the MBs in the picture.

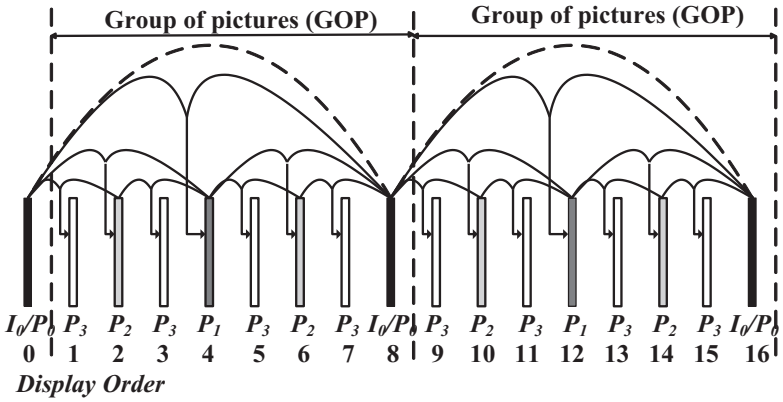


Fig. 11: Dyadic hierarchical P coding with 4 temporal levels.

In the hierarchical temporal prediction structures, the motion-compensation prediction can be expressed as the linear high-pass filtering along the motion trajectory with filter coefficients {1, -1} when using inter prediction, or with coefficients {-1/2, 1, -1/2} when using bi-prediction. The scaling-factor is to balance the residual energies of the whole picture in contrast to the energy of the pictures in a higher temporal level, and thus controls the QP value of the picture. The scaling-factor is derived as the weighted average of the relative energy increase caused by the filtering process which is actually performed during inter predicted motion compensation prediction [P1]. After motion estimation, the energy factor of a picture *m* with a temporal level *t* can be calculated as in equation (2):

$$E_{t,m} = \frac{1}{N}\sum_{i=1}^{N} \alpha_i \qquad (2)$$

where *t*, ranging from 0 to *T-1*, inclusive, denotes the temporal level, *m* denotes the picture index within a temporal level t, N is the total number of MBs in a picture, and α_i represents the weighting factor of the relative energy of the i-th MB during motion compensation. Note that for simplicity, it is assumed that all MB partitions (if more than one) in an MB are treated with the same intra, inter-P or inter-B modes.

The weighting factor α_i of an MB depends on the prediction mode of the MB, as shown in equation (3):

$$a_i = \begin{cases} 1 & Intra \\ \sqrt{2} & Inter\text{-}P \\ \sqrt{3/2} & Inter\text{-}B \end{cases} \qquad (3)$$

If an MB is an inter-P MB, the temporal filtering is with a filter of {1, -1}. If it is an inter-B MB, the filter coefficients are {-1/2, 1, -1/2}). In the Baseline profile, there is no inter-B MB or MB partition. For each MB, if it is not intra-coded, all its MB partitions must be inter-P.

After $E_{t,m}$ is obtained, the corresponding scaling-factor of the m-th picture with temporal level t is:

$$SF_{t,m} = \overline{SF_{t+1}} \Big/ E_{t,m} \quad and \ \overline{SF_{t+1}} = \sum_m SF_{t+1,m} / \sum_m 1, t = 0,1 \ldots T-1, and \ \overline{SF_T} = 1 \qquad (4)$$

where $\overline{SF_{t+1}}$ is the average $SF_{t+1,m}$ of all pictures with temporal level t+1 in the current GOP.

After the scaling-factor $SF_{t,m}$ is obtained, the QP value for the corresponding picture, denoted as $QP_{t,m}$, can be calculated as in equation (5):

$$QP_{t,m} = QP_{T-1} + 6\log_2 SF_{t,m} \qquad (5)$$

where QP_{T-1} is the input QP value and $6\log_2 SF_{t,m}$ is the delta QP value. The final value of $QP_{t,m}$ is rounded (and clipped if needed) to be an integer value in the range of 0 to 51, inclusive. Note that all the highest temporal level pictures have the same QP value in our method.

2.5. BIT-DEPTH SCALABILITY

Besides, spatial resolution, frame rate and SNR quality, enhancements of the video representations in other dimensions are also desirable. Two examples are color bit-depth scalability and chroma sample format scalability, as in the future, professional applications will require high chroma sample format such as 4:4:4 and high bit-depth, e.g., up to 16 bits. In this section, bit-depth scalability coding is presented.

2.5.1. Architecture of Bit-depth Scalability Coding

While eight-bit playback and display devices will be dominating the market in the near future, superior visual quality by high bit-depth videos is desirable for applications such as high standard entertainment and healthcare. Hence, conventional eight-bit and high bit-depth digital imaging systems will coexist in the market. Content distributors supporting both formats need to provide different contents for different users, e.g., independently code the different representations for the same video content. This requires more storage or bandwidth for video content delivery.

Bit-depth scalability is an efficient tool to solve this problem. However, video coding techniques can allow flexible usage of various versions of the same visual content that may have spatial resolutions and even alterations in color.

A bit-depth scalable coding solution compatible with SVC was thus proposed in [P2]. The solution is capable of providing an 8-bit AVC main profile or high profile compliant base layer which is multiplexed with a high bit depth (e.g., 10-, 12-, or up to 14-bit) enhancement layer through MB level inter-layer bit-depth prediction.

As shown in Fig. 12, the generated scalable bitstream can be decoded by a scalable decoder and the reconstructed video can be sent to a high-end display. The base layer sub-bitstream, however, can be decoded by e.g., an H.264/AVC high profile decoder and sent to a standard or a high definition TV.

New decoding processes for inter-layer bit-depth prediction are introduced to enable bit-depth scalability [P2]. Combinations with other types of scalability: temporal, spatial and SNR scalability, as well as single-loop decoding are also supported since the algorithm is

developed based on the SVC standard. Furthermore, the solution supports adaptive inter-layer prediction to determine whether or not the inter-layer bit-depth prediction shall be invoked for each MB. The presented experimental results show that the bit-depth scalability solution outperforms simulcast coding significantly [P2].

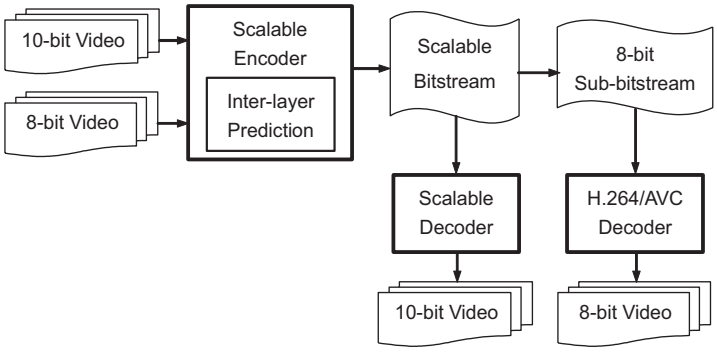


Fig. 12: Diagram of bit-depth scalable coding.

2.5.2. Discussion

When a bit-depth scalable video stream is generated, the source video content is typically of high-bit depth and the 8-bit video needs to be generated. Tone mapping techniques could be used to convert the high-bit video to 8-bit [54]. Although good tone mapping techniques providing better display quality in legacy 8-bit displays are preferred, linear tone mapping, e.g., by chopping the least significant bits, is also acceptable. The way 8-bit video is generated actually impacts the coding performance of the proposed video codec.

However, as according to the results in [P2], no matter what tone mapping technique is used, significant coding gain can be obtained, varying from 0.35 dB to 1.1 dB, although the best coding performance is achieved in the linear tone mapping case.

Objectively, the proposed scalable codec has good RD performance, but its subjective quality has not been investigated sufficiently. The quality comparison between the high-bit enhancement layer video of the scalable bitstream and the high bit-depth video coded directly by the H.264/AVC FREXt should be investigated further.

2.6. OTHER SCALABILITIES

In contrast to the bit-depth scalability, the chroma format scalability is relatively straightforward. A typical scenario is to enhance the traditional 4:2:0 video to 4:4:4 video. As shown in Fig. 13, in 4:2:0 chroma sample format, the chroma samples are with half the sampling rate in both the horizontal and vertical directions. Note a chroma sample can have different relative positions of the nearby luma samples, however, in H.264/AVC, the relative position is the same as shown in Fig. 13. When chroma components have the same sampling rate as luma, the chroma sample format turns to 4:4:4. There is also a 4:2:2 sample format, wherein only in the vertical direction, chroma samples are sampled with half the sampling

rate, while in the horizontal direction, the sample rates for chroma and luma components are the same.

When only chroma sample scalability is required, the luma component does not require any extra enhancement. Thus only the resolution enhancement of the chroma sample is needed. This dyadic resolution enhancement can be realized by the tools adopted in SVC for spatial scalability.

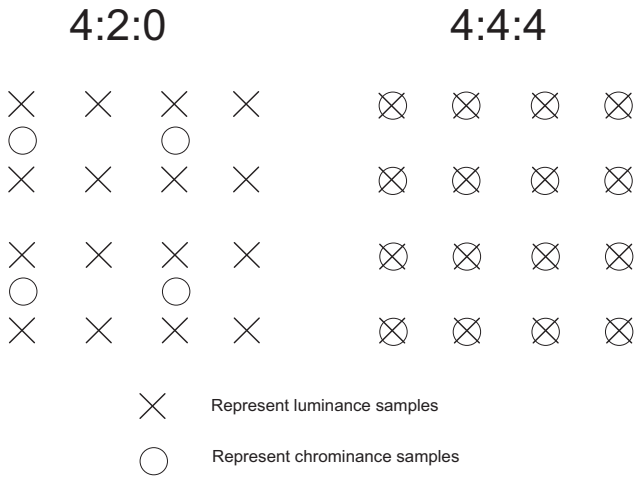


Fig. 13: Chroma sample format: 4:2:0 and 4:4:4.

Chapter 3

The Emerging Multiview Video Coding (MVC) Standard for 3D Video Services

3D video has gained a wide interest recently, especially in the standard committees. 3D applications, e.g., 3D TV, free-viewpoint video require a mature representation of 3D video. Multiview video is the one that represents a 3D scene with multiple views, each of which usually corresponds to a 2D digital video. Multiview/stereo video technologies are the most popular or feasible one in terms of capturing, transmission and display. Furthermore, with the advances in the acquisition and display technologies, 3D video is becoming a reality in the consumer domain with different application opportunities. Given a certain maturity of capture and display technologies and with the help of multiview video coding (MVC) techniques, a number of different envisioned 3D video applications are becoming feasible.

The huge amount of data needed to be processed by multiview video is a heavy burden for both transmission and decoding. The JVT has recently devoted part of its effort in extending the widely deployed H.264/AVC standard to handle multiview video coding (MVC), and MVC standard has been finalized [54] and the latest MVC standard, integrated with H.264/AVC and SVC, is specified in [56]. The MVC extension of H.264/AVC includes a number of new techniques for improving coding efficiency, reducing decoding complexity, and new functionalities for multiview operations. MVC takes advantage of some of the interfaces and transport mechanisms introduced for the scalable video coding (SVC) extension of H.264/AVC, but the system level integration of MVC is conceptually more challenging as the decoder output may contain more than one view and can consist of any combination of the views with any temporal level. The generation of all the output views also requires careful consideration and control of the available decoder resources. In this chapter, multiview applications and solutions to support generic multiview as well as 3D services are introduced. As an important part of this thesis, solutions, which have been adopted into the MVC specification, cover a wide range of requirements for 3D video related

to interface, transport of the MVC bitstreams and MVC decoder resource management. The features that have been introduced in MVC to support these solutions include marking of reference pictures, support for efficient view switching, and structuring of the bitstream, view scalability SEI and other SEI messages.

MVC shares some design principles with SVC, such as backward compatibility with H.264/AVC, temporal scalability, network friendly adaptation, and many features in SVC have been reused in MVC. However, new mechanisms are needed in MVC, at least related to view scalability, inter-view prediction structure, coexisting of decoded pictures from multiple dimensions (i.e. both temporal and view dimensions) in the decoded picture buffer, multiple representations in the display, and parallel decoding at the decoder. These mechanisms cover the challenges and requirements, identified above, for 3D video services, except for the compression efficiency challenge. In this thesis, we describe how these mechanisms are realized in the existing MVC standard. The main MVC features discussed in this thesis include reference picture management to achieve optimal memory consumption at the decoder; time-first coding to support consistent system level design; SEI messages and other features for view and scalability information signaling, adaptation, random access, view switching, and reference picture list construction.

In this Chapter, MVC applications are introduced within a given MVC system architecture as described in Section 3.1; then the MVC requirements are reviewed in Section 3.2. The MVC features, including the thesis contributions, designed for the requirements are described in the following sections.

3.1. SYSTEM ARCHITECTURE FOR MVC

3D video applications can be grouped under three categories, free-viewpoint video, 3D TV, and immersive teleconferencing. The requirements of these applications are quite different and each category has its own challenges to be addressed [57].

To illustrate these challenges, consider Fig. 14, where the end to end architecture of different applications is shown [P3]. In this illustration, a multiview video is first captured and then encoded by a multiview video coding (MVC) encoder. A server transmits the coded bitstream(s) to different clients with different capabilities, possibly through media gateways. The media gateway is an intelligent device, also referred to as a Media Aware Network Element (MANE), which is in the signaling context and may manipulate the incoming video packets (rather than simply forward packets). At the final stage, coded video is decoded and rendered with different means according to the application scenario and capabilities of the receiver. To provide smoothly immersive experience when a user adjusts the viewing position, view synthesis [58][59] may be required at the client to generate “virtual” views of a real-world scene. However, till now, this process is out of the scope of any existing coding standard.

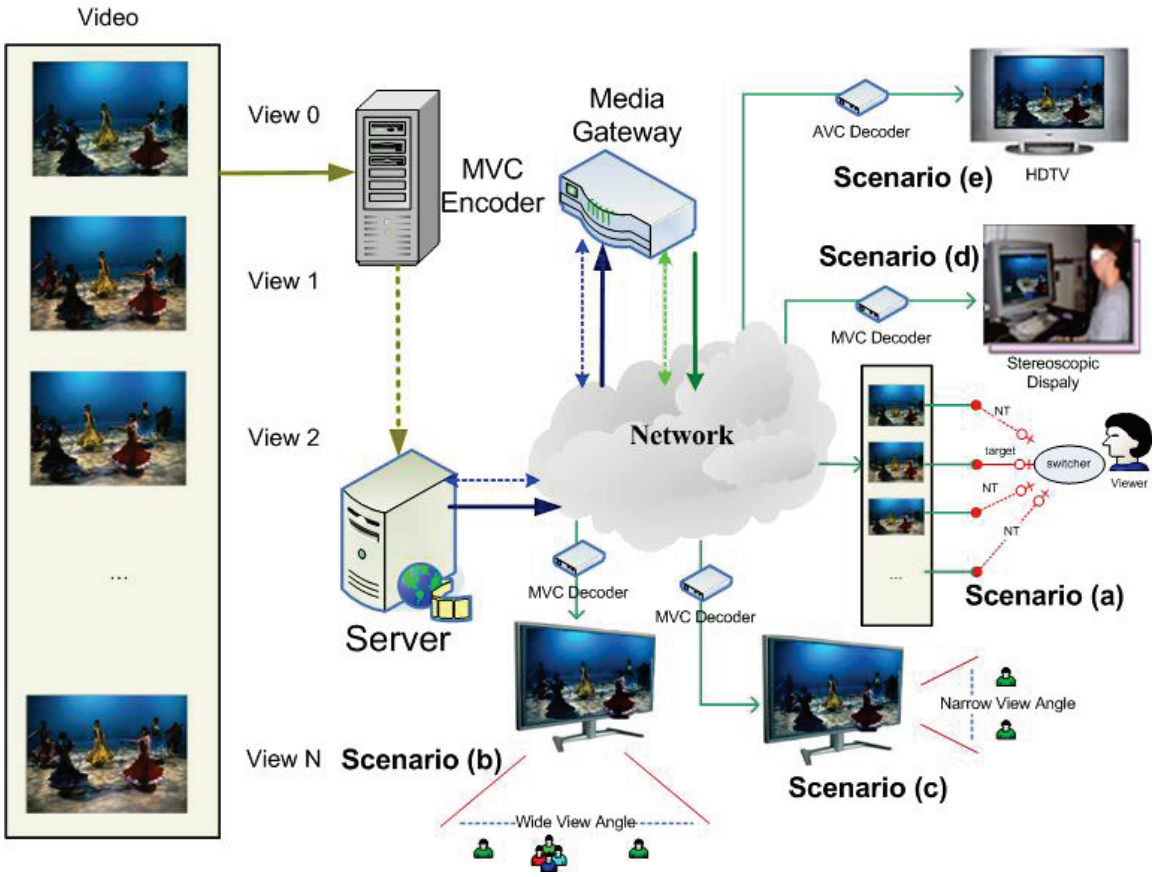


Fig. 14: MVC system for different application scenarios.

In free-viewpoint video, the viewer can interactively choose the viewpoint in 3D space to observe a real-world scene from preferred perspectives [60]. It provides realistic impressions with interactivity, i.e., the viewer can navigate freely in the scene within a certain range, and view the 3D scene from different viewing angles. Such a video communication system has also been reported in [61]. Unlike holography, which generates 3D representation and requires changing of the relative geometry position of a viewer to switch the view point, this scenario is actually realized by switching between rendered view(s) using an interface such as remote controller. In case the desired viewpoint is not available, interpolating a virtual view from other available views. Scenario (a), in Fig. 14, illustrates this application, where there exists several candidate views for the viewer, and one of them is selected as the target view that is displayed (views that are not targeted and thus are not outputted are denoted as "NT" for simplicity in Fig. 14). In this scenario, not all the candidate views are required to be decoded, thus the decoder can focus its resources only on decoding of the target view. For this purpose, the target view need to be efficiently extracted from the bitstream and thus only the packets that are required for successfully decoding the desired views are transmitted. To enable navigation in a scene, important functionality to be achieved by the system is efficient switching between different views. The related solution was described in [P3] and elaborated in Section 3.5.

3D TV refers to the extension of traditional 2D TV displays to displays capable of 3D rendering. In this application, more than one view is decoded and displayed simultaneously [62]. A simple 3D TV application can be realized by stereoscopic video. Stereoscopic display can be achieved by using special data glasses (e.g., shuttle glasses or polarization glasses) or other means. However, it is nicer for the user to get the 3D feeling directly through a 3D display with added the feature of rendering binocular depth cues [63], which can be realized by auto-stereoscopic displays. Advanced auto-stereoscopic displays can support head-motion parallax, by decoding and displaying multiple views from different view-points simultaneously. That is, a viewer without extra facilities such as glasses, can move to different geometry angle ranges, each of which contains typically two views rendered and shed by 3D displays. 3D TV displays are discussed in [64]. The viewer can then experience a slightly different scene by moving his/her head (for example the user may look what is behind a certain object in the scene). In this scenario, multiple views need to be decoded simultaneously; and therefore, parallel processing of different views is very important to realize this application. In addition, displaying multiple views is important also to realize a wide viewing angle as shown in Fig. 14 (b). This scenario is also referred to as auto-stereoscopic 3D TV for multiple viewers [63]. However, if the decoder capability is limited or the transmission bandwidth decreases, the client at a receiver may simply decode and render just a subset of the views but still provide 3D display with a narrow view angle, as shown in Fig. 14 (c). The media gateway plays an important role to provide the adaptation functionality to support this use case. Such a 3D TV broadcast or multicast system must then support flexible stream adaptation. As summarized in [P3], stream adaptation can be achieved at the server or media gateway, where only the sub-bitstream desired by the client and having the appropriate bandwidth is transmitted and other packets are discarded. After bitstream extraction, the sub-bitstream must be decodable by an MVC decoder. Detailed use cases and solutions for stream adaptation and extraction is given in Section 3.3.

Free-viewpoint video focuses on its functionality in free navigation of a 3D scene while displaying only one 2D video of a view at each specific time period. 3D TV, however, emphasizes on 3D immersive experience based on binocular perspective. In an immersive teleconference, both interactivity and virtual reality may be preferred by participants and thus free viewpoint or 3DTV style can both be supported. In the immersive teleconferencing, where there is interactivity among viewers, immersiveness can be achieved either in a free-viewpoint video or 3D TV manner.

Typically two mechanisms can make people perceptually feel immersed in a 3D environment. A typical technique, known as Head-Mounted Display (HMD) needs a device worn on the head, as a helmet, which has a small display optic in front of each eye. An easier solution for stereo video, however, is to use glasses, to filter the views and guarantee the left/right view reaches only the left/right eye of the viewer. A filter with such a functionality can also be built in the screen of the display, similar to the display techniques for 3D TVs. This scenario is shown in Fig. 14 (d). Other substitutions for HMD or glasses need to introduce head tracking [65] or gaze tracking [66] techniques, as shown in the solutions

discussed in [63]. In 3D TV, however, each stereoscopic display can have effect on a certain small range of a view angle, thus, a viewer can change his/her viewing position when he/she is trying to view the scene in another viewpoint, as if there was a natural object.

For rendering 3D TV content or view synthesis, depth information is needed. Depth-images storing the depth information as a monoscopic color video can be coded with existing coding standards, for example, as auxiliary pictures in H.264/AVC [3].

As the normal 2D TV or High Definition TV (HDTV) applications are still dominating the market, MVC content shall provide a way for those 2D decoders, e.g. H.264/AVC decoder in the STBs (Set-Top Box) of digital TV to generate a display from an MVC bistream, as shown in Fig. 14 (e). This requires MVC bitstreams to be backward compatible, e.g. to H.264/AVC. So part of an MVC bitstream needs have a sub-bitstream, namely base view, understandable by H.264/AVC decoder but also have characteristics of the base view which are usually required for successful decoding of the other views. The design of new NAL unit and the NAL unit header in the MVC context [P4], make the above scenario feasible. The design of the MVC NAL units, as well as other basic aspects of the MVC bitstreams therefore is reviewed in Section 3.3.

3.2. REQUIREMENTS OF MULTIVIEW VIDEO CODING

The requirements of these 3D video applications are quite different and each category has its own challenges to be addressed.

Due to the huge amount of data, particularly when the number of views to be decoded is large, transmission of multiview video applications relies heavily on the compression of the video captured by cameras. Therefore, efficient compression of multiview video contents is the primary challenge for realizing multiview video services.

A natural way to improve compression efficiency of multiview video content is to exploit the correlation between views, in addition to the use of inter prediction in mono-view coding. This requires buffering of additional decoded pictures. When the number of views is large, the required memory buffer may be prohibitive. In order to make efficient implementations of MVC feasible, the codec design should include efficient memory management of decoded pictures.

The above challenges and requirements, among others [67], are the basis of the objectives for the emerging MVC standard, which is under development by the JVT, and will become the multiview extension of H.264/AVC [3]. MVC standardization in the JVT started in July 2006 and was finalized in mid-2008. The most recent draft of MVC is available in [56]. The contribution of the thesis for MVC includes techniques proposed to MVC to meet the MVC requirements, such as mechanisms for bitstream adaptation, random access and view switching and decoder memory management.

In the MVC standard, redundancies among views are utilized to improve compression efficiency compared to independent coding of views. This is made possible with the so-called inter-view prediction, in which decoded pictures of other views can be used as

reference pictures when coding a picture as long as they all share the same capturing or output time. View dependencies for inter-view prediction are defined for each coded video sequence.

With the exception of inter-view prediction, pictures of each view are coded with the tools supported by H.264/AVC. In particular, hierarchical temporal scalability was found to be efficient for multiview coding [47]. A typical prediction structure of MVC, utilizing both inter-view prediction and hierarchical temporal scalability, is shown in Fig. 15, wherein 8 views are coded and the GOP size is 8. S0 corresponds to the base view (will be defined later), and other views are non-base views. In each view, hierarchical B temporal prediction structure is used thus the pictures in one view are with different temporal levels. It is noted that the MVC standard provides a greater deal of flexibility than depicted in Fig. 15, in terms of GOP structure and arrangement for temporal or view prediction references.

Except the coding efficiency requirement, the following important aspects of the MVC requirements for the design of the MVC standard are listed.

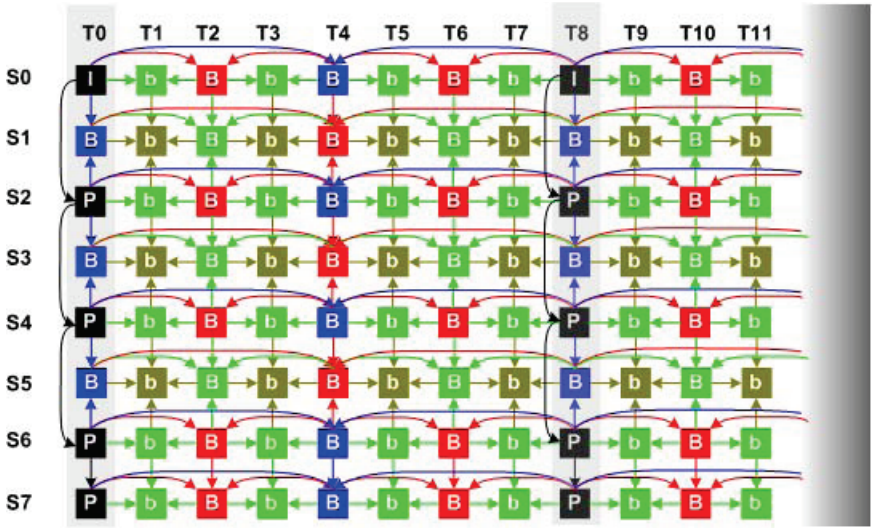


Fig. 15: Typical MVC prediction structure.

Scalabilities

View scalability and temporal scalability are considered in the MVC design for adaptation of user preference, network bandwidth, and decoder complexity. View scalability is useful in the scenario shown in Fig. 14 (c), wherein some of the views are not transmitted and decoded.

Decoder resource consumption

In 3D TV scenarios, as shown in Fig. 14 (b) and (c), a number of views are to be decoded and displayed, an optimal decoder in terms of memory and complexity is of vital importance to make the real-time decoding of MVC bitstreams possible.

Random access

Besides temporal random access, view random access is to be supported to enable accessing a frame in a given view with minimal decoding of frames in the view dimension. For example, free-viewpoint video described in Fig. 14 (a) needs advanced view random access functionality to support smooth navigation.

Robustness

When transmitted in a lossy channel, the MVC bitstream shall have error resiliency capabilities. There are error resilient tools in H.264/AVC which can benefit the MVC applications. Other techniques, which are designed only for MVC and discussed later, can also be utilized to improve error resilience of MVC bitstreams.

Parallel processing

In 3D TV scenarios, since multiple views need to be decoded simultaneously, parallel processing of different views is very important to realize this application and to reduce the computation time to achieve real-time decoding.

The contribution to MVC, as part of the thesis work, is related to the previous requirements except the parallel processing. Most of the contributions in this part were originally proposed by the author and other researchers in the proposals to JVT standard. They are also summarized in [P3]

3.3. STRUCTURE OF MVC BITSTREAMS

This section reviews the concept of Network Abstraction Layer units (NAL units) and summarizes how the NAL unit types defined in H.264/AVC and SVC are reused for MVC. Some syntax elements in the NAL unit header in the MVC context are also discussed.

In H.264/AVC, the coded video bits are organized into NAL units. NAL units can be categorized into VCL NAL units and non-VCL NAL units. The supported VCL NAL unit types and non-VCL NAL units in H.264/AVC are defined in [3] and well categorized in [17].

In MVC, there is a base view, which is coded independently and is compliant with H.264/AVC, this meets the requirement in Scenario (e) shown in Fig. 14. Consequently, the coded picture information for the base view is included in the VCL NAL units specified in H.264/AVC. A new NAL unit type, called coded slice of MVC extension, is used and contains coded picture information for non-base views. When an MVC bitstream containing NAL units of the new NAL unit type is fed to an H.264/AVC decoder, NAL units of any new NAL unit type can be ignored and the decoder only decodes the bitstream subset containing NAL units of the existing NAL unit types defined in H.264/AVC.

There are useful properties of the coded pictures in the H.264/AVC-compliant base view, such as temporal level, which are not indicated in the VCL NAL units of H.264/AVC. To

indicate those properties for the base view coded pictures, the prefix NAL unit, of another new NAL unit type, has been introduced. Note that prefix NAL unit is also specified in SVC. A prefix NAL unit precedes each H.264/AVC VCL NAL unit and contains its essential characteristics in multiview context. As H.264/AVC decoders ignore prefix NAL units, the backward compatibility to H.264/AVC is still maintained.

Non-VCL NAL units include parameter set NAL units and SEI NAL units, among others. Parameter sets contain the sequence-level header information (in sequence parameter sets—SPS) and the infrequently changing picture-level header information (in picture parameter sets—PPS). With parameter sets, this infrequently changing information needs not to be repeated for each sequence or picture, hence coding efficiency is improved. Furthermore, the use of parameter sets enables out-of-band transmission of the important header information, avoiding the need of redundant transmissions for error resilience. In out-of-band transmission, parameter set NAL units are transmitted on a different channel than the other NAL units. More discussions on parameter sets can be found in [8].

In MVC, coded pictures from different views may use different sequence parameter sets (SPSs). An SPS in MVC can contain the view dependency information for inter-view prediction. This enables signaling-aware media gateways to construct the view dependency tree, which describes the relationship hierarchy of the views. Therefore, each view can be mapped to the view dependency tree and view scalability can be fulfilled, without any extra signaling inside NAL unit headers [P3].

The scalable nesting SEI message [17], which was also introduced in SVC with the same name, is set apart from other SEI messages in that it contains one or more ordinary SEI messages, but in addition it indicates the scope of views or temporal levels for which the messages apply. In doing so, it enables the reuse of the syntax of H.264/AVC SEI messages for a specific set of views and temporal levels.

Some of the other SEI messages specified in MVC are related to the indication of output views, available operation points and information for parallel decoding.

In H.264/AVC, a NAL unit consists of a 1-byte header and a payload of varying size. In MVC, this structure is retained except for prefix NAL units and MVC coded slice NAL units, which consist of a 4-byte header and the NAL unit payload. New syntax elements in MVC NAL unit header include priority_id, temporal_id, anchor_pic_flag, view_id, non_idr_flag and inter_view_flag.

anchor_pic_flag indicates whether a picture is an anchor picture or non-anchor picture. Anchor pictures and all the pictures succeeding it in the output order (i.e. display order) can be correctly decoded without decoding of previous pictures in the decoding order (i.e. bitstream order) and thus can be used as random access points. Anchor pictures and non-anchor pictures can have different dependencies, both of which are signaled in the sequence parameter set. Other flags are to be discussed and used in the following sections of this chapter.

The following sections of this chapter introduce the MVC features which were designed to meet the requirements mentioned in the above. The thesis's contributions for supporting those features will also be mentioned, when those sections are introduced,

3.4. EXTRACTION AND ADAPTATION OF MVC BITSTREAMS

MVC supports temporal scalability and view scalability. A portion of an MVC bitstream can correspond to an operation point that gives an output representation for a certain frame rate and a number of target views [P3], this part of MVC design is to meet the first requirement described in 3.2, scalabilities.

Data representing a higher frame rate, views closer to the leaves of the dependency tree or views that are not preferred by the client, can be truncated during the stream bandwidth adaptation at the server or media gateway, or ignored at the decoder for complexity adaptation.

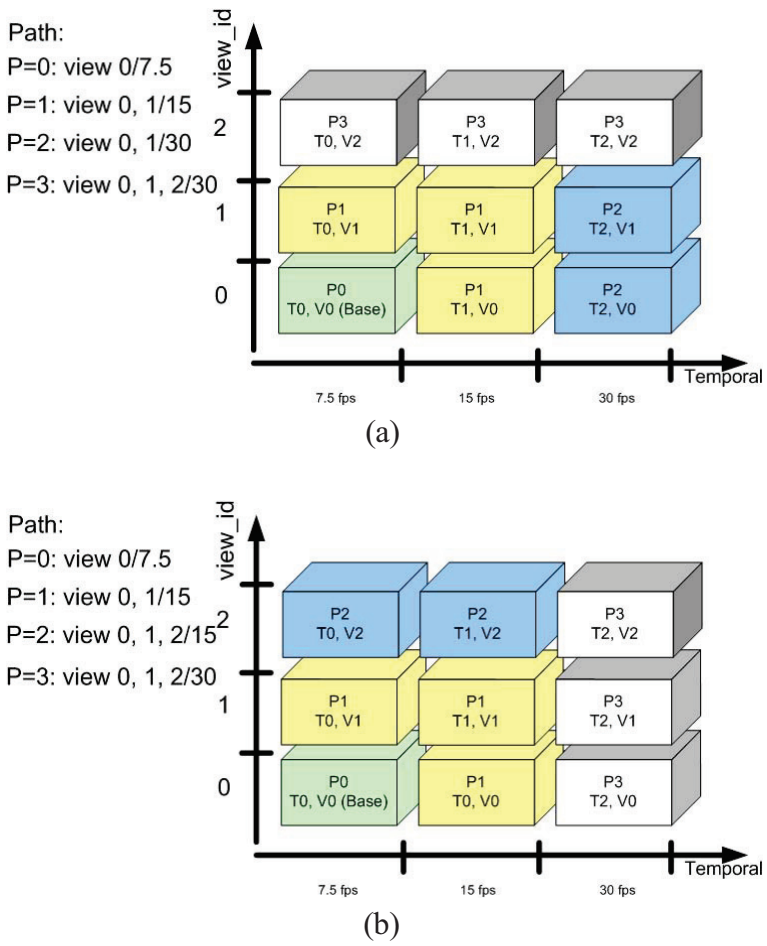
The bitstream structure defined in MVC is characterized by two syntax elements: view_id and temporal_id. The syntax element view_id indicates the identifier of each view. This indication in NAL unit header enables easy identification of NAL units at the decoder and quick access of the decoded views for display. The syntax element temporal_id indicates the temporal scalability hierarchy or, indirectly, the frame rate. An operation point including NAL units with a smaller maximum temporal_id value has a lower frame rate than an operation point with a larger maximum temporal_id value. Coded pictures with a higher temporal_id value typically depend on the coded pictures with lower temporal_id values within a view, but never on any coded picture with a higher temporal_id.

The syntax elements view_id and temporal_id in the NAL unit header are important for both bitstream extraction and adaptation. Another important syntax element in the NAL unit header is priority_id [P3], which is mainly used for the simple one-path bitstream adaptation process.

Whenever the operation point contains only a subset of the entire MVC bitstream, such as in Scenario (a) and Scenario (c) shown in Fig. 14, a bitstream extraction process is then needed to exact the required NAL units from the entire bitstream. The bitstream extraction process should be a lightweight process without heavy parsing of the bitstream. For this purpose, the mapping between each operation point (identified by the combination of required view_id values and temporal_id values) and the required NAL units is specified as part of the view scalability information SEI message (VSSEI) [P3]. After the operation point is agreed upon, the server can simply extract the required bitstream subset by discarding non-required NAL units by checking the view_id and temporal_id values in the fixed-length coded NAL unit headers.

Media gateways can perform single-path adaptation by simply discarding NAL units with priority_id greater than a certain value. The priority_id has no normative effect to the decoding process. The only constraint to priority_id values is that any bitstream subset extracted based on any value of priority_id must be a conforming MVC bitstream. It is the

encoder's responsibility to set priority_id values for the NAL units and the values can be rewritten, e.g. when the preference of the decoder changes.



T: temporal level;
V: view identifier;
P: priority identifier

Temporal level equal to 0 corresponds to 7.5 fps (frame per second), 1 corresponds to 15 fps, and 2 corresponds to 30 fps

Fig. 16: Assignment of priority_id for NAL units of a 3-view bitstream with two levels of temporal resolution.

Fig. 16 depicts two examples of priority_id assignments which yield two different adaptation paths for the same MVC bitstream that contains 3 views with 3 temporal levels. In Fig. 16 (a), the priority_id is assigned such that, the 7.5 Hz base view is with priority_id equal to 0, and then frame rate of 15 Hz including both view 0 and view 2 is with priority_id equal to 1, and then higher frame rate is preferred to more views. In Fig. 16 (b), the first two steps are the same as in Fig. 16 (a), while in the last two steps, more views are preferred compared to a higher frame rate.

Although a simple media gateway may perform stream adaptation exclusively based on priority_id, more intelligent implementations may jointly employ the values of priority_id, view_id, and temporal_id, in order to perform combined adaptation. For example, for the bitstream discussed in Fig. 16, there can be two adaptation steps, the first step is to have NAL units with temporal_id equal to 1 (15 Hz) and view_id through 0 to 1; the second step

is to increase frame rate directly to 30 Hz and include all the NAL units in view 2. Note that in this case, the NAL units corresponding to each adaptation step can have different values of priority_id, for example, when the priority_id assignment follows Fig. 16 (a).

An MVC bitstream may contain a large number of views (the view_id in the current MVC draft specification is of 10 bits). This makes the possible number of combinations of view_id values and temporal_id values huge. However, in practical applications, typically only limited combinations, i.e. operation points, would be used. The VSSEI has been designed to be flexible to signal any subset of all the possible operation points. Beside the mapping of operation points and NAL units, the following information for each indicated operation point are also included in the VSSEI, either to enable establishment of the communication session or more efficient bitstream extraction or adaptation.

Profile and level: This information describes the capacity required by a decoder to decode a bitstream. Profile and level can be signaled in the SPS. However, the total number of SPS is limited to a certain value in the bitstream and it may happen that for all the operation points, many of them share the same SPS, the level inside which is not accurate enough to describe the minimum required capacity of the decoders for different operation points. Therefore, profile and level are signaled in the VSSEI for each operation point.

Bit rate: Similar to profile and level, this information is needed in the session negotiation process for the server and the client to agree upon a certain operation point. This information is also useful in rate adaptation by MANEs. For example, to better adapt the bandwidth, it is necessary for intelligent media gateways to know the bandwidth of a session when it switches to another operation point.

Operation point dependencies: In the VSSEI, each operation point is identified by the view_id values of the target views and the temporal_id values. The dependent views as well as the dependent pictures may be known from the active SPS which contains the view dependency information. However, within the view dependency, pictures may have more flexible relationship. For example, assume in a two-view bitstream with 30 fps, 4 temporal levels and according to the SPS MVC extension anchor pictures and non-anchor pictures in view 1 are respectively dependent on anchor and non-anchor pictures in view 0. And if we have two operation points (OP), OP 0 has the pictures in view 0 with temporal level up to 3, i.e., 15 fps and OP 1 has pictures with all the pictures in view 1, however, the pictures with the highest temporal level in view 1 do not really rely on inter-view pictures for reference. Then, OP 1 actually depends only on OP 0, which contains half of the pictures in view 0 and the highest temporal level pictures in view 0 can be neglected for transmission and decoding. However, with only the view dependency signaled in the SPS MVC extension, those pictures are still required to be transmitted and decoded. Thus, the operation point dependency information included in the VSSEI would enable simply identification and discarding of the non-required NAL units that are not indicated by the view dependency information signaled in SPS.

In the following are some MVC stream adaptation examples in a broadcasting system (see Fig. 14). Assume that the entire bitstream contains coded pictures of 8 views.

For Scenario e), NAL units are filtered by the MANE so that only the NAL units that can be recognized by H.264/AVC decoders (by checking the NAL unit type) are fed to the STB of an HDTV.

For Scenario d), an operation point containing e.g., only view 0 and view 1, is in use. The MANE controls the bitstream in a way that only allows the NAL units with view_id (by checking the view_id in the NAL unit header) equal to 0 or 1 to be sent to the client.

Depending on the bandwidth, a client with enough decoding capability for 3D TV may switch between Scenarios (b) and (c), wherein the sub-bitstream corresponding to Scenario (b) forms an operation point that contains only a subset of the views within a narrow view angle. The MANE filters out the views outside the view angle.

In MVC specification, a normative extraction process is also defined to get the views between two views in the bitstream order of each access unit. The extracted sub-bitstream is suitable for a certain computation and display capability, which typically corresponds to a level different from the level of the original bitstream.

3.5. RANDOM ACCESS AND VIEW SWITCHING

Random access operations include the accessing of a picture in a given view or the accessing of a new view. View switching can be enabled by the random access functionality to a view.

3.5.1. Random Access

Random access refers to starting decoding of a bitstream from a point other than the beginning. Support of random access is required for traditional trick play modes such as fast forward and fast backward. In streaming applications, random access is used to seek the desired playback position requested by the users. In broadcast and multicast applications, random access points are required to allow current users to switch program channels and newcomers to tune in on a desired program.

Random access with MVC for the above purposes is not much different from that with single-view coding, as all the target views of an operation point are accessed simultaneously. The only difference is that there may be views dependent on by the target views; hence these dependent views need also to be accessed and decoded.

To access a picture in a given view at a specified time, the decoder should first find the closest preceding temporal location that are random access points to the specific target view and all the dependent views, collectively referred to as the required views. Then the decoder starts decoding the required views from the found location. On average, the random access period (i.e. the length of the temporal dependency chain) is proportional to the number of view pictures that need to be decoded to access a specific target picture and the number of dependent views (i.e. the length of the inter-view dependency chain).

IDR pictures are natural random access points. In an MVC bitstream, IDR pictures in the base view have NAL units of type 5. If the bitstream also contains NAL units that are

unknown to plain H.264/AVC decoders, then the base view IDR picture NAL units are each preceded by a prefix NAL unit, which has non_idr_flag equal to 0. IDR pictures of non-base views, also referred to as view-IDR (V-IDR) pictures in the MVC standard, all have non_idr_flag equal to 0 [P3]. V-IDR pictures may rely on pictures from other views but only within the same access units for decoding by inter-view prediction [P3].

An access unit contains all the NAL units pertaining to a certain time instance. According to the MVC standard, an IDR access unit is an access unit wherein the pictures of all the views are IDR pictures. Such an IDR access unit provides random access support at the time instance to all the views. Note that the MVC standard allows for such access unit wherein pictures of some views are IDR pictures while pictures of other views are non-IDR pictures.

IDR pictures disallow any picture succeeding the IDR picture in decoding order (i.e. bitstream order) to be inter predicted from earlier pictures in the same view. This leads to reduced compression efficiency compared to the typical open GOP coding structures such as the IBBP structure, where the B pictures after the I picture in decoding order precedes the I picture in display order, and can use pictures before the I picture in decoding order for inter prediction. The pictures in such open GOP coding structures are defined as anchor pictures in the MVC standard. Also note that in MVC, an anchor picture is designed to use inter-view prediction so it is not necessarily an I picture [P3]. Anchor pictures can be used as random access points, while application implementers must bear in mind that a few pictures after such random access points may not be correctly decoded when the random access is carried out at these points.

It is also possible to perform random access at non-intra pictures, e.g. using GDR based on the isolated regions technology [23]. In this case, the GDR random access points can be indicated by the recovery point SEI message as specified in H.264/AVC, but included in the scalable nesting SEI message that tells to which views the semantics apply.

3.5.2. View Switching

View switching refers to changing of the target view(s). The number of target view(s) may be one or more. In case the number of target view(s) changes or any of the target view is changed from one view to another, a view switching occurs. View switching must happen at view switching points, after which the new target view(s) can be correctly decoded. A typical application for view switching is free-viewpoint video, which has been shown in Scenario (a) of Fig. 14 [P3].

All random access points can also be used as view switching points. There is another type of switching points that are not random access points. For example, if at picture X the target views can be switched to view subset C from view subset A but not from view subset B, then picture X is a view switching point from view subset A to view subset C. This type of switching points can be realized by specifically setting the inter-view prediction relationship, or by using the SP/SI coding technology [10].

3.6. DECODING ORDER ARRANGEMENT

Decoder order arrangement is related to the requirement of decoder resource consumption as well as interactivities with systems, such as file container or transport protocol.

In H.264/AVC, the order how NAL units are placed inside the bitstream is referred to as the decoding order. In multiview video, where two dimensions, time and view, are involved, and the decoding order description gets more complicated.

Two fundamentally different decoding order arrangements, view-first coding and time-first coding, have been considered by the JVT. In view-first coding [68], within each group of pictures (GOP), pictures of each view are contiguous in decoding order, as shown in Fig. 17, where the horizontal direction denotes time (each time instance is represented by Tm), and the vertical direction denotes view (each view is represented by Sn). Pictures of each view are grouped into GOPs, e.g., pictures T1 to T8 for any view in Fig. 15 form a GOP.

Note that either time-first coding or view-first coding shares the same prediction structure that does not allow inter-view prediction from pictures in other time instances. This design that constrains the prediction inside either the same time instance or the same view in MVC is for simplicity, e.g., fast random access and efficient memory management. In [69], the mixed inter-view/temporal prediction modes are tested. Those modes enable the prediction from pictures within a different view and within a different time instance. It was concluded that the coding gain of having those modes are minor, as usually less than 10% of the blocks will select those modes and in average the percentage is less than 6% [69]. It also brings considerable encoding complexity reduction, if those modes are not supported in MVC.

View-first coding causes a fundamental problem for storage of multiview video bitstreams in media container files based on ISO base media file format [70]. Coded pictures belonging to different views but with the same time instance are interleaved with pictures of other time instances in a bitstream, and thus cannot be in the same access unit. These different access units, when composed into a file according to the ISO base media file format, correspond to different samples. The ISO base media file format requires samples to be ordered in their decoding order. According to the ISO base media file format, the decoding time of a sample is an increasing function of sample number, and the composition time (also used as presentation time) of a sample is indicated as a non-negative increment compared to its decoding time. Consequently, view-first coding would require a composition time offset proportional to the GOP size multiplied by the number of views, which would be perceived as significant initial buffering delay. Furthermore, possibility for parallel decoding would be hard to realize when view-first coded streams are included in files compliant with ISO base media file format, because the indicated decoding and composition times assumed single-processor operation.

To overcome the mentioned problems, time-first coding was introduced in MVC [P3]. In time-first coding, pictures of any temporal location are contiguous in decoding order, as shown in Fig. 18. In this case, pictures of the same time instance can be defined, but belonging to different views as one access unit [P3]. Note that the decoding order of access

units may not be identical to the presentation order. The order of the views in each access unit is the same and there is a map from the view identifiers to the view order index.

With time-first coding, an access unit contains NAL units which are contiguous in decoding order. This definition is similar to the access unit definition in SVC. Therefore, many mechanisms designed in the SVC file format, such as extractors and aggregators, are useful for MVC too. Design principle and technical details for MVC file format can be found in [71].

The following sub-sections on buffer requirement analysis and buffer management apply to time-first coding only.

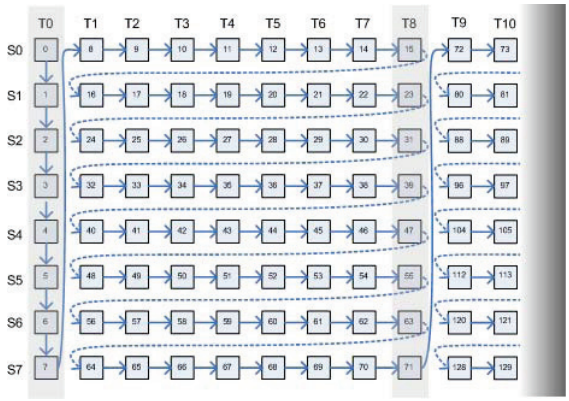


Fig. 17: View-first coding.

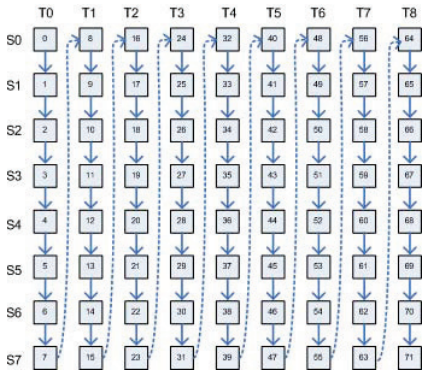


Fig. 18: Time-first coding.

In MVC, pictures in the same time instance are assumed to be outputted simultaneously. Decoded pictures used for prediction or future output are buffered in the Decoded Picture Buffer (DPB). To efficiently utilize the buffer memory, the DPB management processes have been specified, which include a storage process of decoded pictures into the DPB, a marking process of reference pictures, and an output and removal process of decoded pictures from the DPB.

When output is considered, in view-first coding, all the previously decoded pictures in a GOP cannot be removed from the decoded until the picture with the same time instance in the last view in the decoding order is decoded. So, time-first coding requires a much smaller DPB buffer, especially when relatively less number of views and a large GOP size are utilized [S1].

3.7. DECODED PICTURE BUFFER MANAGEMENT

Within the time-first decoding order, DPB buffer management mechanisms are introduced to MVC to further fulfill the decoder resource consumption requirement [P3]. In MVC, decoded pictures from different views are globally managed and the following mechanisms are utilized to enable the management of pictures with a view when one picture is decoded or the management of pictures from views other than the view which the current picture belongs to.

3.7.1. Buffer Management inside a View

Because of the time-first coding structure, whether a picture is a reference picture or non-reference picture can be decided only by its temporal prediction structure. For any two pictures in a view, if picture A follows picture B, then, in the whole bitstream, picture A also follows any picture with the same time instance as picture B. This is not the case in view-first coding, so it may require cross-view explicitly or implicit marking to make those pictures with the same time instance as B but with early decoding time as A as "unused for reference". Therefore, all the memory management control operation commands, if present, are effective inside a view. And the sliding window also takes effect inside a view, which was proposed into JVT in the same time in [P3][72][73].

3.7.2. Buffer Management for Inter-view Reference Pictures

In MVC, there are pictures that are used for inter-view prediction reference but not inter prediction reference. Therefore, whether to mark those pictures as "used for reference" (corresponding to nal_ref_idc larger than 0 in the NAL unit header) after decoding is an issue. Considering the following use case, it was designed in MVC that those pictures can be marked as "unused for reference" [P3].

If such a picture is marked as "used for reference" and is stored as a reference picture, when only base view sub-bitstream is decoded, it is definitely an extra memory burden for the H.264/AVC decoder and the encoder may need to design extra Memory Management Control Operation (MMCO) commands.

This solution solves the problem mentioned above but another question arises: how would those pictures used only for inter-view prediction be managed to reach a better buffer management in terms of DPB size. One argument is if an inter-view picture is not used for inter prediction and is a non-reference picture, it may not be available in the DPB.
Because of the time-first coding structure and the assumption that pictures are outputted at the same time, the concern mentioned above is solved. So there is no extra marking process for those pictures if all views are required for output.

3.8. REFERENCE PICTURE LIST CONSTRUCTION

The reference picture list construction process can flexibly arrange temporal and view prediction references. This provides not only potential coding efficiency gain but also error resilience, since reference picture selection and redundant picture mechanisms can then be extended to the view dimension [P3]. This strengthens the error robustness of the MVC bitstreams.

As H.264/AVC reference picture list construction process, the process in MVC also contains sub-processes, reference picture list initialization and reference picture list modification. The initialization process of MVC is based on the initialization process of

H.264/AVC, with the inter-view reference pictures appended after the inter prediction reference pictures.

The reference picture list modification process and arrange any applicable inter prediction or inter-view prediction reference picture into any position of the final reference picture list. Inter prediction pictures in the same view of the current picture are, as in H.264/AVC, indentified by frame_num; Inter-view prediction pictures, are identified by the inter-view reference index specified in the SPS.

3.9. SEI MESSAGES IN MVC

To make a complete introduction of the MVC standard, besides the previous mentioned view scalability information SEI and scalable nesting SEI messages, other SEI messages are also listed and discussed.

3.9.1. SEI Messages for Adaptation Purposes

These SEI messages are designed to achieve less transmission and/or decoding computations for different scenarios. These SEI messages were proposed by the author's team and are relevant to this thesis, although of less importance.

Non-required view component SEI message

A picture in one view is defined as a view component in the MVC specification. This SEI message indicates the view components that are not required to be transmitted, decoded or buffered for the reconstruction of the output views, in a communication system. A non-required view component refers to such a view component that does not affect the reconstruction of the views to be outputted. Typically a non-required view component is not used for inter-view prediction but may be listed as dependent on by the views to decode according to the view dependency information included in the sequence parameter set in the MVC extension and may have inter_view_flag equal to 1.

One example of the non-required view components are shown in Fig. 19, wherein view 1 is the only output view. View 0 contains non-required view components in the highest temporal levels (odd time instances). Those view components are with inter_view_flag equal to 1 [74].

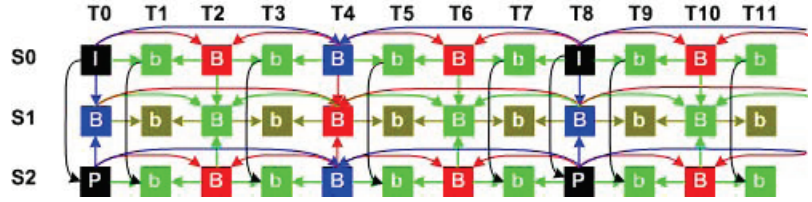


Fig. 19: Example of view components with inter_view_flag equal to 1 in view 0 but not required for the reconstruction of view 1.

View dependency change SEI message

Even though SPS MVC extension specifies the view dependencies, it might happen that the encoder discovers at a certain point that some dependent views are no longer useful for inter-view prediction. Consequently, the view dependency presented in SPS MVC is less accurate and it would be desirable that the information of the new view dependency is signaled. Decoders may use the information to omit the decoding of view components that are no longer used for inter-view prediction and are not for output, and the server or a media aware network element can omit the transmission of these view components. The view dependency change SEI message therefore was introduced to indicate the information of view dependency changes [75].

Operation point not present SEI message

When there are changes in network conditions, e.g. reduction of bandwidth or increase of packet loss, a server or a media-aware network element may have to thin the transmitted or forwarded bitstream in the middle of a coded video sequence. Consequently, some operation points might no longer be present in the transmitted or forwarded bitstream. Another case of operation points becoming not present occurs during a switching of operation point. Signaling of such information about operation points becoming not present is helpful for the decoder to be prepared. Without being aware of this information, the decoder may take some inappropriate actions as it may consider the absence of the NAL units belonging to those operation points as unintentional losses. The operation point not present SEI message was proposed to indicate this information [75].

3.9.2. SEI Messages for other Purposes

Parallel Decoding Information SEI message

To decode multiple views simultaneously is not always possible since there are view dependencies among views. Normally, one view component (picture) cannot be decoded until its dependent view components (pictures) in other views are decoded. This makes parallel decoding of all the views impossible.

This parallel decoding information SEI message introduces the constraints for views such that a dependent view can be decoded right after several rows of the dependent views are decoded and parallel decoding of views can be achieved [76].

Multiview scene information and acquisition information SEI messages

These two SEI messages signal the maximum disparity among the views as well as the camera parameters. The signaled information can be useful for processing the decoded views for better rendering of a 3D display [77].

Chapter 4

Graceful Degradation for Scalable Video and Multiview Video

To avoid drastic deterioration of the video quality when packet losses occur, error resilient coding at the encoder and error concealment at the decoder are necessary. Some techniques for H.264/AVC can be reused for SVC and MVC. However, inter-layer or inter-view dependencies for SVC and MVC receptively, can provide the opportunity for better error resilience and error concealment to realize graceful degradation.

In this chapter, SVC error resilient tools are reviewed in Section 4.1. The contribution of this thesis on SVC and MVC error concealment algorithms are described respectively in Section 4.2 and 4.3.

4.1. ERROR RESILIENCE IN SCALABLE VIDEO

Error resilient and error concealment techniques in general are reviewed in Section 1.2. There are new challenges for these techniques in SVC, as the prediction in the bitstream is more complicated and error propagation could be more severe.

However, if the bitstream is encoded with appropriate error resilient method and the decoder at the client side is equipped with suitable error concealment functions, the system may still maintain graceful degradation for display.

All the standard error resilient video coding tools supported by H.264/AVC are inherited to SVC. However, data partitioning and SP/SI pictures are not included in the currently specified SVC profiles. All the non-standard error control tools are supported by SVC, in the way as in H.264/AVC. Some of these tools that are inherited from H.264/AVC are supported in the SVC reference software, namely the Joint Scalable Video Model (JSVM). These tools are briefly summarized in sub-section 4.2.1.

Besides the tools inherited from H.264/AVC, SVC includes three new standard error resilient coding tools, namely quality layer integrity check signaling, redundant picture property signaling, and temporal level zero index signaling. These tools are especially designed for the SVC features and are discussed in sub-section 4.2.2.

The conventional error resilient coding and error concealment tools for single-layer coding can certainly be applied to the SVC enhancement layers. However, these methods do not utilize the correlations between different layers, which are high in many cases. Improved performance can be expected if inter-layer correlations are utilized. LA-RDO based intra MB refresh is summarized as a non-normative error resilient tool in SVC in sub-section 4.1.3 and while the next sub-section describes in more details the error concealment algorithms that utilize inter-layer correlations in SVC bitstreams.

4.1.1. JSVM Error Control Tools Inherited from H.264/AVC

The reference software of SVC, JSVM, includes the support of FMO, redundant pictures [78][79], slice coding, LA-RDO based intra MB refresh[80], as well as some error concealment methods [S2][P4].

The simplest exact-copy redundant coding for each picture was proposed to the JSVM in [78]. An Unequal Error Protection (UEP) like method which only codes redundant representations for key pictures of enhancement layers was proposed in [79]. The LA-RDO based intra MB refresh algorithm, proposed in [80], was extended from the single-layer method reported in [36]. Four error concealment methods were proposed in [S2] according to the inter-layer prediction characteristics of SVC. Another improved error concealment method using motion copy for key picture was proposed in [P4]. It has also been agreed to be included to the JSVM software, but at the time of this writing, the feature has not yet been integrated. By applying some of these error concealment methods in a combined manner, significant luma PSNR gain compared with single layer error concealment algorithms can be observed.

4.1.2. New Standard Error Resilient Coding Tools in SVC

Quality Layer Integrity Check Signaling

The quality layer integrity check SEI message includes a Cyclic Redundancy Check (CRC) code calculated from all the quality enhancement NAL units (with the syntax element quality_id larger than 0) of a dependency representation (all NAL units in one access unit and with the same value for the syntax element dependency_id). This information can be used to verify whether all quality NAL units of a dependency representation are received by the decoder. When a loss is detected, the decoder can inform the encoder, which in turn may decide to use the error-free base quality layer as reference for encoding subsequent access units. Therefore, the drift error due to the erroneous highest quality layer as a reference can

be avoided. When no loss is detected, the encoder is free to use the highest quality layer as a reference for improved coding efficiency.

Redundant Picture Property Signaling

The redundant picture property SEI message can be used to indicate the correlations between a redundant layer representation and the corresponding primary layer representation. A layer representation consists of all NAL units in one dependency representation and with the same value for the syntax element quality_id. The indicated information includes, when a primary picture is lost, whether redundant representation can completely replace the primary representation: for inter prediction or inter-layer prediction, for inter-layer mode prediction (part of inter-layer motion prediction), for inter-layer motion prediction, for inter-layer residual prediction, and for inter-layer texture prediction. Further details can be found in [79].

Temporal Level Zero Index Signaling

The temporal level zero dependency representation index SEI message provides a mechanism to detect whether a dependency representation at the lowest temporal level (i.e. with temporal_id equal to 0) needed for decoding the current access unit is available when NAL unit losses are expected during transport. Decoders can use the SEI message to determine whether to transmit a feedback message or a retransmission request concerning a lost dependency representation at the lowest temporal level. More details can be found in [81].

4.1.3. LA-RDO Based Intra MB Refresh for SVC

In SVC, when encoding an MB in an enhancement layer picture, the traditional MB coding modes in single-layer coding as well as new inter-layer prediction mode can be used. Similar to single-layer coding, MB mode selection in SVC also affects the error resilient performance of the encoded bitstream. As an encoder algorithm in JSVM, a method which is extended from the single-layer method in [36] to multi-layer coding is presented. In this method, given the target PLR, the 4x4 block-based error propagation maps for a picture is calculated, and the map is taken into account to perform mode decision for pictures in the latter. This method is presented as part of publication [P4].

4.2. FRAME LOSS ERROR CONCEALMENT FOR SVC

The contributions of this thesis in graceful degradation in SVC include the error concealment algorithms which have been adopted into the JSVM software and published in [S2][P4]. Different types of error concealment algorithms are implemented by the author and his co-workers in the current JSVM software. They are summarized as intra-layer error concealment and inter-layer error concealment. One of those methods, if used, is applied to the whole picture, although it is possible that different MBs can selectively use different methods.

From sub-section 4.3.1 to sub-section 4.3.3, the error concealment algorithms in [P4] are introduced. The methods are compared in sub-section 4.3.4.

4.2.1. Reference Picture Management for Lost Pictures

Upon detection of a lost picture, a key picture is concealed as a lost P picture, and the necessary RPLM commands and MMCO commands are set to enable DPB management. The RPLR commands are to guarantee the current picture to be predicted from the previous key picture. The MMCO commands are to mark the unnecessary decoded pictures in the previous GOP so as to guarantee the minimum DPB even when packet losses occur. How to conceal a lost key picture is to be discussed in the following sub-sections.

If a lost picture is not a key picture, usually the RPLM commands can be constructed based on those pictures in the previous GOPs or those in the base layer picture if the lost picture is in the enhancement layer.

On the basis of the current design of SVC, the corresponding enhancement layer picture will not be decodable if the base layer picture is lost unless two layers are independently encoded. So base layer picture loss leads to the “loss” of the whole access unit, and the loss of one picture of a certain layer leads to the “loss” of the pictures in all the higher layers of the same access unit.

4.2.2. Intra-layer Error Concealment Algorithms

Intra-layer error concealment is defined as the method which uses the information of the same spatial or quality layer to conceal a lost picture. The author and his co-workers have introduced three methods in [S2][P4], described as follows.

Picture Copy (PC)

In this algorithm, each pixel value of the concealed picture is copied from the corresponding pixel of the first picture in the reference picture list 0. If multiple-loop decoding is supported for an error concealment method, this algorithm can be invoked for both the base layer and enhancement layers. Otherwise, only the highest layer in the current access unit can be used for concealment.

Temporal Direct (TD) for B pictures

The temporal direct mode specified in H.264/AVC is generated as follows. As can be seen in Fig. 20, it is assumed that a MB or MB partition in the current B picture is coded in temporal direct mode, and then its motion vectors are inferred from its neighboring reference pictures. If the co-located MB or MB partition (belongs to List 1 Reference as shown in Fig. 20) in the reference picture list (namely list for simplicity) 1 uses a picture (named in Fig. 20 as List 0 Reference) as a reference in list 0 and that picture is also in list 0 of the current B picture, then List 0 Reference and List 1 Reference are chosen to bi-predict the MB or MB partition of the current picture which is being processed. List 0 and list 1 motion vectors

MV0 and MV1 are scaled from MVc using the POC (i.e. display order) differences. The detailed derivation can be found in [6].

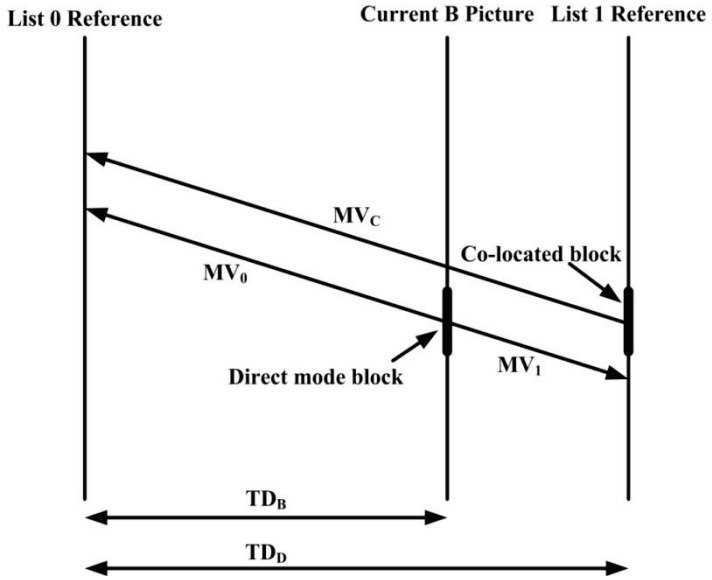


Fig. 20: Example for temporal direct-mode motion vector inference.

The temporal direct mode specified in H.264/AVC standards cannot be used for any spatial or SNR enhancement layer. However, the concealment of the B picture in SVC can still be applicable for both base layer and enhancement layer. Using the calculated MVs including list 0 and list 1 motion vectors, motion compensation from two specific reference pictures is utilized to predict the MB in the lost picture, assuming zero residues.

In the current SVC design, the necessary motion vectors are stored for each layer. This makes it possible to apply TD at the decoder without introducing extra memory requirement.

Motion Copy (MC) for key pictures

The MC algorithm is applicable for the lost key pictures. Key pictures are concealed as P pictures no matter whether they are originally I or P pictures. Since TD is not applicable for key pictures and PC may not be efficient because the gap of two key pictures may be large (depending on the GOP size). To get a more accurately concealed picture for the lost key picture, motion vectors are re-generated by copying the motion field of the previous key picture.

4.2.3. Inter-layer Error Concealment Algorithms

Two methods as inter-layer error concealment algorithms [P4][S2] are introduced: one works for single-loop decoding and the other for multiple-loop decoding.

Intra-layer error concealment algorithms make uses of only the temporal correlation. However, the inter-layer prediction, e.g., in the texture of two layers or in the motion vectors of two layers, if appropriately used, can significantly increase the quality of the concealed

pictures, especially for the case when an enhancement layer picture is lost while the base layer picture is successfully received and correctly decoded.

(a) BLSkip **(b) RU**

(c) TD **(d) PC**

Fig. 21: Subjective quality comparison of concealed pictures for *foreman* packet loss rates of base layer and enhancement layer are both 3% and QP=28.

Base Layer Skip (BLSkip)

This method operates as follows. If the base layer is an intra MB, then texture prediction is used. If the base layer is an inter MB, then motion prediction as well as residual prediction are used to generate information for an MB in a lost picture at the enhancement layer. In this case, motion compensation is done at the enhancement layer using the possibly upsampled motion vectors. This algorithm can directly be used for the enhancement layer if there is no picture loss in the base layer. If a base layer picture is also lost, the motion vectors for base layer picture are generated using the TD method first. This method is called as BLSkip+TD, but for simplicity, BLSkip is used to represent this method.

Reconstruction base layer and possibly Upsampling (RU)

In the RU algorithm, the base layer picture is reconstructed, and may be upsampled for the lost picture at the enhancement layer, which is depended on the spatial ratio between the

enhancement layer and the base layer. This requires full decoding of a base layer and thus leads to the requirement of multiple-loop decoding. This method is helpful when there are continuous picture losses only in the enhancement layer and may be competitive for low motion sequences compared with BLSkip.

4.2.4. Performance of the Proposed Error Concealment Algorithms

Inter-layer error concealment algorithms utilize the inter-layer correlations, which are highly helpful for the reconstruction of the lost packets in the SVC enhancement layer. As shown in paper [S2] and [P4], the quality of the concealed sequences is improved significantly, compared to the other low-complexity error concealment algorithms, i.e., PC and TD. For example, the subjective quality comparison is shown in Fig. 21. The error concealment algorithms improved the objective quality of the concealed sequences with an average PSNR gain of about 3 dB for both high-delay hierarchical B picture coding structure or low-delay IPP… coding structure, compared with the picture copy method.

4.3. FRAME LOSS ERROR CONCEALMENT FOR MVC

Coding tools can utilize correlations among different views to compress multiview content better than coding each view independently. One of the goals for the development of the MVC standard [54] is to develop such tools. Redundancies between views also provide opportunities for improved error concealment. During the video data transmission, packet losses may occur. This may lead to undesirable effects such as system instability, unacceptable video quality and unpredictable decoder behavior.

4.3.1. Motivation of Error Concealment for MVC

The contribution of this thesis for MVC error control mainly focuses on decoder error concealment, as described in [P5]. Some algorithms based on decoder error concealment have been proposed in [37][82][83]. They make use of temporal or spatial correlation between the MBs in damaged area and its adjacent MBs in the same or previous frame. These algorithms assume that if either a single MB or a slice consisting of several consecutive MBs is lost, information from the neighboring available MBs or MBs in the adjacent frames can be used to estimate both motion vectors and texture of the missing MB. However, in some applications, a coded picture typically fits in one packet, and a transmission error will lead to a loss of a whole slice or frame. Algorithms handling frame loss have also been proposed.

Getting motion information for a lost frame is important for the performance of an error concealment algorithm. In an MVC bitstream, there is usually a high correlation between views. This correlation, such as motion field similarity, can be used to get a more accurate estimation of coding modes and motion vectors. An algorithm utilizing this correlation has

been proposed. Experimental results show that the proposed algorithm can improve video quality comparing to low complexity temporal error concealment algorithms.

4.3.2. Algorithm Description

Moreover, redundancies between views also provide opportunities for improved error concealment. During transmission of video data, packet losses may occur. This may lead to undesirable effects such as system instability, unacceptable video quality and unpredictable decoder behavior [P5].

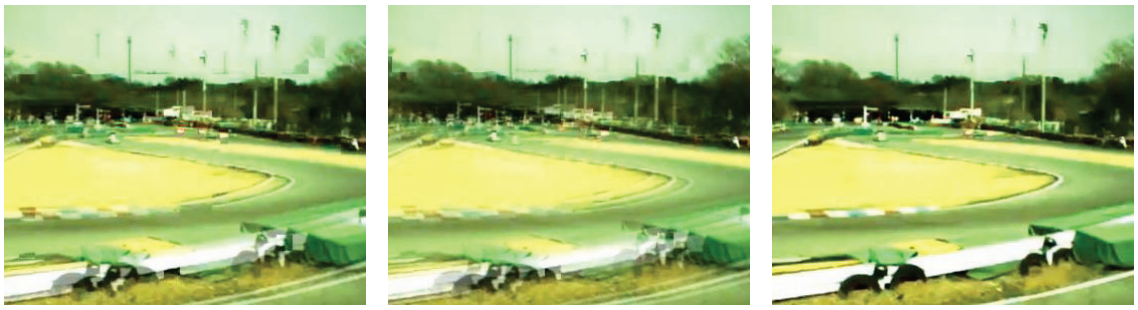
In the encoder side, the global disparity motion regarding the inter-view reference picture relative to the base view is sent for every anchor picture. In the decoder, the Motion Prediction algorithm is adopted to generate the motion field of the lost frame. The MP algorithm is described as follows. When a picture is detected as lost, each MB of the lost frame is processed as follows. First, the corresponding MB (CMB) in a dependent view is found according to the global disparity motion. The mode and motion vectors of CMB are then copied for the lost MB. Finally, motion compensation is used to generate the concealed picture.

There are cases when the forward method is not applicable for some special MBs or even a whole slice. They are concealed by other methods. If the CMB does not contain motion information, e.g., it is intra coded, spatial error concealment for this MB is utilized. If the slice is a P slice, the current MB is set to skip mode in P picture; if the slice is a B slice, spatial direct mode is utilized. If an anchor picture in one view that is not the base view is lost (it is detected to be an anchor picture if its corresponding picture in the base view with the same time instance is an anchor picture), it is concealed by copying sample values from that corresponding anchor picture of its dependent view.

4.3.3. Performance of the Algorithm

PC and TD error concealment algorithms are used for comparison in publication [P5]. The experimental results show that the proposed algorithm obtained significant gains compared to the PC and TD error concealment algorithms, exceeding 2.6 dB and on average 0.97 dB compared with PC algorithm in PSNR, as better motion vectors can be estimated with the proposed method which uses inter-view correlations, compared with PC or TD.

As an example, decoded pictures with degradation because of error propagation are shown in Fig. 22. They are the 162nd frame of *Race1* and correspond to the three different error concealment algorithms. As shown in Fig. 22, the proposed algorithm, motion prediction (MP), introduces less degradation.



(a) PC: Cropped picture **(b) TD: Cropped picture** **(c) MP: Cropped picture**

Fig. 22: Subjective quality comparison of pictures for *race1*, with no loss in view 0 and 10%, loss rate in view 1, QP = 37.

The complexity of the proposed algorithm is also comparable to that of normal decoding computations, as the modes and motion vectors of another view are directly used.

Chapter 5

Coding Algorithms for Multiview Video

3D video refers to any video representation that provides immersive rendering of a scene. In the context of this thesis, multiview video covers video representation with two or more views, namely stereoscopic video or multiview video, whereas 3D video refers to the video presentation that has rendering capability for a view in any position or angle. The 3D video content is usually presented by texture video sequences as well as depth maps.

In the thesis, techniques for the coding of multiview video content are proposed. The goal of the proposed methods is to code those video sequences with high efficiency.

This chapter is organized as follows. In Section 5.1, the coding tools in MVC and Joint Multiview Video Model (JMVM) are reviewed. In Section 5.2, single-loop decoding for multiview video is proposed. A series of coding tools, focusing on the asymmetric coding for stereoscopic video but can also be extended to multiview video, are proposed in Section 5.3. Although the focus of this chapter is on multiview video, the multiview content can be extended to 3D video with depth maps included for 3D rendering. Therefore, in Section 5.4, a method of jointly coding texture and depth is proposed, followed by an introduction of 3D rendering technology.

5.1. REVIEW OF MVC CODING TOOLS

Besides the inter-view prediction enabled in the MVC specification, other potential features, especially those tools focusing on improved coding efficiency, have been investigated in JVT and are described in the JMVM [84]. In this sub-section, discussions on the compression efficiency and decoder complexity of the coding tools are provided. As a contribution of this thesis, the coding techniques in MVC and JMVM have been compared

and analyzed in [P6], and some of the major techniques are summarized as follows while more simulation results can be found in [P6].

5.1.1. Description of the Coding Tools

The JMVM specifies two extra coding tools focusing on compression efficiency: Illumination Compensation (IC) and motion skip. Note that JMVM is not part of the MVC standard and thus these coding tools are not supported in the MVC specification. Both of these tools are added as new coding modes for inter picture coding. Although both of them are designed to improve the inter-view prediction, IC, however, can be used for inter prediction in the non-base views. Motion Skip has more constraints in terms of applicability: it does not apply to anchor pictures wherein the inter-view references do not have inter prediction motion vectors. Since both of them are just additional modes, an RDO encoder can select the best mode among these two and other existing modes, if applicable.

Illumination and color inconsistencies can happen due to the different lighting conditions for the different cameras. Although this problem can be somewhat solved by proper condition settings when capturing the video or by pre-processing. They are not always guaranteed when video sequences are provided to an MVC encoder.

IC is a technique to solve this problem in the codec level, by subtracting the difference of the means of the reference block and the original block during motion compensation, as has been described in [85]. It can be decided independently for each MB whether IC is to be used, based on Rate Distortion Optimization (RDO). This difference, namely local illumination change, is signaled for those blocks that use IC mode. At the decoder, an IC block is reconstructed by adding the motion compensation predictor, the residual and the illumination change, as shown in equation (6).

$$I(i,j) = R(i,j) + r(i+x, j+y) + C \qquad (6)$$

wherein $I(i,j)$ represents the reconstructed block, $R(i,j)$ is the residual signal, $r(i+x, j+y)$ is the reference block and C is the illumination change value for this block. At the encoder, each motion search needs an extra calculation of the means of the reference block.

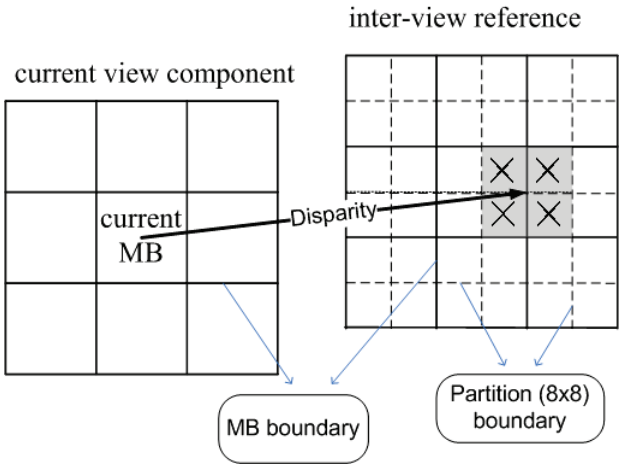


Fig. 23: Motion Vector Derivation in Motion Skip.

In SVC, there is motion correlation between layers and the correlation is used for inter-layer motion prediction. Similarly, in MVC, motion correlation between views can also be utilized to improve the coding performance. Motion Skip (MS) is a motion vector (MV) derivation tool to enable reusing of the MVs from a view component in the same time instance but belonging to other views by a given disparity, which was signaled globally and extended locally to each MB that utilizes motion skip [84]. Motion skip is a technique that can be utilized to realize single-loop decoding for MVC [P7], which is to be discussed in Section 5.2. The technique is similar to SVC inter-layer motion prediction while it does not enable motion refinement. In JMVM, a global disparity is maintained and an offset is given for an MB with MS mode to calculate the local disparity, which is of 8-pixel accuracy. An example of motion reuse for a MB using MS mode is shown in Fig. 23. The current MB has disparity points to four 8x8 blocks in an inter-view reference. The MVs of those blocks are reused for inter prediction motion compensation within the current view. The complexity increase at the decoder is minor, although the increase at the encoder is substantial due to local disparity search.

5.1.2. Experimental Results of Coding Efficiency

Simulations have been performed based on the JVT common test conditions [86] and the reference software [87] for MVC. The coding efficiency comparisons of different coding tools in MVC and JMVM are reported in [P6], and further experiments are made as part of this thesis. Briefly speaking, MS and IC provide respectively 4% and 5% bitrate reduction over MVC, and in total, JMVM coding tools can achieve around 8.5% bitrate saving, which is equivalent to 0.36 dB PSNR gain. MVC, however outperforms simulcast AVC with a bitrate reduction of 20%. A typical Rate Distortion (RD) curve is shown in Fig. 24. It can be concluded that the coding efficiency improvements from simulcast to JMVM are mostly from the inter-view prediction tools defined in MVC. As also shown in Fig. 24, each coding tool, MS or IC, is potentially useful.

5.1.3. Decoder Complexity and Implementation

At the decoder, IC requires pixel level processing: adding a value to each block; while, MS requires only parsing and deriving the MVs. Altogether, JMVM tools do not require significant complexity increase, although for hardware solutions, completely new modules need to be realized. MVC only requires slice header or higher syntax changes based on H.264/AVC, thus needs no new hardware implementations for major processing modules.

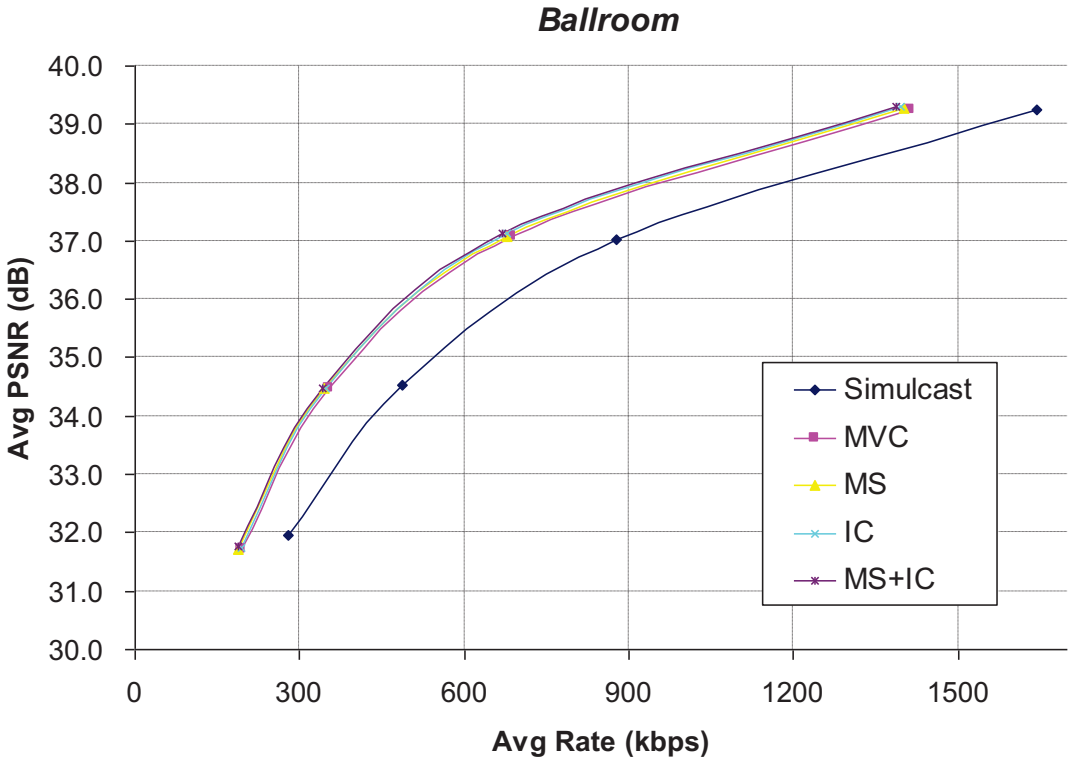


Fig. 24: RD curves for JMVM, MVC and simulcast H.264/AVC.

5.2. SINGLE-LOOP DECODING FOR MVC

Among typical MVC applications, such as free-viewpoint video, 3D TV, and immersive teleconferencing, there are cases that display only a subset of the encoded views. In this thesis, the views displayed are referred to as the target views or output views, while the remaining views are referred to as the dependent views. Target views are related to dependent views through inter-view prediction. For example, in free-viewpoint video, only one view needs to be displayed at a certain time instance and that view is the target view. As part of the contributions of this thesis, single-loop decoding was proposed to solve this problem by avoiding the decoding of those views that are not used for output, as described in [P7].

In the latest specification of MVC [54], inter-view prediction is realized by utilizing pictures from other views as reference pictures for motion compensation.

Single-loop decoding (SLD) [49] is supported in SVC [39]. The basic idea of SLD in SVC is as follows. To decode a target layer that depends on a number of lower layers, only the target layer itself needs to be fully decoded, while for the lower layers only parsing and decoding of Intra macroblocks (MBs) are needed. Essentially, SLD in SVC requires motion compensation only at the target layer.

Consequently, SLD provides a substantial complexity reduction. Furthermore, since the lower layers do not need motion compensation and no sample values need to be stored in the

decoded picture buffer (DPB), the decoder memory requirement is greatly reduced compared to conventional Multiple-Loop Decoding (MLD), where motion compensation and full decoding is needed in every layer, as in the scalable profiles of earlier video coding standards.

In this thesis, SLD is applied to MVC similarly as it is applied to SVC. In the proposed SLD for MVC, only the target views are fully decoded, while the non-anchor pictures of the dependent views need only to be parsed to obtain information required for inter-view prediction. The SLD scheme is proposed in [P7] to be included in MVC. As has been reported in [P7], a similar idea was also proposed to JVT, with differences on high level syntax design. The proposed SLD scheme was adopted into the JMVM.

5.2.1. Proposed Single-loop Decoding in MVC

To achieve SLD, for coding of non-anchor pictures, only motion skip is applied for inter-view prediction. Inter-view sample prediction is only used for coding the anchor pictures. A flow chart for SLD at the decoder is shown in Fig. 25. When decoding an anchor picture, inter-view sample prediction can be used and the picture is fully decoded and stored into the DPB. When decoding a non-anchor picture, inter prediction and motion skip can be used, if this non-anchor picture belongs to a target view, it is motion compensated, reconstructed and stored in the DPB; otherwise, it is only parsed and only its coding modes and motion field are constructed.

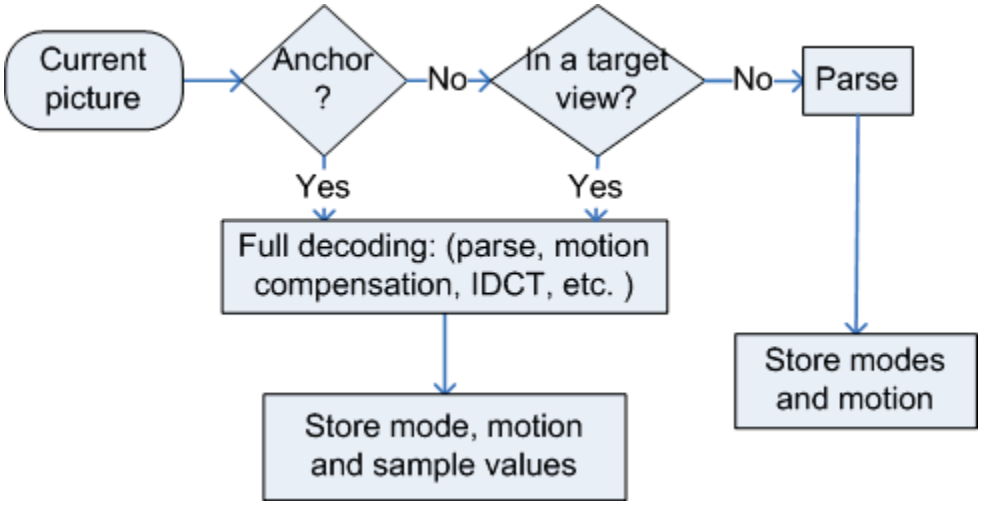


Fig. 25: Single-loop Decoding scheme in MVC.

An example of decoding complexity reduction and memory saving in the free-viewpoint video case is given for the prediction structure shown in Fig. 15. It is assumed that view 5 is the only target view. When the MLD scheme is in use, all the pictures in views 0, 2, 4, 5 are fully decoded and stored in the DPB in order to successfully decode view 5. However, if the SLD scheme is in use and a non-anchor target picture is decoded, the dependent pictures in the same access unit need only to be parsed for generating the coding modes and motion

vectors, while their sample values do not need to be constructed. Consequently, approximately 60% of the decoding complexity and 50% of the memory for decoded pictures can be saved, compared with MLD, wherein all the pictures in view 0, 2, 4, 5 are fully decoded.

5.2.2. Experimental Results of Coding Efficiency

Experimental results in [P7] showed that the proposed SLD scheme achieves a significant compression gain compared to the constrained MVC coding, which enables inter-view prediction only at anchor pictures to realize single-loop decoding. Furthermore, the simulation results indicated that the proposed SLD scheme entails a minor compression efficiency loss compared to MLD scheme (that is supported by MVC) but reduces decoder complexity and memory usage remarkably, as illustrated in sub-section 5.2.1. Compared with simulcast, the proposed SLD scheme has an average bitrate saving of about 25%. Compared with another scenario that supports SLD by disabling inter-view prediction for non-anchor pictures, 7.5% bitrate saving in average can be achieved. The proposed SLD scheme has been adopted into the JMVM.

5.3. ALGORITHMS FOR ASYMMETRIC CODING

The term asymmetric Multiview Video Coding (asymmetric MVC), introduced in [P8], refers to the coding scenarios wherein a subset of the views have lower fidelity than others. The subset can be with lower SNR (Signal-to-Noise Ratio) values or with lower temporal or spatial resolutions. However, most early research works have been focusing on the so-called mixed-resolution stereoscopic video.

In this section, the concept of asymmetric stereo video coding is firstly introduced and the previous research works in this specific area is reviewed in sub-section 5.3.1; a low-complexity method in asymmetric coding is introduced in sub-section 5.3.2; based on that, coding tools improving the inter-view prediction performance in such an asymmetric coding platform are presented in sub-section 5.3.3; decoding complexity analysis and performance assessment of the relevant algorithms are given in sub-section 5.3.4 and 5.3.5 respectively.

5.3.1. Asymmetric Coding for Stereo Video

Mixed-resolution stereo images was proposed based on the suppression theory [88], wherein it is noted that the overall sharpness and depth cue of the stereo image is determined by the high-resolution member of the mixed-resolution pair. Based on the suppression theory, independent research activities also showed that it is possible to reduce the spatial resolution of an image viewed by one eye without affecting the overall impression of sharpness [89][90][91]. Perkins indicated that mixed-resolution stereo image sequences can provide acceptable image quality with reduced transmission bandwidth [92]. In [93], the authors studied the temporal and spatial mixed-resolution approaches based on subjective

tests and it was concluded that using half resolution for one view provides near-transparent quality and quarter resolution for one view leads to a slight quality drop. Both of these two approaches, namely spatial filtering, are effective ways to reduce transmission bandwidth. However, using a lower frame rate for one view (called temporal filtering) leads to a relatively low subjective quality and is not recommended to achieve the same purpose of bandwidth reduction [92][94].

In [95], a different spatial resolution ratio was tested and it is claimed that with 20% bitrate increase on top of mono-view coding, similar subjective quality can be obtained and it is possible to have non-dyadic ratio and better filters to reduce the bit-rate increase to 0%. The solution in [96], however, still focused on half and quarter resolutions. Both of these techniques are based on H.264/AVC and with disparity compensation from the base view. A conventional asymmetric stereoscopic coding scheme based on MVC is shown in Fig. 26. For the stereoscopic scenario, two views are coded, one is in the original resolution (e.g. VGA) and the other is in a lower resolution (e.g. QVGA). Compared to the MVC standard, there is a downsampling process required to generate an inter-view reference picture with the same size as the low-resolution view from the high-resolution view, so that H.264/AVC motion compensation can be applied for inter-view prediction. Downsampling filters with good low-pass frequency response, e.g., the MPEG-4 downsampling filter [97], with coefficients [2, 0, −4, −3, 5, 19, 26, 19, 5, -3, -4, 0, 2]/64, can be used.

According to [98] and [99], different types of distortion in visual appearances impact stereoscopic image quality differently. Blockiness appears to be much more visually disturbing than blur. If the left-eye and right-eye images are with different degrees of blur, the quality seems to be dominated by the high-quality view [98]. However, if both views are distorted with blockiness, the perceived stereoscopic image quality seems to be an average of the image quality of the left and right-eye views [99].

As for the mixed-resolution (spatial) case, when displaying the low-resolution view, upsampling is required. The artifact introduced by upsampling is closer to blur than to blockiness. This justifies the motivation of utilizing a low-resolution picture instead of a low-SNR picture.

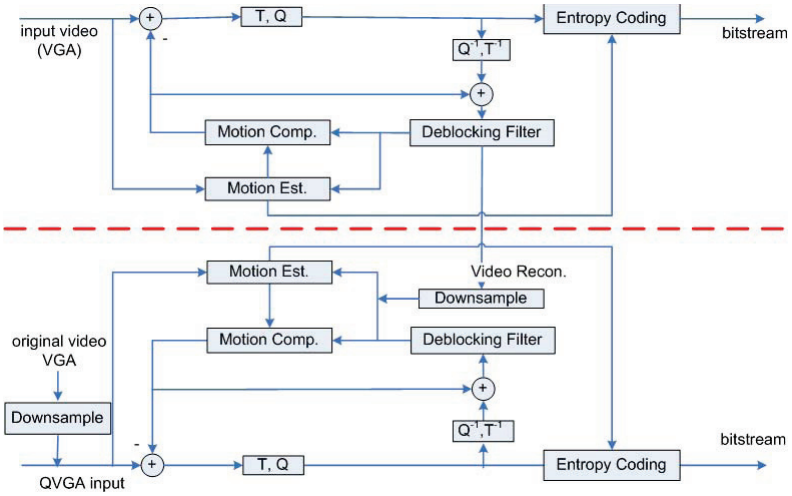


Fig. 26: A block diagram for asymmetric coding.

This stereoscopic scenario can be extended to asymmetric MVC with more than two views, e.g., for bandwidth reduction purpose, as proposed in [100], wherein the MPEG-4 downsampling filter is used. It has been shown in [100] that the conventional asymmetric coding can save bandwidth compared with coding two views with the same resolution using MVC. In [101], simple downsampling, which takes the ¼ of the samples of the high resolution picture, was proposed. This method brings noticeable coding efficiency loss compared to the one proposed in [100].

5.3.2. Direct Motion Compensation for Asymmetric MVC

As published in [P8], this section presents a motion compensation method that provides complexity reduction over the existing techniques as described in [96][100] while maintaining the same level of coding efficiency,. The method preserves all reconstructed samples of the high resolution pictures during inter-view prediction, hence enables advanced inter-view prediction techniques in this asymmetric coding scenario.

A picture in the DPB can play two different roles: an inter prediction reference picture for the following pictures in the same view, and an inter-view reference picture (inter-view picture for simplicity) for the pictures in the same time instance. In asymmetric MVC, an inter-view picture, when referenced by a picture with a low resolution, needs to be downsampled to apply the conventional motion compensation (MC). Therefore, on-the-fly downsampling process or downsampling of the whole inter-view picture is needed.

To avoid the extra computational complexity, the DMC is proposed to be applied in inter-view prediction between views with different resolutions in [96][100]. In DMC, a motion vector is scaled and the scaled motion vector is utilized to find the pixels in the inter-view picture. It has the following sub-processes.

Motion vector scaling

The motion vector is scaled based on the resolutions of the current picture and the inter-view picture. For the dyadic case, the resolution ratio is 2, and $mv'=2mv$.

Luma motion compensation

A scaled motion vector points to even-sample positions (when the original motion vector points to integer-sample positions in the virtually downsampled picture), odd-sample positions (when the original motion vector points to half-sample positions), or half-sample positions (when the original motion vector points to quarter-sample positions) in the inter-view reference picture.

An example of the DMC prediction is shown in Fig. 27. Samples in a 4x4 block can be predicted from an 8x8 block in view 0 picture consisting of integer samples. The integer samples with the same parity then form a 4x4 block (each sample is predicted from the sample marked with the same number in the figure).

When the scaled motion vector points to an odd sample position or an even sample position, no interpolation is required and the sample values in those positions can be directly used. When the scaled motion vector points to a half-sample position, the neighboring integer samples are averaged, similarly to the method used in H.264/AVC for generating the luma values for quarter-sample positions.

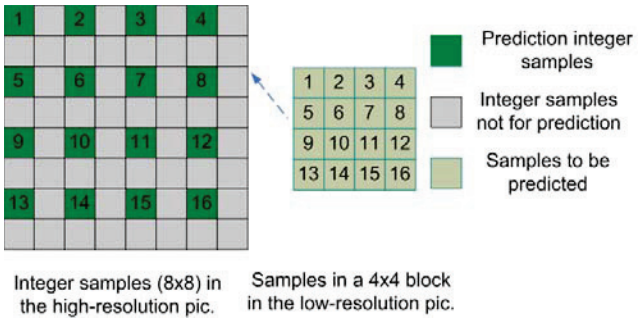


Fig. 27: DMC: the MV points to integer samples in the high resolution inter-view picture.

Chroma motion compensation

Chroma MC is carried out in the same way as in H.264/AVC by applying the scaled motion vector to the chroma components of the inter-view reference picture. The scaled motion vector points to integer-sample positions, half-sample positions, or quarter-sample positions, but never points to 1/8 sample positions. As in H.264/AVC, the bilinear filter, as shown in equation (1) and illustrated in Fig. 6, is used for interpolation of non-integer positions. However, in equation (1), s is 4 and d_x and d_y can be only 1, 2 or 3, while in H.264/AVC, s is 8 and d_x and d_y can be a value from 1 to 7, A, B, C, D are the values of the integer samples and v is the value of the interpolated non-integer sample.

5.3.3. Adaptive Filter Generation for Inter-View Prediction

For multiview content, phenomena, such as the imperfect calibration, different camera parameters and focus changes across views, may lead to less efficient inter-view prediction performance based on H.264/AVC MC or DMC. Moreover, during DMC, only ¼ of the integer pixels in the high-resolution are considered for each compensation of an MB or MB partition and ¾ of the other pixels in the predicted area are not used. Those pixels can be potentially beneficial in coding of the low-resolution view. The adaptive filtering approaches, which take the advantage of the other ¾ pixels in the DMC process, are designed to address the above problems.

Picture-level filtering (global adaptive filtering: GAF) was proposed in [P9] and regionally adaptive filtering was proposed in [P10]. In this chapter, the design approach, and analysis on both compression efficiency and decoding complexity are provided.

This sub-section focuses on the algorithm aspects of the filtering approaches during encoding. The decoding processes and the complexity requirements are discussed in the next sub-section.

The adaptive filters are generated by minimizing the prediction error between the original picture and the picture predicted by the high-resolution picture.

Multiview sequences can have regions with depth level difference, which can exist between background and foreground or among different objects in the foreground. Regions of different depth levels are affected, e.g., blurred at different extents because of focus mismatch.

For each MB or MB partition in the low-resolution picture, its predictor is found by disparity motion compensation. As regions in different disparity levels need different filters, the disparity motion vectors are classified into K clusters. A region, as a collection of MBs or MB partitions associated with the same cluster of disparity motion vector, is formulated together to calculate one adaptive filter. Note that there are also MBs or MB partitions which are not considered in any region for the calculation of adaptive filters. More specifically, the above processes are described as follows.

Regionally Adaptive Filter Generation

This process describes how the adaptive filters are calculated given the original picture and the predicted picture, more precisely, given the collection of MB or MB partitions in the original picture and the predicted blocks. For each pixel in the original picture, its prediction value is calculated as follows. Firstly its corresponding pixel is located in the high-resolution picture used for inter-view prediction. Then the convolution of the nearby pixel values (in a fixed template) and the filter is calculated as the prediction value.

Assume that there are K depth levels in a picture. The following optimization problem is to be solved.

$$\mathrm{H}^{*} = \arg\min_{\mathbf{H}}(e^2) = \arg\min_{\mathrm{H}}\Big(\sum_{i=1\cdots K}\sum_{s_p^i \in S^i}(b_p^i - U_p^i \cdot H^i)^2\Big) \qquad (7)$$

Wherein $\mathbf{H}^* = \{H^1 \quad ... \quad H^K\}$ are the K filter coefficients and $\mathbf{S^i}$ is the set of the pixels belonging to the i-th depth level. b_p^i values are the target sample values of pixels in set $\mathbf{S^i}$ which belong to the original low-resolution picture that is being coded. $U_p^i = \left[u_{p1}^i \quad ... \quad u_{pN}^i\right]$ are the sample values of the group of corresponding pixels used to generate the predictor value for b_p^i . Those pixels U_p^i belong to the inter-view picture. The solution of this problem provides the Least Mean Square (LMS) error between the original samples in the low-resolution picture and the predicted values generated by filtering the high-resolution inter-view picture.

To solve this problem, different sample sets $\mathbf{S^i}$ and the groups of corresponding pixels U_p^i are to be identified [P10].

When only one depth level is considered, inter-view prediction is applied with picture-level global adaptive filtering (GAF); and when multiple depth levels are considered, inter-view prediction is applied with regionally adaptive filtering (RAF). When the sample sets and the corresponding pixels are identified, the above optimization problem is solved by the following sub-problem for each depth level i:

$$H^{i*} = \arg\min_{H^i}(e^2) = \arg\min_{H^i}\left(\sum_{s_p \in S}\left(b_p^i - U_p^i \cdot H^i\right)^2\right) \qquad (8)$$

The solution to the above LMS problem is as follows:

$$H^{i*} = \mathbf{U}^{i+} \cdot b^i \qquad (9)$$

Wherein $\mathbf{U}^{i+} = \left(\mathbf{U}^{iT} \cdot \mathbf{U}^i\right)^{-1} \cdot \mathbf{U}^{iT}$ is the pseudoinverse of the matrix $U^i = \left[U_1^{iT} \quad ... \quad U_M^{iT}\right]$, which is of size NxM and $b^i = \left\{b_p^i\right\} = [b_1^i \quad ... \quad b_M^i]^T$,where T here denotes the transpose operation of a matrix or a vector.

The above problems need the information of the sample set $\mathbf{S^i}$ in the low-resolution picture as well as the corresponding pixels in the high-resolution inter-view picture [P9].

Disparity Motion Estimation and Clustering

For each MB or MB partition of the original picture, its predictors are found by disparity motion vector estimation. After disparity motion vectors (DMVs) are estimated, a clustering method is proposed to segment a picture into different depth regions by the disparity motion vectors.

To get the disparity motion vectors, either an extra block matching module or multi-pass encoding can be applied.

1) 16x16 block matching

As there are disparities between views, the corresponding block in the picture in view 0 is found by block matching. A 16x16 block matching algorithm is utilized. The block matching is applied for each MB of the low-resolution picture and only the integer positions in the high-resolution picture are searched [P9].

2) Multi-pass encoding

In multi-pass encoding, the first pass provides the information that indicates which of the MBs that are related to inter-view prediction and the disparity motion vectors of those MBs.

If the motion vectors point to integer positions in the high-resolution view, they are directly used. If they point to half-sample positions, they are quantized to the nearest half-sample motion vectors, which correspond to the integer positions in the high-resolution view.

Running the entire encoding process including motion estimation is an accurate way to get the disparity motion vectors. However, the shortcoming is extra encoding processes, especially the motion estimation and mode decision invoked in the H.264/AVC as well as MVC encoding.

The clustering method to segment a picture into different depth regions is proposed as follows.

Given the number of desired depth levels K for a picture, a K-means algorithm is utilized to cluster the DMVs into K classes, by minimizing the following squared error function:

$$E = \sum_{i=1}^{K} \sum_{v_j \in V_i} \left\| v_j - \mu_i \right\| \quad (10)$$

where the K clusters of DMVs are $V_i, i = 1 \dots K$ and $\mu_i, i = 1 \dots K$ is the centroid (or the mean) of all the DMVs $v_j \epsilon V_i$. $\|\cdot\|$ is the Euclidean norm.

The K-means problem is solved by Lloyd's algorithm [102], in which the centroids are initialized and then updated based on the following repeatedly executed steps:

1. Classify all DMVs into different clusters based on the current centroids. A DMV is classified to the cluster with the nearest centroid to this DMV.
2. Recalculate the centroids: $\widehat{\mu_i}, i = 1 \dots K$.
3. If the centroids are not changed, that is $\|\mu_i - \widehat{\mu_i}\| < \varepsilon, \forall i = 1 \dots K$, terminate the iteration; else, set $\mu_i = \widehat{\mu_i}, i = 1 \dots K$, and return to Step 1.

In this work, ε is set to 1. After the clustering of the disparity motion vectors, segmentation is done to divide the picture into different depth regions, each of which contains the MBs that have an acceptable distortion and a DMV being classified into the corresponding DMV cluster. So we have

$$\mathbf{S^i} = \left\{ S_t^j \middle| S_t^j \epsilon MB_t, v_t \epsilon V_i \right\}, i = 1 \dots K \quad (11)$$

wherein v_t is the DMV of MB_t, $s_t^j, j = 1,2, \dots 256$ are the pixels of MB_t, and V_i is the i-th cluster of DMVs and MB_t denotes the t-th MB in the picture in view 1 [P10].

Relevant Region Selection for the Adaptive Filter

The optimization problem, described in equations (7) and (8), seeks the least squared error solution for a specific sample set in a picture. The low-resolution view (view 1) is coded in a hybrid way, which enables not only inter-view prediction (that utilizes adaptive filters) but also conventional H.264/AVC (intra-view) modes: inter prediction and intra prediction. The MBs or MB partitions for which intra-view modes are selected cannot benefit from the adaptive filter and thus allowing them to be considered for the adaptive filter generation can lead to a less optimal filter, which is less sensitive to the prediction errors of those samples finally predicted by inter-view prediction. So, the set of samples,

belonging to relevant regions, need to be well defined before the optimization equation is built.

Two solutions are developed in order to select the pixels used for optimization: MB distortion thresholding and multi-pass encoding.

1) MB distortion thresholding

To guarantee that the adaptive filter is generated only by the prediction errors of the MBs for which inter-view prediction mode is preferred, the following function is proposed to select the MBs for the filter optimization.

$$f(MB_t) = \begin{cases} 1 & Distortion(MB_t) \le T \\ 0 & otherwise \end{cases} \quad (12)$$

wherein $f(\cdot)$ equals to 1 indicates that the t-th MB is selected for generating the adaptive filter and $Distortioin(\cdot)$ returns the MSE (Mean Square Error) distortion between the original signal and the predicted signal of an MB [P9]. So the set $\mathbf{S}^i$ are further selected as follows:

$$\mathbf{S^i} = \{S_t^j | S_t^j \epsilon MB_t, f(MB_t) = 1, v_t \epsilon V_i\}, i = 1 \dots K \quad (13)$$

The threshold T is content dependent and can vary from picture to picture.

T is set as follows to satisfy:

$$Rate = |D|/NumMB\ , D = \{MB_t | f(MB_t) = 1\} \quad (14)$$

wherein $|\cdot|$ stands for the cardinality of a set. When the Rate (percentage) of MBs that are used for inter-view prediction is decided, the threshold T can be decided by ordering the distortion values of all the MBs in a picture.

2) Multi-pass encoding

If multi-pass encoding is adopted, those MBs that used the inter-view prediction in the first encoding pass are selected to form the relevant regions. The pixel set S is a collection of all the pixels in those MBs and those pixels are then further clustered into a different sample set.

Locating the Corresponding Group of Pixels

For each pixel, its corresponding group of pixels is selected and their values form a vector. The convolution of the vector and the adaptive filter is the predicted value for that pixel.

The best matching sample in the inter-view picture (named center sample) for a pixel can be located by adding the scaled disparity motion vector to the sample position. Two types of non-separable filters are proposed. Type-I (3x3) filter requires a corresponding group of pixels that has only the center sample and the nearest samples with the same parity (odd or even) as the center sample, as shown in Fig. 28 with upper case letters. Type-II (5x5) filter requires a corresponding group of pixels that includes also the samples with different parity, i.e., the center sample as well as all the nearby integer samples, with both upper and the lower case letters in Fig. 28. The values of the pixels shown in Fig. 28 form vector U_p^i corresponding to the value b_p^i that is to be predicted.

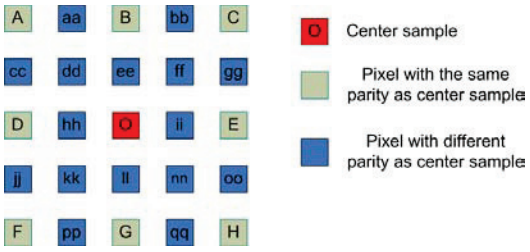


Fig. 28: Corresponding group of pixel with 9 and 25 samples.

5.3.4. Decoder of RAF and Complexity Comparisons

The new regionally adaptive filtering algorithm as a coding tool requires encoder methods which increase the encoder complexity. However, in many (if not most) of the application scenarios as discussed in Chapter 3 and illustrated in Fig. 14, MVC bitstreams are pre-encoded. For such applications, e.g. local playback and streaming, the decoder complexity is the most important factor to be considered when designing decoding algorithms. In this section, the decoder complexity of the conventional asymmetric coding, DMC and the adaptive filtering approaches are discussed.

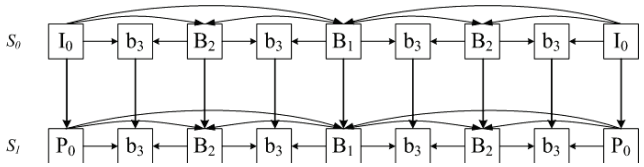


Fig. 29: Typical prediction structure for stereoscopic video.

A typical prediction structure for stereoscopic video is shown in Fig. 29, view 1 (S1) pictures are with quarter resolution and view 0 (S0) pictures are coded with the original resolution. Our discussions on complexity utilize this as an example. Note, in asymmetric stereoscopic video, the pictures in view 1 (S1) will have quarter resolution of view 0 (S0).

Complexity of Conventional Asymmetric Stereoscopic Video

In conventional asymmetric MVC, extra downsampling processes are required either for the whole picture or on-the-fly, which requires significantly more computations. In case of downsampling the whole inter-view picture once, the complexity increase is limited. The whole decoder complexity for the low-resolution view is slightly higher than H.264/AVC decoding with the same resolution.

Complexity of Direct Motion Compensation

In H.264/AVC, interpolation of half-sample luma values is the major part of the entire interpolation process at the decoder, and takes around 40% of the decoder execution time. The results in [103] are for the Baseline profile. This decoder execution time is applicable for the Main profile or the High profile. In these profiles, since bi-predicted (B) pictures are present, which is the case in hierarchical B prediction structure as shown in Fig. 29, although

CABAC requires more computations than CAVLC, the interpolation process will require 40% or even higher of the whole decoding complexity.

These interpolation computations are saved for anchor pictures in the proposed DMC approach. For non-anchor pictures, on the average, approximately 40% complexity reduction can be expected for MBs coded with inter-view prediction modes. Therefore, there is significant decoder complexity reduction for DMC compared with H.264/AVC decoding with the same resolution.

Decoder Realization and complexity of RAF Decoder

When applying RAF at the decoder, each fitter will filter the reconstructed inter-view picture and a new reference frame is generated. Different MB or MB partition can use reference picture list selection supported in H.264/AVC to select a different reference picture at the decoder, to enable RAF or GAF.

So at the decoder, only the filtering processes of the whole inter-view picture are required. Those processes introduce slightly complexity increase. After that, inter-view prediction is done as in the case of DMC.

The whole decoder complexity for the low-resolution view is slight higher than DMC and thus it is still significantly lower than H.264/AVC coding or conventional asymmetric coding.

5.3.5. Performance Assessment

As shown in [P8], the coding efficiency of the DMC method is almost the same, actually a bit better than the downsampling based methods. As reported in paper [P8], compared with the downsampling based methods, DMC has even slight coding gain, which is about 0.03 dB on average for all the tested sequences, for all the views and about 0.07 dB for the views with lower resolution. Both of them have a significant efficiency improvement, i.e., about 14% bitrate saving, for all views, compared with coding two sets of views with different resolutions separately. This also verifies that it is beneficial to use inter-view prediction for asymmetric MVC or stereoscopic video.

The GAF method proposed in [P9] gives an average bit-rate saving of 4.81% for all the tested sequence. However, it also introduces loss to some sequences, especially for *Akko&Kayo* and *Race1*, as the depth distributions in those sequences are more complex [P9].

The RAF method gives more than 10% bit-rate saving for three of the test sequences: *ballroom*, *rena* and *breakdancers*. The average bitrate saving is more than 8% for the low-resolution view [P10]. On average RAF approximately doubled the bitrate saving of the GAF. RAF improved the compression efficiency for the sequences for which GAF performed poorly, i.e., *Race1*, *Akko&Kayo* and *Exit*. Meanwhile, it further increases the gain for the other sequences significantly, such as *Ballroom* and *Breakdancers*.

Unlike most of the previous adaptive filter algorithms, which either apply to mono-view coding with one reference picture or can only be applied when inter-view prediction is

available [85], the proposed method works in a hybrid mode, where intra-view and inter-view coding are enabled, thus the modes provided by adaptive filtering must be competitive enough to be chosen. More simulation results and analysis can found in [P9][P10].

5.4. JOINT DEPTH AND TEXTURE CODING USING SVC

Depth-Image-Based Rendering (DIBR) is widely used for view synthesis in 3D video applications, such as 3D TV and free-viewpoint video. Compared with traditional 2D video applications, not only the texture video but also its associated depth map is required to be transmitted in a communication system that supports DIBR. A short introduction of DIBR will be given in sub-section 5.4.1. To efficiently utilize a limited transmission bandwidth, video coding algorithms, e.g. the H.264/AVC standard, can be adopted to compress the depth map as a monochromatic video.

However, when the correlation between the texture video and the depth map is exploited, the compression efficiency may be improved compared with coding them independently using H.264/AVC. A new encoder algorithm which employs Scalable Video Coding (SVC), the scalable extension of H.264/AVC, compresses the texture video, and its associated depth map was proposed in [P11].

5.4.1. Depth-Image-Based Rendering

DIBR renders a virtual view by warping the existing view(s). Such a rendering process is called view synthesis. Although differentiated in details, most of the view synthesis algorithms in a 3D transmission system are based on 3D warping, employing explicit or implicit geometry [58]. In both [104] and [105], depth map images are used for view synthesis. Those depth map images are referred to as explicitly geometry [106]. McMillian's approach emphasizes a non-Euclidean formulation of the 3D warping, which is normally under the condition that the acquisition camera parameters are unknown or the camera calibration is poor. Mark's approach, however, strictly follows Euclidean formulation, assuming the camera parameters for the acquisition and view interpolation are known.

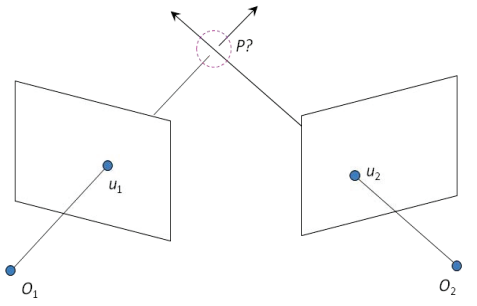


Fig. 30: Pixel re-projection for 3D warping.

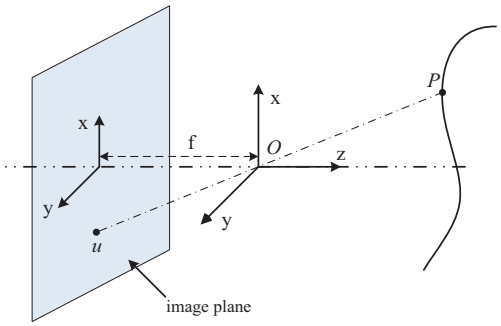


Fig. 31: Pinhole camera model.

The 3D geometry of 3D warping is shown in Fig. 30. Given the depth and the camera model, a pixel in $\overline{u_1}$ of a reference view is first projected from the 2D camera coordinate to

the point (voxel) P in the world-space coordinate. The point P is then projected to the destination view (the virtual view to be generated) along the direction of $\overrightarrow{PO_2}$, which corresponds to the view angle of the destination view. Assume the projected coordinate is $\overline{u_2}$, then the pixel values (in different color components) of the $\overline{u_1}$ in the reference view are considered as the pixel values for the $\overline{u_2}$ in the virtual view.

Projection from a point in the 3D world coordination system to the camera plane can be done by following a camera model, e.g., a pinhole camera model, which relies on intrinsic and extrinsic parameters. The extrinsic parameters define the position of the camera center and the camera's heading in world coordinates with the following transform:

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \boldsymbol{A} \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix} \quad (15)$$

Wherein $(x \quad y \quad z)^T$ is the coordinate in the 3D camera coordinate system and $(x_w \quad y_w \quad z_w)^T$ is the coordinate in the world coordinate system. **A** (4x4) is usually an orthonormal transform that can be illustrated as follows:

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{R} & \boldsymbol{T} \\ \boldsymbol{0} & \boldsymbol{1} \end{pmatrix} \quad (16)$$

wherein $\boldsymbol{R}$ is the 3x3 rotation matrix and $\boldsymbol{T}$ is the translation (although $\boldsymbol{T}$ is not the position of the camera). In the 3D camera coordinate system, the z axis is called principal optical axis, and x and y axis form the image plane. For example, as shown in Fig. 30, $\overrightarrow{O_1P}$ is the principal optical axis. The plane perpendicular to the principal optical axis and containing $\overline{u_1}$ is the image plane.

The world coordinate system can be defined as the same as the 3D camera coordinate system of a camera. In this case, $\boldsymbol{R} = \boldsymbol{I}$ and $\boldsymbol{T} = \boldsymbol{0}$. If a 3D camera coordinate system has only one translation from the world coordinate system, we have $(x \quad y \quad z)^T = (x_w \quad y_w \quad z_w)^T + \boldsymbol{T}^T$.

The intrinsic parameters specify the transformation from a 3D camera coordinate system to a 2D image plane. As shown in Fig. 31, wherein O is still the origin of the 3D camera coordinate system, which e.g., is the center of the camera (sensor) plane. In such a model, $\frac{u}{x} = \frac{v}{y} = \frac{-f}{z}$, wherein $-f$ is called the focal length and sometimes is denoted as f for simplicity; $(u,v)^T$ is the coordinates in the image plane.

Note that sometimes, more than one view can be considered as reference views. The projection from $\overline{u_1}$ to $\overline{u_2}$ itself is not always a one-to-one projection. When more than one pixel is projected to the destination pixel $\overline{u_2}$, visibility problem occurs [105]. To solve it, z-buffering or other interpolation methods can be adopted. The process of generating the pixel values based on multiple candidate reference pixels is typically named reconstruction in view synthesis [105]. When no pixel is projected to the destination pixel $\overline{u_2}$, on the contrary, a hole may exist at the picture of the virtual view. If there is a large area which has no pixels mapped to, the phenomenon is called occlusion. If the holes distributed sparsely in a picture,

they are called pinholes. Occlusion can be solved by introducing another reference view in a different direction. Pinhole filling usually takes neighboring pixels as candidates [105].

The following sub-sections present an algorithm to code the depth map video and texture video jointly. However, DIBR algorithms fall outside the scope of this thesis, and could be subject of future work.

5.4.2. Motion Correlation between Texture Video and Depth Map

Generally, a texture video is represented in YUV 4:2:0 format and a depth map is regarded as luma-only video in YUV 4:0:0 format. Fig. 32 is an example of a texture video frame and its associated depth map, taken from the *ballet* sequence provided by Microsoft [107]. Obviously, the intensity values of the texture and those of the depth map are less relevant. However, from Fig. 32, it can be observed that both the texture video and its associated depth map have similar object silhouette, so they should have similar object boundary and movement. To confirm this observation, the following experiment was performed.

Fig. 32: A texture video frame and its associated depth map.

The texture video and its associated depth map were coded with H.264/AVC, and their motion fields in the unit of the 4x4 block were extracted. Let $\vec{t}$ and $\vec{d}$ be the motion field of a specific frame of the texture video and its associated depth map, respectively. The correlation coefficient between the two motion fields is calculated as:

$$\rho = Cov(\vec{t},\vec{d}) / \sqrt{Var(\vec{t}) \cdot Var(\vec{d})} \qquad (17)$$

wherein

$$Cov(\vec{t},\vec{d}) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\left(\vec{t}_{m,n} - \bar{\vec{t}}\right)\cdot\left(\vec{d}_{m,n} - \bar{\vec{d}}\right) \qquad (18)$$

$$Var(\vec{t}) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\left\|\vec{t}_{m,n} - \bar{\vec{t}}\right\| \qquad (19)$$

$\bar{\vec{t}}$ and $\bar{\vec{d}}$ denote the mean motion vectors of $\vec{t}$ and $\vec{d}$, M/N is the picture height/width divided by 4, "·" denotes the inner product, and ||·|| denotes the norm operator. Note that equation (19) calculates the variance of the motion field of texture $\vec{t}$, and it can also be used to calculate the variance of the motion filed of depth map $\vec{d}$. The correlation coefficients were calculated for 100 frames in the Ballet test sequence, and the behavior of the correlation coefficient per frame is plotted in Fig. 33. It can be observed that, the texture video motion field and its associated depth map motion field are slightly correlated. Therefore, coding efficiency would be improved if one can efficiently use this correlation, at least for those blocks in the depth map that have similar motion vectors as those of the co-located blocks in the texture video.

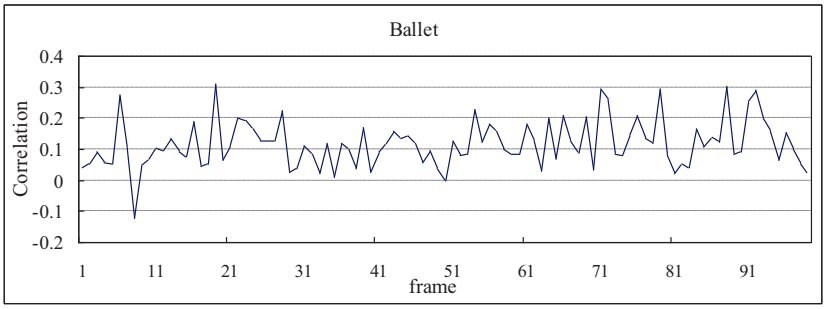


Fig. 33: Correlation coefficient of motion fields per frame.

5.4.3. Texture Video and Depth Map Compression Using SVC

A joint coding method was proposed to code the texture video and its associated depth map [P11]. In the proposed method, two layers are coded: the texture video is coded as the base layer, and the depth map is coded as the CGS enhancement layer. The texture video is coded using the same mechanism as H.264/AVC single layer coding, while the depth map is coded using SVC inter-layer motion prediction in addition to the single layer coding techniques.

It should be noted that the other two inter-layer prediction tools (inter-layer texture and residual predictions) are disabled in the proposed method. In the depth map motion estimation process, the conventional spatial MV predictor or the inter-layer MV predictor can be chosen for each MB in the enhancement layer. Furthermore, when the co-located MB in the base layer is inter coded, the MB in the enhancement layer can adaptively choose the base mode in addition to the conventional H.264/AVC modes in the mode decision process. After motion estimation and mode decision, transform and quantization are applied to the enhancement layer as in SVC.

The proposed method introduces significant gain, up to 0.97 dB, for the coding of depth maps of the test sequences, compared with coding depth maps as an independent bitstream, as the proposed method reduces the bits for the motion vectors. It is also shown that other

inter-layer prediction tools, which required much more computations than inter-layer motion prediction, cannot help in such a scenario. For more details refer to [P11].

Chapter 6

Conclusions

H.264/AVC has been a successful video coding standard and in recent years, extensions of the standard to scalable video and multiview video have been standardized. Scalable video targets video streaming applications over heterogeneous networks and devices. Multiview video coding codes the 3D content representation for a real-world scene captured by multiple cameras. Those two extensions to H.264/AVC have been finalized as SVC and MVC, respectively.

The major contributions of this thesis are the techniques proposed to facilitate the scalable video and multiview video applications. One part of this work has been adopted into the standards or standard reference software modules, and the other part targets further enhancing the functionalities over the existing standards.

In Chapter 2, hierarchical P coding for SVC or AVC baseline profile was firstly proposed, providing temporal scalability as well as a high coding efficiency for bitstreams which are compliant to AVC baseline profile. Then bit-depth scalability coding based on the SVC standard was introduced. The approach enables bandwidth reduction for the scenario in the near future: 8-bit depth video and 10 or higher -bit depth video will both be desirable for the same content.

In Chapter 3, the MVC, as a video coding standard, was introduced, together with the technical contributions that were proposed by the author and his co-workers and adopted by the MVC standard. The contribution of these techniques focuses on better decoder resource control, bitstream adaptation, error resiliency, and view switching, to meet various requirements for the MVC applications.

In Chapter 4, in order to enable graceful degradation for SVC, error concealment methods were proposed. The methods can greatly improve the quality of the reconstructed video, to achieve better graceful degradation. In addition, the error concealment method for MVC was also described, also providing better video quality during the transmission of multiview video over a lossy channel.

In Chapter 5, coding algorithms for MVC and 3D Video were proposed and discussed. The proposed techniques include single-loop decoding for MVC, asymmetric video coding algorithms based on the MVC standard and joint texture and depth map coding. Single-loop decoding avoids transmission and decoding of some pictures, without a noticeable coding efficiency sacrifice. Asymmetric coding codes some views with quarter resolution of the others. In stereoscopic video, it requires only a bandwidth comparable to 2D video, but with a subjective quality similar to 3D representation of two views with the original resolution. The proposed methods firstly much lowered the decoding complexity for the low-resolution view, and then decreased the bandwidth for the low-resolution view with adaptive filtering algorithms, which were used for the inter-view prediction from the high-resolution view to the low-resolution view. Another representation of 3D video is to have both texture video and depth maps for a view. It was proposed to use SVC to jointly code the 3D video represented by texture and its associated depth map.

In summary, this thesis discussed the technologies in SVC and MVC standards and methods to further benefit the applications for scalable video and multiview video. With the deployment of the SVC and MVC standards, the proposed techniques are expected to be widely used by industry, in this field, and are becoming important references for other relevant academic research.

Bibliography

[1] ITU-T Recommendation H.262 (02/2000) | ISO/IEC International Standard 13818-2:2000, "Information Technology – Generic Coding of Moving Pictures and Associated Audio Information: Video."

[2] ISO/IEC International Standard 14496-2:2001, "Information Technology – Coding of Audio-Visual Objects – Part 2: Visual."

[3] ITU-T Recommendation H.264 (05/2003), "Advanced Video Coding for Generic Audio-Visual Services."

[4] T. Wiegand, G.J. Sullivan, G. Bjøntegaard and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 645–656, Jul. 2003.

[5] H.S. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, "Low-complexity transform and quantization in H.264/AVC," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 598–603, Jul. 2003.

[6] M. Flierl and B. Girod, "Generalized B pictures and the draft JVT/H.264 video compression standard," *IEEE Transactions Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 587–597, Jul. 2003.

[7] P. List, A. Joch, J. Lainema, G. Bjøntegaard, and M. Karczewicz, "Adaptive deblocking filter," *IEEE Transactions Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 614–619, Jul. 2003.

[8] S. Wenger, "H.264/AVC over IP," *IEEE Transactions Circuits and Systems for Video Technology*, vol. 13, no.7, pp. 645–656, Jul. 2003.

[9] J. M. Boyce, "Weighted prediction in the H.264/MPEG AVC video coding standard," *IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 3, May 2004, pp. III-789–92.

[10] M. Karczewicz and R. Kurceren. "The SP- and SI-frames design for H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 637–644, Jul. 2003.

[11] D. Marpe, H. Schwarz, and T.Wiegand, "Context-adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no, 7, pp. 620–636, Jul. 2003.

[12] G. J. Sullivan, P. Topiwala, and A. Luthra, "The H.264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions," *SPIE Conference on Applications of Digital Image Processing*, vol. 5558, 454, 2004.

[13] ITU-T Recommendation H.264 (11/2007), "Advanced Video Coding for Generic Audiovisual Services."

[14] M. Wien, "Variable Block-Size Transforms for H.264/AVC", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 604-613, Jul. 2003.

[15] G. J. Sullivan, H. Yu, S. Sekiguchi, H. Sun, T. Wedi, S. Wittmann, Y. Lee, A. Segall, and T. Suzuki, "New Standardized Extension of MPEG-4 AVC/H.264 for Professional-quality Video Applications," *IEEE International Conference on Image Processing, ICIP* 2007, San Antonio, USA, Sept. 2007, pp. I.13-16.

[16] Y.-L. Lee, K.-H. Han, and G. J. Sullivan, "Improved Lossless Intra Coding for H.264/MPEG-4 AVC", *IEEE Transactions on Image Processing*, vol. 15, no. 9, pp. 2610-2615, Sept. 2006.

[17] Y.-K. Wang, M. M. Hannuksela, S. Pateux, A. Eleftheriadis, and S. Wenger, "System and Transport Interface of SVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1149–1163, Sept. 2007.

[18] W. Thomas and H. G. Musmann, "Motion-and Aliasing-Compensated Prediction for Hybrid Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 13, no. 7, pp. 577- 586 Jul. 2003.

[19] J. Postel, IETF STD 6 (RFC 0768), "User Datagram Protocol," Aug. 1980.

[20] Y. Wang, and Q. Zhu, "Error Control and Concealment for Video Communication: A Review," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 974-997, May 1998.

[21] Y. Wang, J. Ostermann, and Y.-Q. Zhang, "Video Processing and Communications," *Prentice Hall*, 2002.

[22] ITU-T Recommendation H.263, "Video Coding for Low Bit Rate Communication," Version 1: Nov. 1995, Version 2: Jan. 1998, Version 3: Nov. 2000.

[23] M. M. Hannuksela, Y.-K. Wang, and M. Gabbouj, "Isolated Regions in Video Coding," *IEEE Transactions on Multimedia,* vol. 6, no. 2, pp. 259–267, Apr. 2004.

[24] Y.-K. Wang, M. M. Hannuksela, and M. Gabbouj, "Error Resilient Video Coding using Unequally Protected Key Pictures," *2003 International Workshop on Very Low Bitrate Video (VLBV 2003)*, Madrid, Spain, Sept. 2003, pp. 290-297.

[25] S. Rane, P. Baccichet, and B. Girod, "Modeling and Optimization of a Systematic Lossy Error Protection System Based on H.264/AVC Redundant Slices," *Picture Coding Symposium, PCS'06,* Beijing, China, Apr. 2006.

[26] I. Radulovic, Y.-K. Wang, S. Wenger, A. Hallapuro, M. Hannuksela, and P. Frossard, "Multiple Description H.264 Video Coding with Redundant Pictures," in *Proceedings of Mobile Video Workshop, ACM Multimedia '07*, Augsburg, Germany, Sept. 2007, pp.37–42.

[27] T. Tillo, M. Grangetto, and G. Olmo, "Redundant Slice Optimal Allocation for H.264 Multiple Description Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol.18, no.1, pp.59–70, Jan. 2008.

[28] C. Zhu, Y.-K. Wang, and H. Li, "Adaptive Redundant Picture for Error Resilient Video Coding," *IEEE International Conference on Image Processing, ICIP'07*, San Antonio, TX, USA, Sept. 2007, pp. IV.253–256.

[29] D. Tian, M. M. Hannuksela, Y. -K. Wang and M. Gabbouj, "Error Resilient Video Coding Techniques Using Spare Pictures," *International Packet Video Workshop*, Nantes, France, Apr. 2003.

[30] Y.-K. Wang, M. Hannuksela, K. Caglar, and M. Gabbouj, "Improved Error Concealment Using Scene Information," *International Workshop on Very Low Bitrate Video*, Madrid, Spain, Sept. 2003, pp.283–289.

[31] G. Cote, and F. Kossentini, "Optimal Intra Coding of Blocks for Robust Video Communication over the Internet," *Signal Processing: Image Communication*, vol. 15, pp. 25–34, Sept. 1999.

[32] Q. Zhu, and L. Kerofsky, "Joint Source Coding, Transport Processing and Error Concealment for H.323-based Packet Video," *SPIE Visual Communications and Image Processing, VCIP'99*, vol. 3653, San Jose, CA, Jan. 1999, pp. 52–62.

[33] R. Zhang, S. L. Regunathan, and K. Rose, "Video Coding with Optimal Inter/Intra-mode Switching for Packet Loss Resilience," *IEEE Journal Select. Areas Communication*, vol. 18, pp. 966–976, Jun. 2000.

[34] H. Yang and K. Rose, "Recursive end-to-end Distortion Estimation with Model-based Cross-correlation Approximation," *IEEE International Conference on Image Processing, ICIP'*03, vol. 3, Barcelona, Spain, Sept. 2003, pp. 469–472.

[35] T. Stockhammer, D. Kontopodis, and T. Wiegand, "Rate-distortion Optimization for JVT/H.26L Coding in Packet Loss Environment," *International Packet Video Workshop*, Pittsburgh, USA, Apr. 2002.

[36] Y. Zhang, W. Gao, H. Sun, Q. Huang, and Y. Lu, "Error Resilience Video Coding in H.264 Encoder with Potential Distortion Tracking," *IEEE International Conference on Image Processing*, *ICIP'*04, vol. 1, Singapore, Oct. 2004, pp. 163–166.

[37] Y.-K. Wang, M. Hannuksela, V. Varsa, A. Hourunranta, and M. Gabbouj, "The Error Concealment Feature in the H.26L Test Model," *IEEE International Conference on Image Processing*, *ICIP'02*, Singapore, Sept. 2002, pp. 729–732.

[38] Z. Wu and J. Boyce, "An Error Concealment Scheme for Entire Frame Losses based on H.264/AVC," *IEEE International Symposium on Circuits and Systems*, *ISCAS'06*, May 2006, pp. 4463–4466.

[39] T. Wiegand, G. Sullivan, J. Reichel, H. Schwarz, and M. Wien (eds.), "Joint Draft 11 of SVC Amendment," *JVT-X201*, *Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG,* Geneva, Switzerland, 29 Jun. – 5 Jul. 2007.

[40] S.-J. Choi and J. W. Woods, "Motion-compensated 3-D Subband Coding of Video," *IEEE Transactions on Image Processing*, vol. 8, no. 2, pp. 155–167, Feb. 1999.

[41] B. Pesquet-Popescu and V. Bottreau, "Three-dimensional Lifting Schemes for Motion-compensated Video Compression," in *IEEE International Conference on Acoustic, Speech, and Signal Processing, ICASSP'01*, pp. 1793–1796, Salt Lake City, Utah, USA, May 2001.

[42] L. Luo, J. Li, S. Li, Z. Zhuang, and Y.-Q. Zhang, "Motion Compensated Lifting Wavelet and its Application in Video Coding," in *IEEE International Conference on Multimedia and Expo ICME'01*, Tokyo, Japan, Aug. 2001, pp. 365–368.

[43] A. Secker and D. Taubman, "Motion-compensated Highly Scalable Video Compression Using an Adaptive 3D Wavelet Transform based on Lifting," in *IEEE International Conference on Image Processing, ICIP'01*, vol. 2, Greece, Oct. 2001, pp. 1029–1032.

[44] J.M. Shapiro, "Embedded Image Coding Using Zero Trees of Wavelet Coefficients," *IEEE Transactions on Signal Processing,* vol.40, no.12, pp.3445–3462, Dec. 1993.

[45] D. Taubman, "High Performance Scalable Image Compression with EBCOT," *IEEE Transactions on Image Processing*, vol. 9, no. 7, pp. 1158–1170, Jul. 2000.

[46] W. Li, "Overview of Fine Granularity Scalability in MPEG-4 Video Standard," *IEEE Transactions on Circuits and System for Video Technology,* vol. 11, no. 3, pp. 301–317, Mar. 2001.

[47] D. Tian, M.M. Hannuksela, and M. Gabbouj, "Sub-sequence Video Coding for Improved Temporal Scalability," in *IEEE International Symposium on Circuits and Systems, ISCAS'05*, Kobe, Japan, May 2005, pp. 6074–6077.

[48] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of Scalable Video Coding Extension of H.264/AVC Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol.17, no.9, pp.1103–1120, Sept. 2007.

[49] H. Schwarz, T. Hinz, D. Marpe, and T. Wiegand, "Constrained Inter-layer Prediction for Single-loop Decoding in Spatial Scalability," in *IEEE International Conference on Image Processing, ICIP'05*, Genova, Italy, Sept. 2005.

[50] P. Amon, T. Rathgen, and D. Singer, "File Format for Scalable Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol.17, no.9, pp.1174–1185, Sept. 2007.

[51] S. Wenger, Y.-K. Wang, and T. Schierl, "Transport and signaling of SVC in IP networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol.17, no.9, pp.1164–1173, Sept. 2007.

[52] Y.-K. Wang, M. M. Hannuksela, S. Pateux, A. Eleftheriadis, and S. Wenger, "System and Transport Interface of SVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol.17, no.9, pp.1149–1163, Sept. 2007.

[53] H. Schwarz, D. Marpe, and T. Wiegand, "Analysis of Hierarchical B Pictures and MCTF," *IEEE International Conference on Multimedia and Expo, ICME'06*, Beijing, China, Jul. 2006, pp.1929–1932.

[54] K. Devlin, A. Chalmers, A. Wilkie, and W. Purgathofer. "STAR Report on Tone Reproduction and Physically Based Spectral Rendering," in *EUROGRAPHICS 2002*. DOI: 10.1145/1073204.

[55] "Joint Draft 8.0 on Multiview Video Coding," *JVT-AB204, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG,* Hannover, Germany, Jul. 2008.

[56] ITU-T Recommendation H.264 (03/2009), "Advanced Video Coding for Generic Audiovisual Services."

[57] A. Smolic, H. Kimata, and A. Vetro, "Development of MPEG Standards for 3D and Free Viewpoint Video," *SPIE International Symposium on Optics East*, Oct. 2005.

[58] H.-Y. Shum, S.B. He, and S.-C. Chan, "Survey of Image-based Representations and Compression Techniques," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 11, pp. 1020–1037, Nov. 2003.

[59] C. Fehn, "Depth-Image-Based Rendering (DIBR), Compression and Transmission for a New Approach on 3D-TV", *SPIE Stereoscopic Displays and Virtual Reality Systems* XI, San Jose, CA, USA, Jan. 2004, pp. 93–104.

[60] H. Kimata, M. Kitahara, K. Kamikura, Y. Yashima, T. Fujii, and M. Tanimoto, "System Design of Free Viewpoint Video Communication," *International Conference on Computer and Information Technology, CIT,* New York, USA, Jun. 2004.

[61] A. Smolic and P. Kauff, "Interactive 3-D Video Representation and Coding Technologies," *Proceedings of the IEEE,* vol. 93, no. 1, pp. 98–110, 2005.

[62] A. Vetro, W. Matusik, H. Pfister, and J. Xin, "Coding Approaches for end-to-end 3D TV systems," *Picture Coding Symposium, PCS'04*, San Francisco, CA, USA, Dec. 2004, pp. 319–324.

[63] C. Fehn, R. de la Barré, and S. Pastoor, "Interactive 3-DTV-concepts and Key Technologies," *Proceedings of the IEEE*, vol. 94, no. 3, pp. 524–538, Mar. 2006.

[64] J. G. Eden, "Information Display Early in the 21st Century: Overview of Selected Emissive Display Technologies," *Proceedings of the IEEE*, vol. 94, no. 3, pp. 567–574, 2006.

[65] B. Fröba and C. Küblbeck, "Face Detection and Tracking Using Edge Orientation Information," *SPIE Visual Communications and Image Processing, VCIP 2001*, vol. 4310, San Jose, CA, USA, Jan. 2001, pp. 583–594.

[66] L. Young and D. Sheena, "Methods & Designs: Survey of Eye Movement Recording Methods," *Behavior Research Methods & Instrumentation*, vol. 7, no. 5, pp. 397–429, 1975.

[67] "Requirements on Multi-view Video Coding v.5," *ISO/IEC JTC1/SC29/WG11, MPEG Doc N7539*, MPEG, Nice, France, Oct. 2005.

[68] "Joint Multiview Video Model (JMVM) 1.0," *JVT-T208*, *Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG,* Klagenfurt, Austria, Jul. 2006.

[69] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Efficient Prediction Structures for Multiview Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 11, pp. 1461–1473, Nov. 2007.

[70] ISO/IEC IS 14496-2, "Information Technology – Coding of Audio-Visual Objects – Part 12: ISO Base Media File Format," 2005.

[71] ISO/IEC 14496-15:2004/FPDAM 3, "Information Technology — Coding of Audio-Visual Objects — Part 15: Advanced Video Coding (AVC) File Format, Amendment 3: File Format Support for Multiview Video Coding," 2009.

[72] A. Vetro and S. Yea, "Comments on MVC Reference Picture Marking," *JVT-U062*, *Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG,* Hangzhou, China, Oct. 2006.

[73] P. Pandit, Y. Su, P. Yin, and C. Gomila, "Comments on High-level Syntax for MVC," *JVT-U026, Joint Video Team* (*JVT) of ISO/IEC MPEG & ITU-T VCEG,* Hangzhou, China, Oct. 2006.

[74] Y. -K. Wang, Y. Chen, and M.M. Hannuksela, "Non-required Video Component SEI message for MVC," *JVT-AA034, JVT of ISO/IEC MPEG & ITU-T VCEG*, Geneva, Apr. 2008.

[75] Y. Chen, Y. -K. Wang, and M.M. Hannuksela, "View Dependency Change SEI message and Operation Point not Present SEI Message for MVC," *JVT-AA035, JVT of ISO/IEC MPEG & ITU-T VCEG,* Geneva, Apr. 2008.

[76] U. Ugur, H. Liu, J. Lamina, M. Gabbouj, and H. Li, "Parallel Encoding - Decoding Operation for Multiview Video Coding with High Coding Efficiency," *3DTV Conference*, Kos Island, Greece, May 2007.

[77] A. Vetro, S. Yea, W. Matusik, H. Pfister, and M. Zwicker, "Anti-aliasing for 3D displays," Apr. 2007, *JVT-W060, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG,* San Jose, CA, USA.

[78] J. Jia, H. Kim, H. Choi, et. al., "Implementation of Redundant Pictures in JSVM," *JVT-Q054*, *Joint Video Team (JVT)of ISO/IEC MPEG & ITU-T VCEG,* Nice, France, Oct. 2005.

[79] C. He, H. Liu, H. Li, Y.-K. Wang, and M.M. Hannuksela, "Redundant Picture for SVC," *JVT-W049, Joint Video Team (JVT)of ISO/IEC MPEG & ITU-T VCEG,* San Jose, USA, Apr. 2007.

[80] Y. Guo, Y.-K. Wang, and H. Li, "Error Resilience Mode Decision in Scalable Video Coding," *IEEE International Conference on Image Processing, ICIP'06*, Atlanta, USA, Oct. 2006, pp. 2225–2228.

[81] A. Eleftheriadis, S. Cipolli, and J. Lennox, "Improved *Error Resilience Using* Temporal Level 0 Picture Index," *JVT-W062*, *Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG,* San Jose, CA, USA, Apr. 2007.

[82] J. Zheng and L. P. Chau, "A Temporal Error Concealment Algorithm for H.264 Using Lagrange Interpolation," *IEEE International Symposium on Circuits and Systems, ISCAS'04,* pp. 133–136, Vancouver, Canada, May 2004.

[83] L. W. Kang and J. J. Leou, "A Hybrid Error Concealment Scheme for MPEG-2 Video Transmission based on Best Neighborhood Matching Algorithm," *Journal Visual Communication and Image Representation*, vol. 16, issue 3, pp.288-310, Jan. 2005.

[84] "Joint Multiview Video Model (JMVM) 8.0," *JVT-AA207*, *JVT of ISO/IEC MPEG & ITU-T VCEG,* Geneva, Switzerland, Apr. 2008.

[85] J. H. Kim, P. Lai, J. Lopez, A. Ortega, Y. Su, P. Yin, and C. Gomila, "New Coding Tools for Illumination and Focus Mismatch Compensation in Multiview Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 11, pp. 1519–1535, Nov. 2007.

[86] "Common Test Conditions for Multiview Video Coding," *JVT-T207, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG,* Klagenfurt, Austria, Jul. 2006.

[87] "JMVM 5 software," *JVT-X208*, *Joint Video Team* (*JVT) of ISO/IEC MPEG & ITU-T VCEG,* Geneva, Switzerland, Jun.-Jul. 2007.

[88] B. Julesz, "Foundations of Cyclopean Perception," *University of Chicago Press*, Chicago, IL, USA, 1971.

[89] I. Dinstein, M. G. Kim, J. Tselgov, and A. Henik, "Compression of Stereo Images and the Evaluation of its Effects on 3-D Perception," *SPIE Proc. Applications of Digital Image Processing XII*, vol. 1153, 1989, pp. 522–530.

[90] S. Pastoor, "3D-television: A Survey of Recent Research Results on Subjective Requirements," *Signal Processing: Image Communication*, vol. 4, no. 1, pp. 21–32, 1991.

[91] S. Yano and Y. Yuyama, "Stereoscopic HDTV: Experimental System and Psychological Effects," *SMPTE Journal*, vol. 100, pp. 14–18, 1991.

[92] M. G. Perkins, "Data Compression of Stereopairs," *IEEE Transactions on Communication*, vol. 40, pp. 684–696, Apr. 1992.

[93] L. Stelmach, W. Tam, D. Meegan, and A. Vincent, "Stereo Image quality: effects of mixed spatio-temporal resolution," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 10, no. 2, pp.188–193, Feb. 2000.

[94] W. D. Reynolds and R. V. Kenyon, "The Wavelet Transform and the Suppression Theory of Binocular Vision for Stereo Image Compression," *IEEE International Conference on Image Processing, ICIP 1996*, vol. 1, Lausanne. Switzerland, Sept. 1996, pp. 557–560.

[95] A. Aksay, C. Bilen, E. Kurutepe, T. Ozcelebi, G. B. Akar, M. R. Civanlar, and A. M. Tekalp, "Temporal and Spatial Scaling For Stereoscopic Video Compression," *in Proc. of 14th European Signal Processing Conference*, Florence, Italy, Sept. 2006.

[96] C. Fehn, P. Kauff, S. Cho, H. Kwon, N. Hur, and J. Kim, "Asymmetric Coding of Stereoscopic Video for Transmission over T-DMB," *Proc. 3DTV Conference*, Kos Island, Greece, May 2007.

[97] "MPEG-4 Video Verification Model Version 18.0," *ISO/IEC JTC1/SC29/WG11, MPEG Doc. N3908*, Jan. 2001.

[98] L. M. J. Meesters, W. A. IJsselsteijn, and P. J. H. Seuntiens, "A Survey of Perceptual Evaluations and Requirements of Three-dimensional TV, " *IEEE Transactions on Circuits and Systems for Video Technology*, no. 3, pp. 381–391, 2004.

[99] D. V. Meegan, L. B. Stelmach, and W. J. Tam, "Unequal Weighting of Monocular Inputs in Binocular Combination: Implications for the Compression of Stereoscopic Imagery," *Journal of Experimental Psychology:* Appl., vol. 7, pp. 143–153, 2001.

[100] H. Kimata and S. Shimizu, "Inter-view Prediction with Down-Sampled Reference Pictures," *JVT-W079*, *JVT of ISO/IEC MPEG & ITU-T VCEG,* San Jose, CA. USA, Apr. 2007.

[101] H. Kimata and S. Shimizu, "Experimental Results on Down-sampled Inter-view Prediction for MVC," *JVT-Y030*, *JVT of ISO/IEC MPEG & ITU-T VCEG,* Shenzhen, China, Oct. 2007.

[102] S. P. Lloyd, "Least Squares Quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129-137, 1982.

[103] V. Lappalainen, A. Hallapuro, and T.D. Hämäläinen, "Complexity of Optimized H.26L Video Decoder Implementation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 717–725, Jul. 2003.

[104] L. McMillan, "An Image-Based Approach to Three-Dimensional Computer Graphics," PhD thesis, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 1997.

[105] W. R. Mark, "Post-Rendering 3D Image Warping: Visibility, Reconstruction, and Performance for Depth-Image Warping. PhD thesis," University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, Apr.1999.

[106] D. Scharstein and R. Szeliski, “A Taxonomy and Evaluation of Dense Two-frame Stereo Correspondence Algorithms,” *International Journal of Computer Vision*, 47: pp. 7–42, Apr.-Jun. 2002.

[107] C.L. Zitnick, S.B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, “High-quality Video View Interpolation Using A Layered Representation,” *Proc. of ACM SIGGRAPH*, Aug. 2004, pp. 600–608.

Publications

[P1] W. Wan, Y. Chen, Y.-K. Wang, M. M. Hannuksela, H. Li, and M. Gabbouj, "Efficient Hierarchical Inter Picture Coding for H.264/AVC Baseline Profile," *Picture Coding Symposium, PCS*'09, Chicago, Illinois, USA, May 6-8, 2009.

EFFICIENT HIERARCHICAL INTER PICTURE CODING FOR H.264/AVC BASELINE PROFILE

Weixing Wan[1], Ying Chen[2], Ye-Kui Wang[3], Miska M. Hannuksela[3], Houqiang Li[1], and Moncef Gabbouj[2]

[1]University of Science and Technology of China
[2]Department of Signal Processing, Tampere University of Technology
[3]Nokia Research Center

ABSTRACT

Bi-predictive (B) slices are not supported in the Baseline profile of the Advanced Video Coding (H.264/AVC) standard, which results in a decreased coding efficiency compared with other profiles supporting B slices. However, many application standards, such as the mobile multimedia services specified by the Third Generation Partnership Project (3GPP), use only the Baseline profile for H.264/AVC. Therefore, it is worth investigating H.264/AVC coding when only intra (I) and inter (P) slices are supported. In this paper, a content-adaptive Quantization Parameter (QP) cascading scheme for the hierarchical P coding method compatible with Baseline profile of H.264/AVC is proposed. The proposed method is based on a picture-level QP optimization. The proposed method has a significantly better rate-distortion performance than the traditional IPPP coding structure and outperforms hierarchical P coding methods using fixed delta QP settings between temporal levels noticeably with up to 0.53 dB gain in average luminance Peak Signal-to-Noise Ratio (PSNR).

Index Terms—H.264/AVC Baseline profile, Hierarchical P, Quantization Parameter

1. INTRODUCTION

The Advanced Video Coding (H.264/AVC) standard [1][2] was developed by the Joint Video Team (JVT). Using state-of-the-art coding tools, H.264/AVC achieves a significant improvement in terms of Rate-Distortion (RD) performance compared with earlier standards. It specifies several profiles targeted for different application environments and trade-offs between compression efficiency and computational complexity. The discussions in this paper focus on the compression efficiency of the Baseline profile.

The work of Weixing Wan and Houqiang Li is supported by NSFC General Program under contract No. 60672161, 863 Program under contract No. 2006AA01Z317, and NSFC Key Program under contract No. 60632040. The work of Ying Chen is partly supported by the Nokia Foundation Award granted by Nokia Research Center.

The Baseline profile of H.264/AVC does not support bi-predictive (B) slices, in which a picture may be predicted by two reference picture lists and thus each sub-macroblock partition may be predicted from two reference pictures. In other words, only intra (I) slices and inter (P) slices are supported by the Baseline profile. Using B slices, an average of 0.5-1 dB Peak Signal-to-Noise Ratio (PSNR) gain can be achieved by adjusting the Quantization Parameter (QP) of the B slices [3]. Compared with many other profiles, the Baseline profile does not support Context-based Adaptive Binary Arithmetic Coding (CABAC), which provides a bit-rate reduction between 5%–15% compared with Context-based Adaptive Variable Length Coding (CAVLC) [1], the only entropy coding tool supported by the Baseline profile.

Regardless of the absence of the support for B slices and CABAC, the Baseline profile has been chosen as the only supported H.264/AVC profile into many application standards, e.g., the Third Generation Partnership Project (3GPP) multimedia services [4][5][6]. Therefore, it is important to investigate how to improve the coding efficiency when only I and P slices are allowed.

The hierarchical B coding structure has been demonstrated as an effective tool for improving coding efficiency, e.g. compared with the traditional IBBP coding structure [7]. In this structure, the importance of pictures at each temporal level differs because of the hierarchically structured temporal prediction chain. Therefore, improved bit-rate saving under the same quality constraint can be achieved by using higher QP values for higher temporal levels. In [7], Schwarz et al. proposed a QP setting method which fixed the difference of QP between each pair of temporal levels without considering the content characteristics change across sequences or pictures. In [8], an exhaustive search method to get the best QP for each temporal level was presented. However, the encoding complexity is much higher than for the method described in [7] because of the exhaustive search. The above methods use picture-level QP optimization, which selects one constant QP value for a whole slice.

In this paper, a hierarchical P coding structure similarly to the hierarchical B coding structure is used for coding H.264/AVC Baseline profile compatible content. To

improve the coding efficiency further, we propose a picture-level content-adaptive QP cascading mechanism. In this mechanism, QP delta values in relative to the QP value of the highest temporal level pictures are decided based on the prediction modes of the Macroblocks (MBs).

The rest of this paper is organized as follows. In Section 2, an introduction to the hierarchical P coding structure is presented. Section 3 describes the content-adaptive QP cascading mechanism as well as two simple QP cascading mechanisms which are content independent. Experimental results are presented in Section 4, followed by conclusions in Section 5.

2. HIERARCHICAL P CODING STRUCTURE

One example of the hierarchical P coding structure (with 4 hierarchical/temporal levels) is shown in Fig. 1. The first picture of a video sequence is an Instantaneous Decoding Refresh (IDR) picture. A picture is called a key picture when all previously coded pictures also precede the picture in display order. As illustrated in Fig. 1, a key picture and all pictures that are temporally located between the current key picture and the previous key picture are considered as a Group Of Pictures (GOP). In the hierarchical P coding structure, multiple references in inter prediction are supported. The prediction relationship is as follows. The key pictures are either intra-coded or inter-coded using previous key pictures as references. The remaining pictures of a GOP are hierarchically predicted and it is possible to use pictures from the past or/and from the future in display order as references. Referring to Fig. 1, picture 1 refers to pictures 0 and 2 which means that the inter prediction reference of each MB or MB partition is either from picture 0 or picture 2, however any MB or MB partition can not simultaneously predicted from both picture 0 and 2, as only one reference picture list (list 0) is constructed.

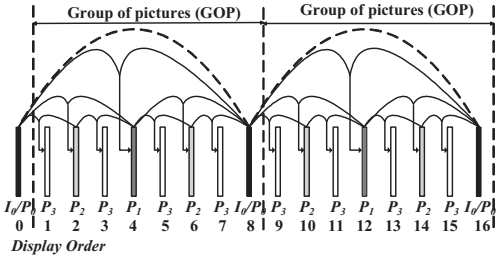


Fig. 1. Dyadic hierarchical P coding with 4 temporal levels

It should be noted that the usage of hierarchical P coding structure can be more flexible than the dyadic structure shown in Fig. 1. Typically to support temporal scalability for each temporal level, a picture is predicted from pictures of a lower or the same temporal level.

3. CONTENT-ADAPTIVE QP CASCADING

Let T be the total number of temporal levels and the QP value for the pictures in the highest temporal level, denoted as the QP_{T-1}, is set according to the desired target bit-rate, and the QP value for one lower temporal level picture is set according to a delta QP value and QP_{T-1}. In this section, we firstly describe the proposed content-adaptive QP cascading mechanism. For comparison, two content independent methods are also introduced.

3.1. Content-Adaptive QP Cascading

The difference between the QP value of a picture in a temporal level lower than T and the input QP value QP_{T-1} is decided by a scaling-factor, which depends on the prediction modes of the MBs in the picture.

In the hierarchical temporal prediction structures, the motion-compensation prediction can be expressed as the high-pass filtering along the motion trajectory with filter {1, -1} when using inter prediction, or with filter {-1/2, 1, -1/2} when using bi-prediction [9]. The scaling-factor is to balance the residual energies of the whole picture in contrast to the energy of the pictures in a higher temporal level, and thus controls the QP value of the picture. The scaling-factor is derived as the weighted average of the relative energy increase caused by the filtering process which is actually performed during inter predicted motion compensation prediction. After motion estimation, an energy factor of a picture can be calculated as in equation (1):

$$E_{t,m} = \frac{1}{N}\sum_{i=1}^{N}\alpha_i \qquad (1)$$

where t, ranging from 0 to $T-1$, inclusive, denotes the temporal level, m denotes the picture index within a temporal level t, N is the total number of MBs in a picture, and α_i represents the weighting factor of the relative energy of the i-th MB during motion compensation. Note that for simplicity, we assume all MB partitions (if more than one) in an MB are treated with the same intra, P or B modes.

The weighting factor α_i of an MB depends on the prediction mode of the MB. If an MB is an inter-P MB (with filter {1, -1}), the factor is $\sqrt{2}$. If it is an inter-B MB (with filter {-1/2, 1, -1/2}), the factor is $\sqrt{3/2}$. For an intra MB, the factor is 1. In the Baseline profile, there is no inter-B MB or MB partition. So, for each MB, if it is not intra-coded, all its MB partitions must be inter-P. So, in the H.264/AVC Baseline profile coding, only two factors are in use for all pixels of each MB: $\sqrt{2}$ and 1, so equation (1) holds for the hierarchical P coding.

After $E_{t,m}$ is obtained, the corresponding scaling-factor of the m-th picture with temporal level t is:

$$SF_{t,m} = \overline{SF_{t+1}} \div E_{t,m} \qquad (2)$$

where t is in the range of 0 to $T-1$, inclusive , $\overline{SF_{t+1}}$ is the average of $SF_{t+1,j}$ for all values of j within the current GOP. $SF_{T-1,j}$ is initialized to 1.0, for any j.

After the scaling-factor $SF_{t,m}$ is obtained, the QP value for the corresponding picture, denoted as $QP_{t,m}$, can be calculated as in equation (3):

$$QP_{t,m} = QP_{T-1} + 6\log_2 SF_{t,m} \quad (3)$$

where QP_{T-1} is the input QP value and $6\log_2 SF_{t,m}$ is the delta QP value. The final value of $QP_{t,m}$ is rounded and clipped to be an integer value in the range of 0 to 51, inclusive. Note that all the highest temporal level pictures have the same QP value in our method.

The prediction modes of MBs are unknown until the completion of the motion estimation and mode decision processes. Therefore, for each GOP of the target sequence to be encoded, it is first analyzed to find the MB prediction modes by performing the motion estimation and mode decision processes. Then the final QP value for each picture can be calculated as above mentioned.

3.2. Content Independent QP Cascading

Two methods that have fixed delta QP values among temporal levels for all sequences are presented as follows. Note that these two methods are not part of the proposed mechanism and are used only for comparison purposes.

- Fixed Scaling-Factor (FSFC): For each P picture, a fixed percentage of the MBs are assumed as P MBs. The final QP of a temporal level is still set as described by the equations above.
- Fixed QP Cascading (FQC): The delta QP between specific two adjacent temporal levels is set to a specific constant value.

4. EXPERIMENTAL RESULTS

The presented hierarchical P coding methods were implemented based on the SVC reference software JSVM_8_6 [10], which is capable of generating H.264/AVC compatible bitstreams. In our simulation, GOP size was set to 16 and the initial QP values for the highest temporal level pictures were 28, 32, 36 and 40, respectively.

A wide range of sequences were tested. They were "*Container*", "*Foreman*", "*Irene*", "*Mobile*", "*News*", "*Paris*", "*Silent*", and "*Tempete*". All the sequences were QCIF@30Hz. The following four coding scenarios were compared. Note that in these four scenarios, only the first picture of each sequence was coded using I slice

- Content-Adaptive QP Cascading (CAC): The method mentioned in Section 3.1. In the pre-processing of performing motion estimation and mode decision processes, the QP values for all the pictures were set in a simple way, in which the delta QP between two adjacent levels is set to 2.
- Traditional IPPP coding structure (IPPP): The GOP size was 1 and the number of reference pictures was equal to 2. That is, a sequence was coded as "I_0... $P_nP_{n+1}P_{n+2}$ ...", where I_0 was I picture, any other picture P_n, was a P picture. P_{n+2} referred to P_{n+1} and P_n, and so on. All pictures in a coded video sequence had the same QP value.
- FSFC: The percentage of P MBs in each picture was considered to be a fixed value. To achieve a good performance for this method, we tested a wide range of percentages from 60% to 100%, and found that when the percentage was set to 100%, the best average performance was achieved. Therefore, FSFC with 100% of the MBs considered as P MBs was chosen for the comparison. In this case, E_{tm} is $\sqrt{2}$ and delta QP between two adjacent temporal levels is 3.
- FQC: The delta QP value between adjacent two temporal levels *t+1* and *t* was fixed, denoted as ΔQP_{t+1}. A wide range of ΔQP_{t+1} (t = *0, 1, ..., T-2*) values were tested to achieve a good performance for this method. Finally, we found that when ΔQP_1 was set to 4 and ΔQP_{t+1} (t = *1, 2, ..., T-2*) was set to 1, the best RD performance was achieved. Therefore, FQC with ΔQP_1 equal to 4 and ΔQP_t ($t > 1$) equal to 1 was chosen for the comparison in this paper. Note that the QP setting method for hierarchical B coding in [7] uses the same delta QP values as in this method.

TABLE I
Performance comparison between CAC and other methods

Sequence	CAC vs IPPP		CAC vs FQC		CAC vs FSFC	
	PSNR (dB)	*Bit-rate*	*PSNR (dB)*	*Bit-rate*	*PSNR (dB)*	*Bit-rate*
Container	*1.39*	*-7.2%*	*0.14*	*-2.7%*	*0.28*	*-5.1%*
Foreman	*1.31*	*-19.8%*	*0.16*	*-2.9%*	*0.26*	*-4.1%*
Irene	*1.19*	*-20.0%*	*0.14*	*-2.5%*	*0.21*	*-3.7%*
Mobile	*3.16*	*-43.0%*	*0.23*	*-5.2%*	*0.32*	*-7.3%*
News	*1.11*	*-22.6%*	*0.38*	*-5.7%*	*0.58*	*-8.6%*
Paris	*2.20*	*-28.6%*	*0.48*	*-7.5%*	*0.71*	*-10.7%*
Silent	*2.14*	*-29.7%*	*0.53*	*-8.6%*	*0.73*	*-11.5%*
Tempete	*2.13*	*-34.2%*	*0.14*	*-3.2%*	*0.19*	*-4.1%*
Average	*1.84*	*-25.6%*	*0.28*	*-4.8%*	*0.41*	*-6.9%*

The average RD performances of the four methods are listed in Table I. The results were generated using the Bjontegaard measurement [11], which is based on the bit-rates and average luma PSNR values of the four test points corresponding to four different input QP values.

From Table I, it can be observed that hierarchical P coding outperforms traditional IPPP coding significantly. Compared with traditional IPPP coding, the CAC method brings 25.6% bit-rate saving on average for all tested sequences. Although PSNR fluctuations inside a GOP exist in CAC method, no annoying subjective pumping artifacts occur.

Compared with other QP cascading methods, i.e., FQC and FSFC, the proposed CAC method has noticeable better performance. Up to 8.6% and 11.5% bit-rate savings can be

achieved when compared with FQC and FSFC, respectively. On average, the PSNR gains are about 0.3 dB (compared with FQC) and 0.4 dB (compared with FSFC) respectively.

The tested sequences can be classified according to the motion activity into three types, namely sequences with low motion and almost still background, sequences with low to medium motion, and sequences with high motion.

For the sequences with low motion and almost still background, let us take "*Silent*" as an example. The RD curves are shown in Fig. 2. The average PSNR gain compared with FSFC and FQC is about 0.6 dB.

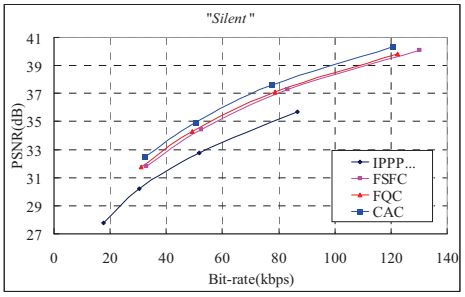


Fig. 2. RD curves for "*Silent*".

The RD curves for "*News*", belonging to sequences with low to medium motion, are shown in Fig. 3. Compared with FSFC and FQC, an average of about 0.5 dB PSNR gain was achieved.

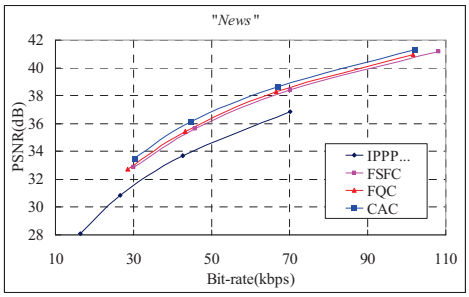


Fig. 3. RD curves for "*News*".

For the sequences with high motion, the average PSNR gain is a little smaller. However, even in this case, the proposed method CAC outperforms FSFC and FQC by about 0.2dB PSNR gain.

It can be concluded that the proposed CAC method provides greater compression gains for sequences with lower motion activity. The reason for this is that the percentage of I MBs increases for sequences with higher motion activity.

Additionally, it is worth mentioning that constant QP value for all the pictures with the same hierarchical P structure provides even less efficiency than IPPP, which is about 0.3 dB PSNR loss.

5. CONCLUSIONS

H.264/AVC Baseline profile lacks the support for bi-predictive (B) slices but supports hierarchical inter prediction structures, which can be used to improve compression efficiency at the cost of increased latency. In this paper, we presented a content-adaptive method for adjusting the Quantization Parameter (QP) for hierarchical inter (P) picture structures. We compared the presented method against traditional non-hierarchical P picture coding (IPPP) as well as two hierarchical P picture coding schemes that selected the QP for a picture based on its temporal level similarly to the methods presented in the literature for hierarchical B picture coding. Simulation results showed that significant gains, more than 25 % bit-rate saving, were achieved over the traditional IPPP coding structure. Compared with the fixed QP setting methods, the proposed method provided a noticeable coding efficiency improvement, about 0.3 dB in luma PSNR on average, which is equivalent to about 5 % bit-rate saving.

6. REFERENCES

[1] T. Wiegand, G. J. Sullivan, G. Bjontegarrd and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Tech.*, Volume 13, pp. 560 - 576, July 2003.

[2] ITU-T Recommendation H.264, "Advanced video coding for generic audiovisual services," Nov. 2007.

[3] M. Flierl and B. Girod, "Generalized B pictures and the draft H.264/AVC video compression standard," IEEE *Trans. Circuits Syst. Video Tech.*, Volume 13, pp. 587 - 597, July 2003.

[4] 3GPP TS 26.234, "Transparent end-to-end packet-switched streaming service (PSS) Protocols and codecs".

[5] 3GPP TS 26.346, "Multimedia broadcast/multicast service (MBMS); Protocols and codecs".

[6] 3GPP TS 26.114, "IP multimedia subsystem (IMS); Multimedia telephony; Media handling and interaction".

[7] H. Schwarz, D. Marpe, and T. Wiegand, "Analysis of hierarchical B pictures and MCTF," *Proc. of IEEE Int. Conference on Multimedia and Expo (ICME)*, pp.1929 - 1932, July 2006.

[8] D. Prannatha, M. Kim, S. Hahm, B. Kim, K. Lee and K. Park, "Dependent Quantization for Scalable Video Coding," *The 9th Int. Conference on Advanced Communication Technology*, pp. 222 - 227, Feb. 2007.

[9] L. Luo, F. Wu, S. Li, and Z. Zhang, "Advanced lifting-based motion threading (MTh) technique for 3D wavelet video coding," *Proc of SPIE VCIP2003*, Volume 5150, pp. 707 - 718, July 2003.

[10] J. Vieron, M. Wien, and H. Schwarz, "JSVM 8_6 software," Joint Video Team Doc. *JVT-U202*, Hangzhou, China, Oct. 2006.

[11] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," *VCEG-M33*, Mar. 2001.

[P2] Y. Gao, Y. Wu, and Y. Chen, “H.264/Advanced Video Coding (AVC) Backward-Compatible Bit-Depth Scalable Coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 4, pp. 500–510, April 2009.

H.264/Advanced Video Coding (AVC) Backward-Compatible Bit-Depth Scalable Coding

Yongying Gao, Yuwen Wu, and Ying Chen, *Member, IEEE*

***Abstract*— This paper presents a bit-depth scalable coding solution that is compatible with the scalable extension of H.264/Advanced Video Coding (AVC), also referred to as scalable video coding (SVC). The proposed solution is capable of providing an 8-bit AVC main profile or high-profile base layer-coded bitstream multiplexed with a higher bit-depth-enhancement layer coded bitstream generated through macroblock level inter-layer bit-depth prediction. New decoding processes for inter-layer prediction are introduced to enable bit-depth scalability. Compatibility with other types of scalability in the SVC standard—temporal, spatial, and SNR scalability—is ensured. It also supports the single-loop decoding required in the SVC specification. Furthermore, it supports adaptive inter-layer prediction to determine whether or not the inter-layer bit-depth prediction shall be invoked. This solution is implemented on the basis of the SVC reference software Joint Scalable Video Model version 8.12. Experimental results are presented on 8-bit to 10-bit bit-depth scalability and also combined bit-depth and spatial scalability.**

***Index Terms*— Bit-depth scalable coding, FREXt, H.264/advanced video coding (AVC), inter-layer bit-depth prediction inverse tone mapping, scalable video coding (SVC), video coding.**

I. Introduction

IN RECENT years, digital images/videos with bit-depths higher than eight are desirable in many fields, such as medical image processing, digital workflows in production and postproduction, and home theater-related applications. In the latest international video coding standard H.264/AVC [1], high bit-depth video coding is standardized in its fidelity range extensions (FREXt) [2]–[4], supporting bit-depths up to 14 bits. On the other hand, the Motion JPEG2000 (Part 3) supports high bit-depth coding up to 32 bits per component [5].

Bit-depth scalability is potentially useful considering the fact that for a long time into the future, conventional 8-bit and high-bit digital imaging systems will simultaneously exist in the market. There are several ways to provide multiple representations of different bit-depths for the same visual content, e.g., an 8-bit and a 10-bit video sequence. One solution is to provide only a 10-bit coded bitstream and to enable tone-mapping methods to give an 8-bit representation for 8-bit display devices [6]. Another solution is to give a simulcast bitstream that contains an 8-bit coded bitstream and a 10-bit coded bitstream. A powerful decoder with support of H.264/AVC High 10 Profile can decode and output a 10-bit video sequence, while an 8-bit decoder can decode and output an 8-bit video. The first solution is not compliant with an H.264/AVC 8-bit decoder. The second solution is compliant with all the current standards at the expense of more overheads. A scalable solution can be a good tradeoff between overhead reduction and backward standard compatibility. In [7]–[12], support of bit-depth scalability in the scalable extension of H.264/AVC (SVC) has been proposed.

The topic of bit-depth scalability has been touched upon in various areas of digital image/video processing. In high dynamic range (HDR) image/video processing [13], [14], how to store or compress the HDR image/video data has been studied; [15] proposed a method to encode the full gamut of still HDR image data; in [16] a subband encoding of high dynamic range imagery is presented; another related work approached interframe encoding for HDR videos, which is embedded in the well-established MPEG-4 video compression standard [17]. The well-known scalable image compression techniques, e.g., embedded zerotree wavelet (EZW) [18], set partitioning in hierarchical tree (SPIHT) [19], and embedded block coding with optimized truncation (EBCOT) [20], can also be treated as approaches for bit-depth scalability because of the nature of bit-plane coding. The idea of bit-plane coding has been integrated into JPEG-2000 standard [21]. JVT-L015 [22] proposes an optional quality-scalable coding method: a fully encoded bitstream can be used for professional video applications, while a subset of it can be decoded by legacy devices, without decoding the enhancement layer bitstream. The signal-to-noise ratio (SNR) methods for scalability in MPEG-2 Video [23] and MPEG-4 Visual [24] also provide a form of bit-depth scalability. The SNR profile defined in MPEG-2 Video adds support for upper layers of DCT coefficient refinement. This SNR method only allows a few selected bitrates to be supported in a scalable bitstream. Different from the SNR scalable coding in MPEG-2 Video, the MPEG-4 FGS (fine-grain scalability) allows arbitrary truncation of the bitstream. Both SNR scalable coding schemes have the following in common: 1) The scalability from the base layer to the quality-refinement layer(s) is done in the transform domain; and 2) the quality refinement layers correspond to the same original input video signal as the base layer does.

Manuscript received March 5, 2007; revised July 31, 2008. First version published March 4, 2009; current version published May 20, 2009. This paper was recommanded by Associate Editor H. Sun.

Y. Gao was with Thomson Corporate Research Beijing, Beijing 100085, China. She is now with the Advanced Technology Division of MediaTeck (Beijing) Corporation, Beijing 100190, China (e-mail: yongying.gao@mediatek.com).

Y. Wu is with Thomson Corporate Research Beijing, Beijing 100085, China (e-mail: yu-wen.wu@thomson.net).

Y. Chen was with Thomson Corporate Research Beijing, Beijing 100085, China. He is now with the Department of Signal Processing, Tampere University of Technology, Tampere 33720, Finland (e-mail: ying.chen@tut.fi).

Digital Object Identifier 10.1109/TCSVT.2009.2014018

The SVC standardization project, which was originally started by the ISO/IEC Moving Picture Experts Group (MPEG), was jointly finalized by the MPEG and the ITU-T Video Coding Experts Group (VCEG) [25]. The SVC system was designed on the basis of a layered approach with a fully H.264/AVC-compliant base layer [26]. Over this base layer, enhancement layers may be added to provide enhanced spatial resolution, temporal resolution, or quality. Spatial scalability is achieved by upsampling the texture and motion vectors of the macroblocks in a base layer. Temporal scalability can be originally supported by H.264/AVC when complicated GOP structures, e.g., hierarchical B pictures [27], are utilized. SNR scalability, also known as quality scalability, is supported by medium-grain scalability (MGS), in which the inter-layer prediction is directly performed in the transform domain by requantizing the residual texture signal in the enhancement layer with a smaller quantization step size relative to that used for the preceding MGS layer. An alternative solution for SNR scalability utilizing FGS tool has been developed, which encodes successive refinements of the transform coefficients by repeatedly decreasing the quantization step size and applying a modified entropy coding process that can be considered as a kind of sub-bit plane coding [28].

We present an H.264/AVC backward-compatible bit-depth scalable coding solution, where the low bit-depth (usually 8-bit) and the high bit-depth (e.g., 10-, 12-, or 14-bit) sequences are encoded as the base layer and enhancement layer(s), respectively. The inter-layer prediction between the base layer and the enhancement layer is done at the macroblock level to take advantage of the redundancy between the low bit-depth and high bit-depth representations of the same visual contents. Compatibility with other types of scalability in the current SVC standard is ensured. The whole solution is implemented on the basis of the SVC reference software Joint Scalable Video Model version 8.12. Experimental results show the advantages of the proposed solution compared to simulcast and other possible solutions, which can also provide multiple representations of the same visual content in terms of coding efficiency.

II. Framework of Bit-Depth Scalable Coding

Without the loss of generality, we assume that there are two layers of bit-depth scalability: one is the base-layer 8-bit video sequence and the other the enhancement-layer 10-bit sequence. The two layers are assumed to have the same chroma sampling format (we focus only on chroma sampling format 4:2:0). More than two bit-depth layers and other values of bit-depth are also supported in the proposed coding scheme. Furthermore, the combination of different bit-depths and chroma sampling formats can also be supported within the proposed framework as shown in Fig. 1.

The input to the scalable encoder is an 8-bit 4:2:0 video sequence and a 10-bit 4:2:0 video sequence, while the output is an scalable bitstream with multiplexed base-layer NAL units and enhancement-layer NAL units. How the 8-bit and 10-bit video sequences are generated is an application-dependent issue.

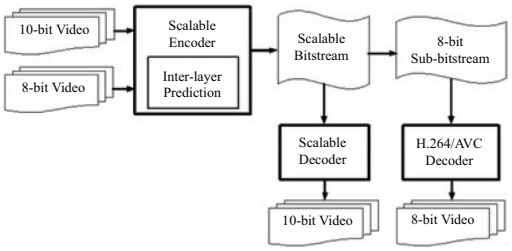


Fig. 1. Diagram of bit-depth scalable coding.

The 8-bit sub-bitstream shall be conforming to the SVC decoder or H.264/AVC decoder, which, e.g., supports of Main Profile, or up to High Profile. The scalable decoder can generate either an 8-bit video sequence by decoding only the base-layer bitstream or generates a 10-bit video sequence by decoding the whole scalable bitstream. By providing multiple versions of different bit-depths for the same visual content to different clients, device adaptation is achieved.

III. Inter-layer Bit-Depth Prediction

The macroblock level inter-layer bit-depth prediction is based on the current structure of spatial scalability/MGS in SVC with modifications for the decoding processes. Predicting the high-bit-depth signal from the low-bit-depth signal is difficult in inter-layer prediction. First, the relationship between the two input sequences can be complicated and sometimes unknown to the encoder, i.e., different gamma-correction procedures, different color corrections, or different spatial filtering processes. Second, the reconstructed low bit-depth signal rather than the original one is used in the inter-layer prediction. Since the reconstructed low bit-depth signal has different statistical characteristics than the original low bit-depth signal after the lossy step of quantization, the resulting residual signal can be difficult to encode. An extreme example is when the input high-bit sequence contains a large amount of noise in the least significant bits (LSBs) and the low bit-depth signal is created from the high-bit signal by linear scaling, and the residual signal has high entropy and does not benefit much from transformation and entropy coding. In Section IV, we provide the experimental results (R-D curves in Fig. 7, 9–11) as well as more detailed discussion regarding this issue.

The proposed inter-layer bit-depth prediction is performed in the spatial domain. In practice, this prediction can be done adaptively: for an enhancement layer macroblock, whether or not the inter-layer bit-depth prediction is used is determined by rate-distortion optimization (RDO). If it is not used, traditional AVC tools will be used to encode this macroblock. For the convenience of statements, we first give the following notations that will be used in the remaining part of this section:

BL_{org}: base layer original input macroblock;
BL_{rec}: base layer reconstructed macroblock;
BL_{res}: base layer (reconstructed) residual macroblock;
EL_{org}: enhancement layer original input macroblock;
EL_{rec}: enhancement layer reconstructed macroblock;
EL_{res}: enhancement layer residual in intra-coding;
EL'_{res}: redefined enhancement layer residual macroblock in inter-coding.

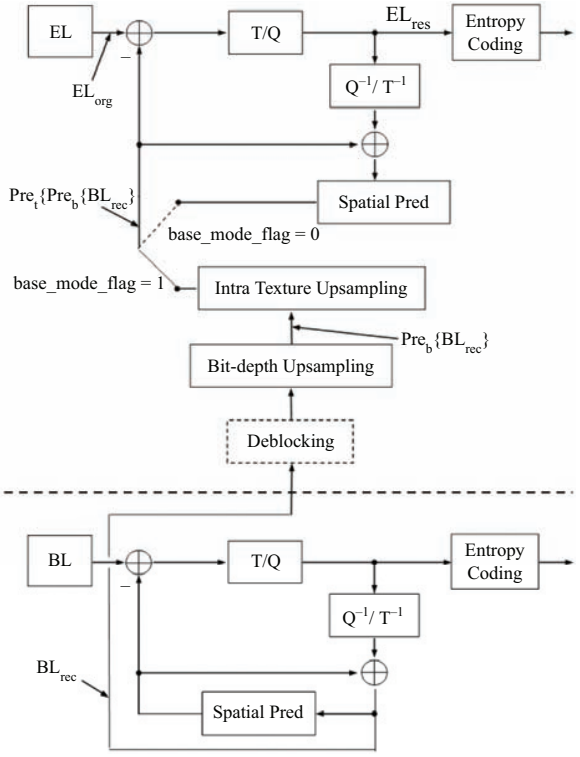


Fig. 2. Diagram of the intra-coding in bit-depth scalable encoder.

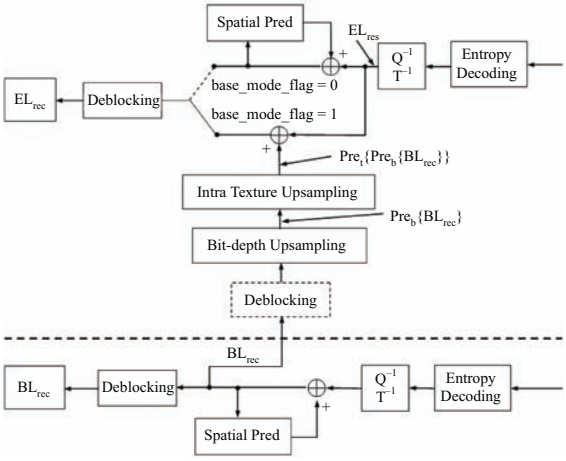


Fig. 3. Diagram of the intra-coding in bit-depth scalable decoder.

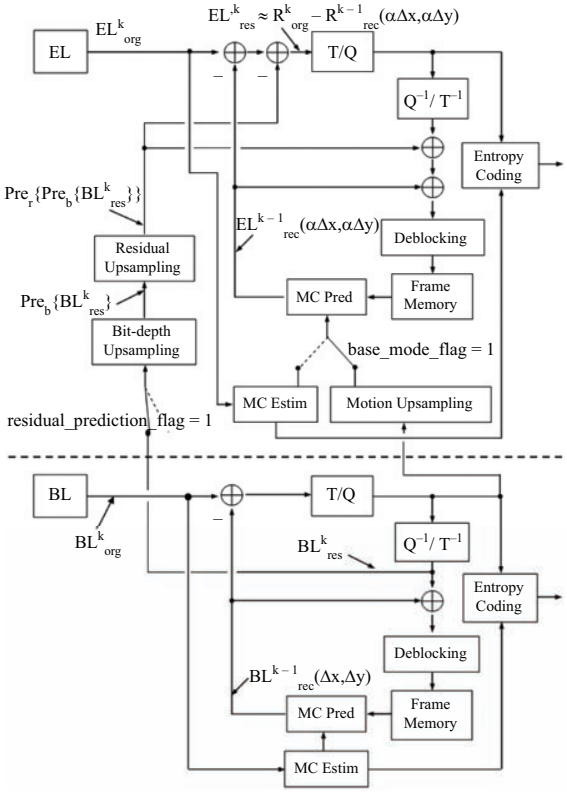


Fig. 4. Diagram of the inter-coding in bit-depth scalable encoder.

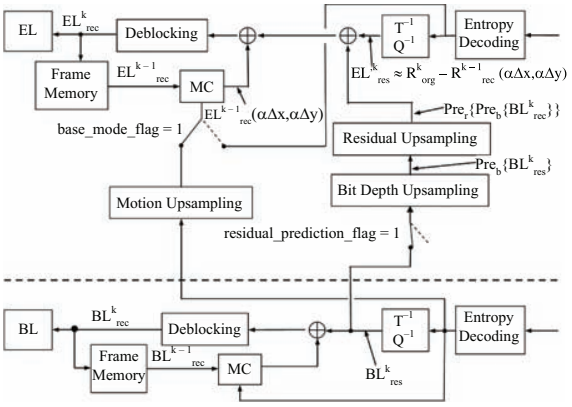


Fig. 5. Diagram of the inter-coding in bit-depth scalable decoder.

A. Inter-layer Intra Bit-Depth Prediction

Figs. 2 and 3 illustrate the bit-depth scalable encoder and decoder, respectively, in intra-coding. The dotted-lined box of deblocking prior to bit-depth upsampling will be enabled only when there is a change in spatial resolution from the base layer to the enhancement layer. The mathematical expression of inter-layer intra bit-depth prediction is shown in (1)

$$EL_{res} = EL_{org} - \Pr e_t\{\Pr e_b\{BL_{rec}\}\} \tag{1}$$

where $\Pr e_b\{\cdot\}$ represents bit-depth upsampling operator while $\Pr e_t\{\cdot\}$ represents intra texture spatial resolution upsampling operator. The symbol base_mode_flag controls whether or not the inter-layer prediction is applied. Its definition can be found in Sections G.7.3.6 and G.7.4.6 in [25].

B. Inter-layer Inter Bit-Depth Prediction

For P and B macroblocks, the inter-layer bit-depth prediction is shown in Figs. 4 and 5 for the encoder and the decoder, respectively. Without losing generality, we assume that the current macroblock has only one reference. The superscripts k and k – 1 represent the current macroblock and the reference macroblock, respectively. The symbol residual_prediction_flag controls whether or not the inter-layer residual prediction is

applied. Please be referred to Sections G.7.3.6 and G.7.4.6 in [25] for a detailed description.

The redefined enhancement layer residual EL'_{res} is different from that defined in SVC spatial scalability/MGS. Assume that both the bit-depth prediction operator $\Pr e_b\{\cdot\}$ and the residual (inter texture) upsampling operator $\Pr e_r\{\cdot\}$ have the attribute of additivity[1] (in practice, the residual upsampling operation that is employed in current SVC spatial scalability/MGS has the characteristics of additivity). We show below that encoding the proposed redefined enhancement layer residual is approximately equivalent to the inter-encode, the so-called inter-layer residual, as defined in the following:

$$R = EL - \Pr e_r\{\Pr e_b\{BL\}\}. \tag{2}$$

According to Fig. 4, the redefined enhancement layer residual EL'^{k}_{res} is

$$\begin{aligned} EL'^{k}_{res} &= EL^{k}_{org} - EL^{k-1}_{rec}(\alpha\Delta x, \alpha\Delta y) - \Pr e_r\{\Pr e_b\{BL^{k}_{res}\}\} \\ &\approx EL^{k}_{org} - EL^{k-1}_{rec}(\alpha\Delta x, \alpha\Delta y) \\ &\quad - \Pr e_r\{\Pr e_b\{BL^{k}_{org} - BL^{k-1}_{rec}(\Delta x, \Delta y)\}\} \end{aligned} \tag{3}$$

where $(\Delta x, \Delta y)$ represents the motion vector of the current base layer macroblock, α represents the spatial scaling factor of the enhancement layer, $BL^{k-1}_{rec}(\Delta x, \Delta y)$ represents the motion-compensated version of the deblocked reconstruction of the reference base layer macroblock, and $EL^{k-1}_{rec}(\alpha\Delta x, \alpha\Delta y)$ represents the (possibly upsampled) motion-compensated version of the deblocked reconstruction of the reference enhancement layer macroblock, as shown in Figs. 4 and 5. The approximation sign "$\approx$" comes from the difference between BL^{k}_{res} and $BL^{k}_{org} - BL^{k-1}_{rec}(\Delta x, \Delta y)$ introduced by quantization.

With the assumption that both $\Pr e_b\{\cdot\}$ and $\Pr e_r\{\cdot\}$ have the attribute of additivity, (3) is equivalent to

$$\begin{aligned} EL'^{k}_{res} &\approx EL^{k}_{org} - EL^{k-1}_{rec}(\alpha\Delta x, \alpha\Delta y) - \Pr e_r\{\Pr e_b\{BL^{k}_{org}\}\} \\ &\quad + \Pr e_r\{\Pr e_b\{BL^{k-1}_{rec}(\Delta x, \Delta y)\}\} \\ &= (EL^{k}_{org} - \Pr e_r\{\Pr e_b\{BL^{k}_{org}\}\}) - (EL^{k-1}_{rec}(\alpha\Delta x, \alpha\Delta y)) \\ &\quad - \Pr e_r\{\Pr e_b\{BL^{k-1}_{rec}(\Delta x, \Delta y)\}\}. \end{aligned} \tag{4}$$

We have two variations of (2)

$$R^{k}_{org} = EL^{k}_{org} - \Pr e_r\{\Pr e_b\{BL^{k}_{org}\}\}; \tag{5}$$

$$R^{k-1}_{rec} = EL^{k-1}_{rec} - \Pr e_r\{\Pr e_b\{BL^{k-1}_{rec}\}\}. \tag{6}$$

The former two items in (4) are equivalent to R^{k}_{org} according to (5), while the latter two items in (4) are a motion-compensated version of R^{k-1}_{rec}. Therefore, (4) can be rewritten as

$$EL'^{k}_{res} \approx R^{k}_{org} - R^{k-1}_{rec}(\alpha\Delta x, \alpha\Delta y). \tag{7}$$

If we regard R^{k}_{org} as the original signal, $R^{k-1}_{rec}(\alpha\Delta x, \alpha\Delta y)$ can be treated as its motion-compensated prediction. The possible spatial correlations between the current residual signal R^{k}_{org} and its counterpart in its reference picture R^{k-1}_{rec} are removed. In other words, encoding EL'^{k}_{res} is approximately equivalent (bounded by the quantization error in quantizing BL^{k}_{res}) to inter-encoding the inter-layer residual R.

[1]The definition of additivity can be found in [[29], p. 53, Ch. 1]

C. Bit-Depth Upsampling

As depicted in Figs. 2–5, the macroblock-level inter-layer bit-depth prediction is implemented through bit-depth upsampling. How the bit-depth upsampling works is an application-dependent issue. As a start point, we propose linear scaling of the collocated base layer macroblock to predict the enhancement layer macroblock, as expressed in the following formula:

$$\overline{EL} = BL \times 2^{N-M} \tag{8}$$

where $\overline{EL}$ represents the predicted enhancement-layer macroblock with bit-depth N and BL is the collocated base layer reconstructed macroblock with bit-depth M.

Other possible bit-depth upsampling methods include referring to the same lookup table (LUT) at both encoder and decoder, or nonlinear approximation such as logarithmic and polynomial. The bit-depth upsampling can be done in either a global manner (the same bit-depth upsampling approach applied on the whole picture), or a localized manner (different bit-depth upsampling approaches applied to different areas within the whole picture). To enable various inter-layer bit-depth prediction approaches, new syntax elements and the corresponding semantics and decoding processes are needed.

D. Combined Bit-Depth and Spatial Scalability

The order of bit-depth upsampling and spatial upsampling does impact the coding efficiency in combined bit-depth and spatial scalability. If linear scaling as specified in (8) is utilized, there is no rounding error introduced in bit-depth upsampling. Therefore, bit-depth upsampling needs to be done before spatial upsampling to reduce the propagation of the rounding error caused by filtering in the spatial upsampling process [25]. We present a proof of the above conclusion for the case in which bit-depth upsampling is conducted from 8 bits to 10 bits and the spatial scaling factor equals 2

$$I_{8,rec}(x, y) \rightarrow \hat{I}_{10}(2x, 2y) \tag{9}$$

where $I_{8,rec}(x, y)$ represents the reconstructed base-layer macroblock, and $\hat{I}_{10}(2x, 2y)$ the predicted version of the collocated enhancement-layer macroblock.

In case the bit-depth upsampling is performed first and then the spatial upsampling, the following operations are applied:

$$\hat{I}_{10}(x, y) = I_{8,rec}(x, y) \times 2^2 \tag{10}$$

$$\hat{I}_{10}(2x, 2y) = \lfloor fs(\hat{I}_{10}(x, y)) \rfloor = \lfloor fs(I_{8,rec}(x, y) \times 2^2) \rfloor \tag{11}$$

where $\lfloor\cdot\rfloor$ represents truncation toward zero, and $fs(\cdot)$ represents filtering for spatial upsampling. The total rounding error is

$$\Delta\hat{I}^{A}_{10}(2x, 2y) = fs(I_{8,rec}(x, y) \times 2^2) - \lfloor fs(I_{8,rec}(x, y) \times 2^2) \rfloor. \tag{12}$$

In case the spatial upsampling is done first and then the bit-depth upsampling, the following operations are applied:

$$\hat{I}_8(2x, 2y) = \lfloor fs(I_{8,rec}(x, y)) \rfloor \tag{13}$$

$$\hat{I}_{10}(2x, 2y) = \hat{I}_8(2x, 2y) \times 2^2. \tag{14}$$

The total rounding error is

$$\begin{aligned}\Delta \hat{I}^B_{10}(2x, 2y) &= \{fs(I_{8,rec}(x, y)) - \lfloor fs(I_{8,rec}(x, y)) \rfloor\} \times 2^2 \\ &= fs(I_{8,rec}(x, y)) \times 2^2 - \lfloor fs(I_{8,rec}(x, y)) \rfloor \times 2^2.\end{aligned} \tag{15}$$

Based on the fact $fs(I_{8,rec}(x, y) \times 2^2) = fs(I_{8,rec}(x, y)) \times 2^2$, and comparing (12) and (15) and we conclude that

$$\Delta \hat{I}^A_{10}(2x, 2y) \leq \Delta \hat{I}^B_{10}(2x, 2y). \tag{16}$$

IV. Experimental Results

We implemented the bit-depth scalable codec based on the SVC reference software JSVM_8_12 [30]. We used four sequences for performance evaluation: CapitolRecords; Cornfield; Plane; and Waves. All the test sequences except for "Cornfield" were originally digital-camera-captured contents and were used in the JVT core experiment for bit-depth scalability [31]. The 8-bit sequences of "CapitolRecords" and "Waves" were created by nonlinear tone-mapping operation, while the 8-bit sequence "Plane" was generated by linear tone mapping [32]. The "Cornfield" is a computer-generated content, and a linear scaling was used to generate its 8-bit signal. We followed the test conditions defined in [31], except that the inter-layer prediction parameter InterLayerPred was set to different values in different test cases. All the test sequences are of 4CIF resolution and 60 frames long. The following are important encoder parameters. FrameRate = 50; GOPSize = 16; IntraPeriod = 32; NumberReferenceFrames = 1; SearchMode = 4 (FastSearch); SearchFuncFullPel = 3 (SAD-YUV); SearchFuncSubPel = 2 (HADAMARD); SearchRange = 96; BiPredIter = 4; IterSearchRange = 8; the QP values for the base layer are 12, 17, 22, 27, and 32.

In the first set of experiments, the spatial resolution and the QP values for the enhancement layer are identical to those for the base layer. InterLayerPred is set to one to always apply inter-layer prediction; For "CapitolRecords" and "Waves," we also tested the R-D performance by applying a linear tone mapping to the original 10-bit signal (implemented by truncation of the two LSBs of the 10-bit signal) to create the 8-bit signal. The R-D curves are depicted in Figs. 6–11. We list only the results of the luma component and one of the chroma components, Cr. We use the term "chroma components" instead of "Cr" in later discussion to clarify that the observations/conclusions work for both Cb and Cr. The curve "TMM scalable" stands for the bit-depth scalable coding scheme; the curve "10-bit single layer" represents encoding only the 10-bit signal; the curve "8-bit upscaled" stands for the process where the 8-bit version of the sequence was encoded using the QP value that resulted in a bitrate closest to that of "TMM scalable"; and then the decoded 8-bit signal was upscaled to 10 bits again by using the bit-depth upsampling operation described in Section III-C. In the figures, the horizontal axis represents the coded bitrate measured at kbits/s. The vertical axis represents the average PSNR (peak signal-to-noise ratio) of the reconstructed sequence measured in dB. $PSNR = 10 \log_N \frac{(2^N-1)^2}{MSE}$, where MSE represents the mean squared error of the reconstructed sequence, and N represents the bit-depth of the coded signal. When the coding performance of 10-bit signals is considered, e.g., the curve "10-bit single layer" and "8-bit upscaled," the PSNR value is calculated in the 10-bit domain: $N = 10$.

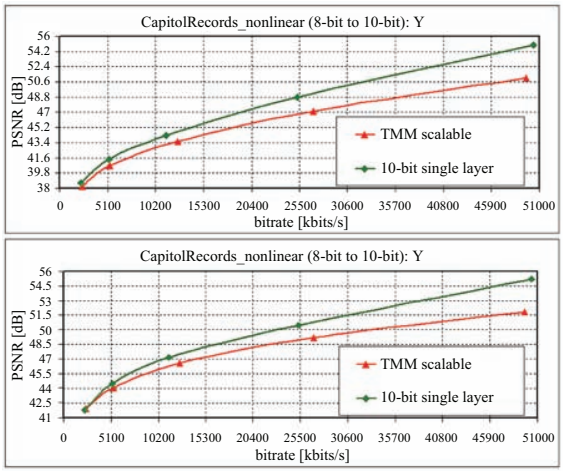


Fig. 6. Bit-depth scalability for "CapitolRecords" with nonlinear tone-mapping operation: $QP_{EL} = QP_{BL}$.

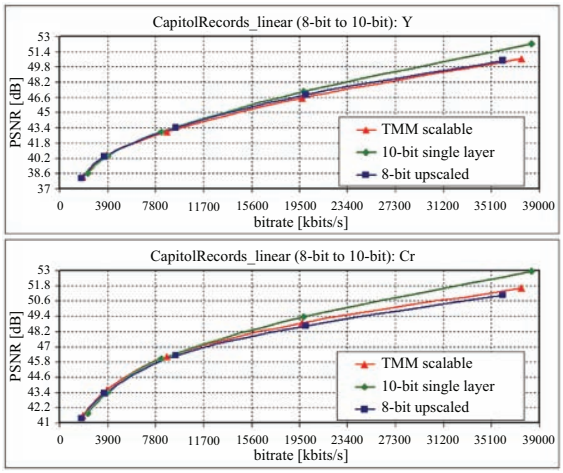


Fig. 7. Bit-depth scalability for "CapitolRecords" with linear tone-mapping operation: $QP_{EL} = QP_{BL}$.

Regarding the luma component in the case of linear tone mapping, the R-D curve "TMM scalable" of "Cornfield" shows the best performance, while that of "CapitolRecords" gives the worst performance, compared with the R-D curve "8-bit upscaled." At low bitrates, the performance of "8-bit upscaled" is slightly better than that of "TMM scalable." As the bitrate goes higher, the "TMM scalable" outperforms the "8-bit upscaled" more and more. We analyze the possible reasons for the above observations. First, when the bit-depth upsampling operation employed in "8-bit upscaled" is an inverse procedure

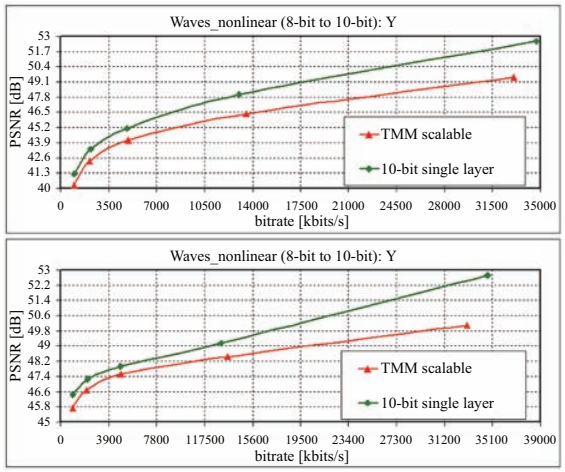


Fig. 8. Bit-depth scalability for "Waves" with nonlinear tone-mapping operation: $QP_{EL} = QP_{BL}$.

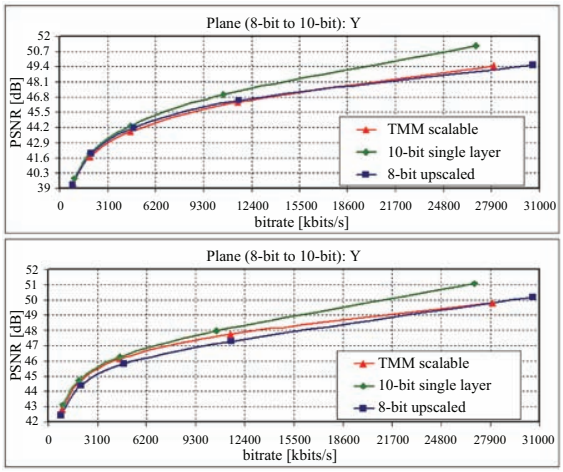


Fig. 10. Bit-depth scalability for "Plane": $QP_{EL} = QP_{BL}$.

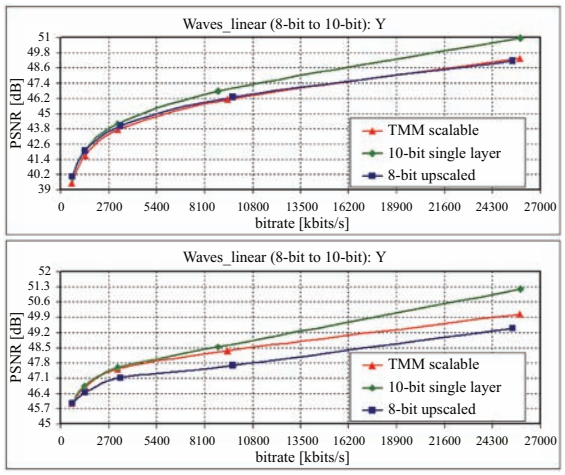


Fig. 9. Bit-depth scalability for "Wave" with linear tone-mapping operation: $QP_{EL} = QP_{BL}$.

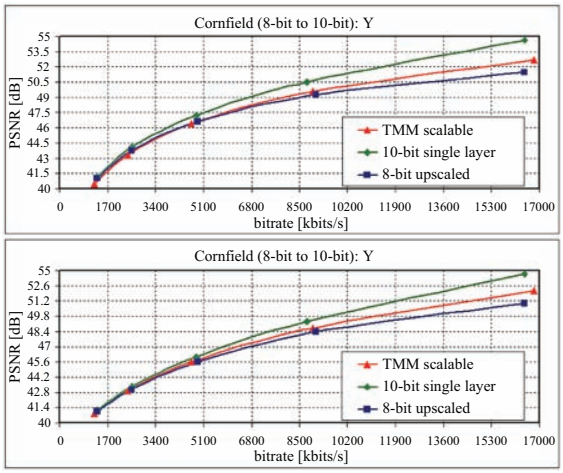


Fig. 11. Bit-depth scalability for "Cornfield": $QP_{EL} = QP_{BL}$.

of the linear tone mapping utilized in creating the 8-bit signal, the upscaled 8-bit reconstruced signal is a good approximation of the 10-bit signal. The two LSBs that are not contained in the 8-bit signal play an important role in determining the performance of encoding the enhancement layer (and therefore the overall performance of the scalable solution), compared with "8-bit upscaled" signal. Since "Cornfield" is a computer-generated graphic content, the entropy of the two LSBs is lower than that for the other three sequences with noise-like LSBs. For "CapitolRecords," the statistical analysis shows that more than 70% of the pixels in the 10-bit signal have a luminance level lower than 4. Accordingly, the collocated pixels in the 8-bit signal have a luminance level of zero. As a result, more than 50% of the enhancement layer macroblocks (in this case, the residual signal after inter-layer prediction) have a luminance level of zero before transformation, within the tested QP range. Such an enhancement-layer macroblock does not contain any additional information compared to the collocated base layer reconstruction macroblock and hence will not improve the reconstruction quality of the 10-bit signal. However, except for the quantized coefficient levels, other syntax elements still need to be encoded. Therefore, the scalable solution underperforms the "8-bit upscaled." Second, on one hand, in the proposed scalable solution, decreasing the quantization step can improve the reconstruction quality towards the original signal; on the other, the "8-bit upscaled" signal is an approximation of the 10-bit signal and its reconstruction quality is bound by the PSNR of the upscaled 8-bit signal (truncated from the 10-bit original signal without encoding/decoding), regardless of the quantization step. It explains the fact that as the bitrate goes higher, the "TMM scalable" outperforms the "8-bit upscaled" more and more. Regarding the chroma components in the linear tone-mapping case, the performance of "TMM scalable" is always better than "8-bit upscaled." This observation can be beneficial in professional applications since, when we are turning from the conventional 8-bit to high-bit-depth, the reconstruction quality of the chrominance components are more important in determining the final visual quality.

TABLE I
BD-PSNR OF THE SCALABLE SOLUTION RELATIVE TO THE 10-BIT SINGLE LAYER CODING, CORRESPONDING TO FIGS. 6–11

	BD-PSNR for Y QP = 12, 17, 22, 27	BD-PSNR for Cr QP = 12, 17, 22, 27	BD-PSNR for Y QP = 12, 17, 22, 32	BD-PSNR for Cr QP = 12, 17, 22, 32
Capitol Records nonlinear	−1.6818	−1.3023	−1.0309	−1.6965
Capitol Records linear	−0.4337	−0.2298	−0.1211	−0.0441
Waves nonlinear	−1.5542	−0.8187	−1.0947	−0.5305
Waves linear	−0.7563	−0.2829	−0.4809	−0.0748
Plane	−0.8013	−0.4026	−0.4157	−0.1549
cornfield	−0.9675	−0.6057	−0.5877	−0.2477

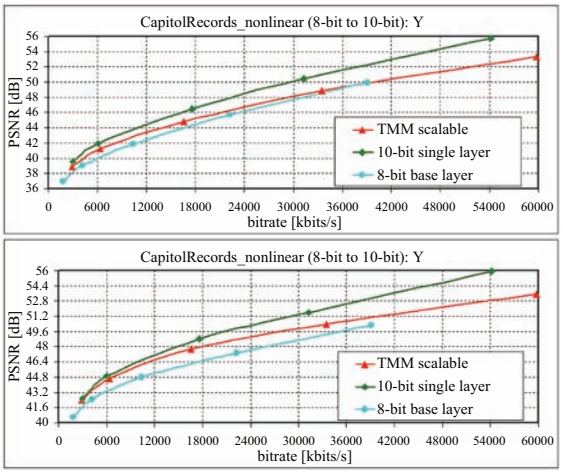


Fig. 12. Bit-depth scalability for "CapitolRecords" with nonlinear tone mapping: bitrate EL = 35%, bitrate BL = 65%.

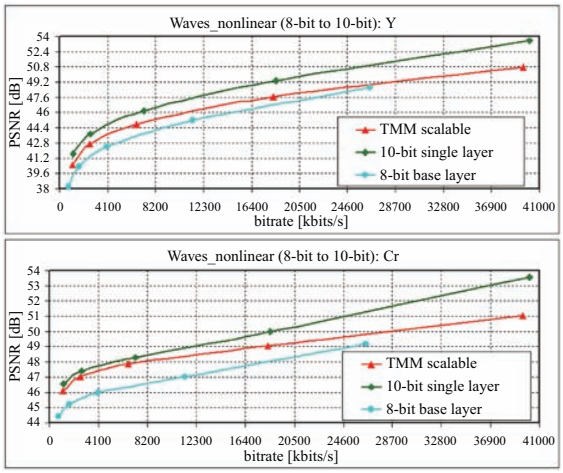


Fig. 13. Bit-depth scalability for "Waves" with nonlinear tone mapping: bitrate EL = 35%, bitrate BL = 65%.

As for the nonlinear tone mapping of "CapitolRecords" and "Waves," it is not surprising that their performance is not as good as that of their counterparts in linear tone mapping since the utilized linear tone-mapping operation is an inverse process of the bit-depth upsampling operation. In general, the performance of nonlinear tone mapping and linear tone mapping heavily depends on how the 8-bit video is generated.

We list in Table I the Bjontegaard PSNR differences (BD-PSNRs) of the proposed scalable solution relative to the 10-bit single layer coding, corresponding to Figs. 6–11.

In the second round of experiments, we used another bitrate allocation strategy (keeping all other test conditions unchanged): The bitrate allocated to the base layer covers approximately 65% of the whole scalable bitstream, while the that allocated to the enhancement layer covers about 35%. Since the QP values for the base layer were already determined, the R-D performance of the base layer is exactly the same as that in the first round. We tried different QP values in coding the enhancement layer and selected the one that results in the closest bitrate allocation to the desired value. The fixed bitrate percentage between the base layer and the enhancement layer is useful in applications where the target bitrate is required, e.g., High Definition DVD authoring. Detailed R-D curves are shown in Figs. 12–15. We also provide the R-D performance of the base layer as "8-bit base layer." The BD-PSNRs of the scalable solution relative to the 10-bit single layer coding are listed in Table II.

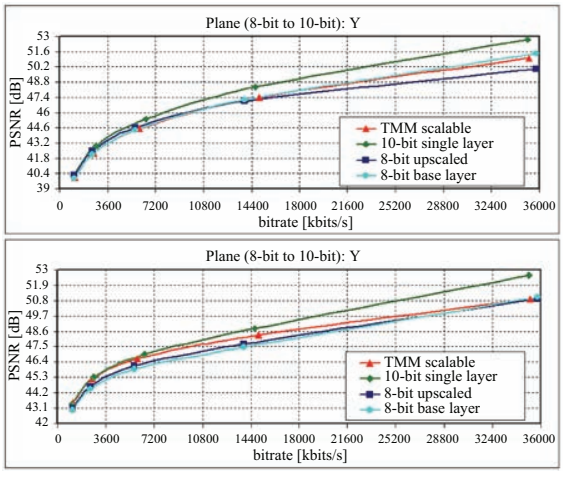


Fig. 14. Bit-depth scalability for "Plane": bitrate EL = 35%, bitrate BL = 65%.

Next, we tested the coding performance of the combined bit-depth and spatial scalability. The input sequence to the enhancement layer is 10-bit, 4CIF resolution, while the input to the base layer is 8-bit, CIF resolution, created by utilizing the downconvert tool in JSVM on the 8-bit, 4CIF sequences. Experimental results are listed in Figs. 16–17. The curve "bit-depth + spatial" stands for bit-depth upsampling first and then spatial upsampling during the inter-layer prediction, while the

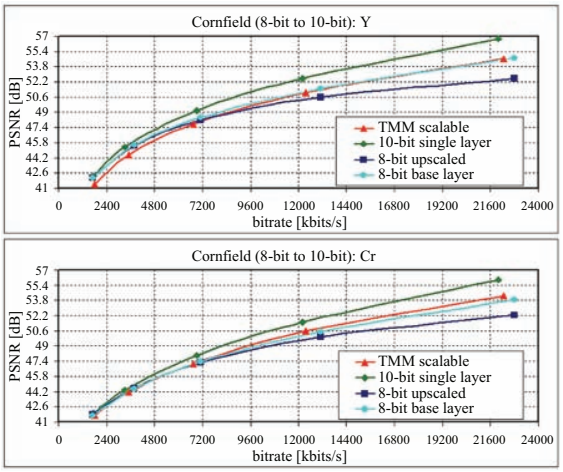


Fig. 15. Bit-depth scalability for "Cornfield": bitrate EL = 35%, bitrate BL = 65%.

TABLE II
BD-PSNR OF THE SCALABLE SOLUTION RELATIVE TO THE 10-BIT SINGLE LAYER CODING, CORRESPONDING TO FIGS. 12–15

	BD-PSNR for Y QP = 12, 17, 22, 27	BD-PSNR for Cr QP = 12, 17, 22, 27	BD-PSNR for Y QP = 12, 17, 22, 32	BD-PSNR for Cr QP = 12, 17, 22, 32
CapitolRecords nonlinear	−1.6415	−1.2148	−1.0874	−0.6608
Waves nonlinear	−1.5065	−0.7551	−1.1865	−0.4183
Plane	−0.9115	−0.4878	−0.6201	−0.1481
Cornfiled	−1.4437	−0.8883	−1.1850	−0.5617

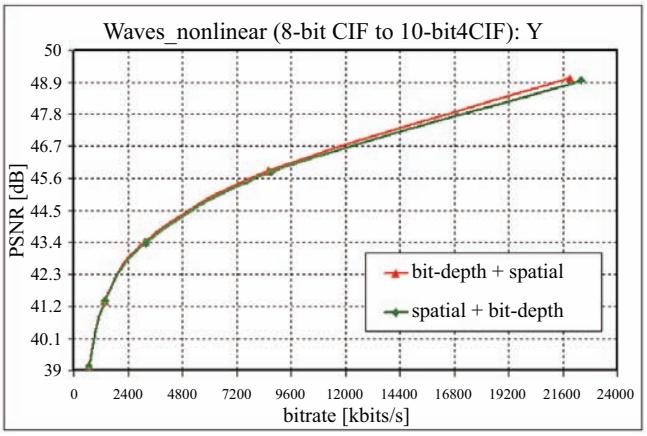


Fig. 16. Comparison between different orders of bit-depth upsampling and spatial upsampling in inter-layer prediction for "Waves."

curve "spatial + bit-depth" stands for spatial upsampling first and then bit-depth upsampling. The corresponding BD-PSNRs are listed in Table III.

For both sequences, bit-depth upsampling first in the inter-layer prediction outperforms spatial upsampling first. This observed fact is consistent with the conclusion we made in Section III-D. Furthermore, the gain of bit-depth upsmpling first over spatial upsampling fist is more significant at high bitrates/high reconstruction qualities. The reason is

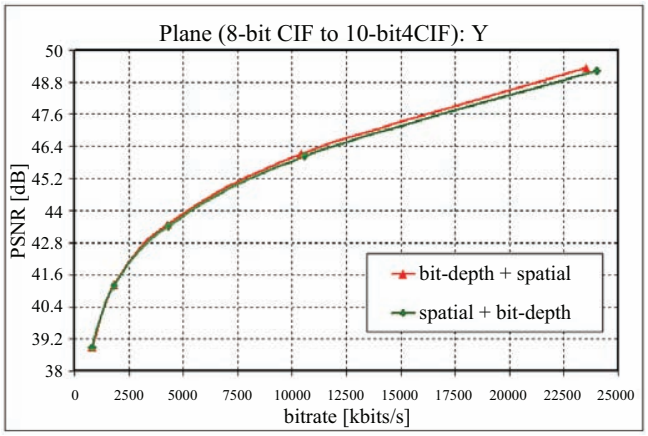


Fig. 17. Comparison between different orders of bit-depth upsampling and spatial upsampling in inter-layer prediction for "Plane."

TABLE III
BD-PSNR OF "BIT-DEPTH + SPATIAL" RELATIVE TO "SPATIAL + BIT-DEPTH" IN INTER-LAYER PREDICTION, CORRESPONDING TO FIGS. 16–17

	BD-PSNR for Y QP = 12, 17, 22, 27	BD-PSNR for Cr QP = 12, 17, 22, 27	BD-PSNR for Y QP = 12, 17, 22, 32	BD-PSNR for Cr QP = 12, 17, 22, 32
Waves nonlinear	0.0908	0.0272	0.0541	0.0131
Plane	0.0964	0.0401	0.0540	0.0196

that at low bitrates, the rounding error introduced by spatial upsampling is trivial compared to the quantization error. Hence, the different prediction orders do not significantly change the R-D performance. At high bitrates, the rounding error caused by spatial upsampling cannot be neglected and the different prediction orders show different R-D performance.

We continue to provide some experimental results applying bit-depth upsampling first in the inter-layer prediction. All the test conditions are consistent with those described in the first paragraph of Section IV, except for InterLayerPred = 2 (adaptive inter-layer prediction) for the enhancement layer. In combined spatial and bit-depth scalability, there are multiple possible prediction modes for a macroblock in the enhancement layer: inter-layer spatial and bit-depth prediction, as well as H.264/AVC intra prediction and inter prediction. In general, by enabling the adaptive inter-layer prediction, the best prediction mode at the macroblock level is selected in terms of R-D performance optimization. We allocated approximate 30% of the bitrate to encode the base layer and 70% to the enhancement layer. We present the R-D curves for the scalable solution, 10-bit single-layer coding and simulcast in Figs. 18–20, and the corresponding BD-PSNRs relative to simulcast in Table IV. The term simulcast means using approximately the same bitrate for the 8-bit and 10-bit simulcast sub-bitstreams as used by the scalable bitstream. Such a definition of simulcast is particularly useful to evaluate the performance of scalable solutions in applications where the bandwidth and/or storage capacity is a critical issue.

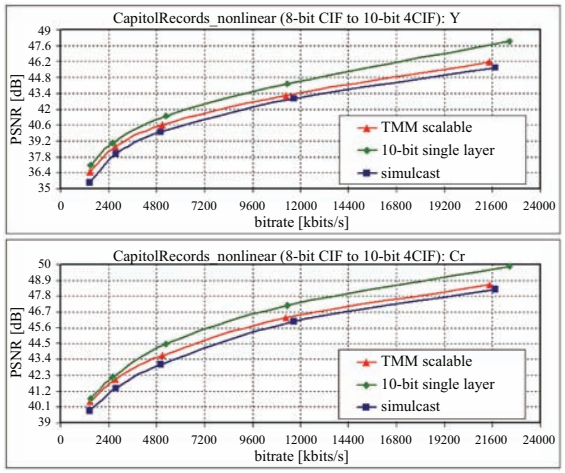


Fig. 18. Combined bit-depth and spatial scalability for "CapitolRecords" with nonlinear tone mapping: bitrate EL = 70%, bitrate BL = 30%.

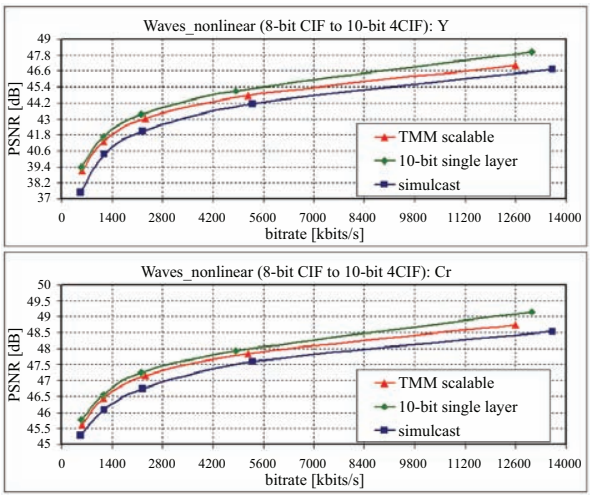


Fig. 19. Combined bit-depth and spatial scalability for "Waves" with nonlinear tone mapping: bitrate EL = 70%, bitrate BL = 30%.

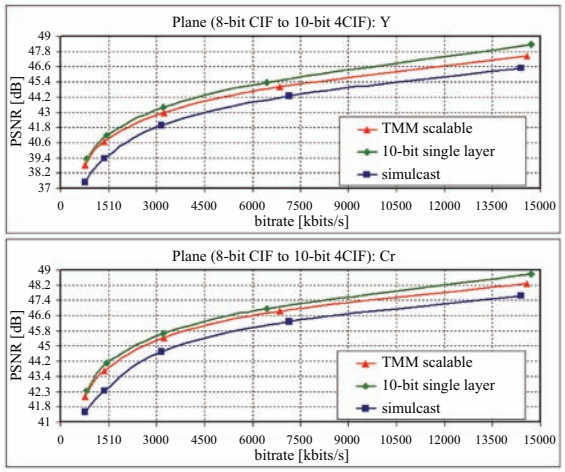


Fig. 20. Combined bit-depth and spatial scalability for "Plane": bitrate EL = 70%, bitrate BL = 30%.

TABLE IV
BD-PSNR OF THE COMBINED BIT-DEPTH AND SPATIAL SCALABILTY RELATIVE TO SIMULCAST, CORRESPONDING TO FIGS. 18–20

	BD-PSNR for Y QP = 12,, 17, 22, 27	BD-PSNR for Cr QP = 12, 17, 22, 27	BD-PSNR for Y QP = 12, 17, 22, 32	BD-PSNR for Cr QP = 12, 17, 22, 32
Capitol Records nonlinear	0.5101	0.4932	0.6049	0.5883
Waves nonlinear	0.7395	0.3201	0.9751	0.3464
Plane	0.9544	0.7003	1.1141	0.8065

The experimental results show the gain of the proposed scalable solution to simulcast. The exact value of the gain is impacted by the visual characteristics of the content, the bitrate allocation strategy in encoding the two layers, and, the characteristics of the utilized tone mapping operation. According to Table IV, "Plane" shows the highest gain over simulcast among the four test sequences, since "Plane" is linearly tone-mapped and benefits from the bit-depth upsampling operation during encoding.

As mentioned before, the combined bit-depth and spatial scalability involves more complicated prediction modes and mode decision strategies than bit-depth scalability. It is well known that by allowing multiple prediction modes, we can improve the R-D performance by using some optimization techniques to find out the mode that best customizes the statistical characteristics of the encoded scene content and/or works most efficiently for the required level of fidelity [33]. In particular, the impact of the following different scenarios was tested. The results are illustrated in Figs. 21–23.

1) "AVC mode": Intralayer prediction only, wherein the enhancement layer is separately encoded using H.264/AVC (Intra and Inter) prediction modes. It is equivalent to simulcast of the different layers utilizing H.264/AVC coding tools. The SVC-specific coding tools are not used in this mode.
2) "ILP mode": Inter-layer prediction only, wherein only inter-layer prediction modes are enabled.
3) "ILP + AVC mode": Combined prediction, wherein H.264/AVC (Intra and Inter) prediction modes and inter-layer prediction modes are both enabled.

The QP values are identical in the three cases: from 7 to 32 with step size 5. The Lagrangian multiplier used in the RDO is: $\lambda = 0.85 \times 2^{(\min(52.0, QP)+6\times(Bitdpeth-8))/3.0-4.0}$, wherein QP represents the QP value of the current enhancement layer slice and *Bitdepth* is the bit-depth of the enhancement layer input.

In the luma component, "AVC mode" achieves better performance than "ILP mode" for "Cornfield"; "ILP mode" achieves better performance than "AVC mode" for "Plane"; "AVC mode" and "ILP mode" have very close performance for "Waves." In chroma components, "AVC mode" performs better than "ILP mode" for "Cornfield"; "ILP mode" performs better than "AVC mode" for "Waves" and "Plane." In general, "ILP mode" performs better in chroma components than in luma component, compared with "AVC mode." In

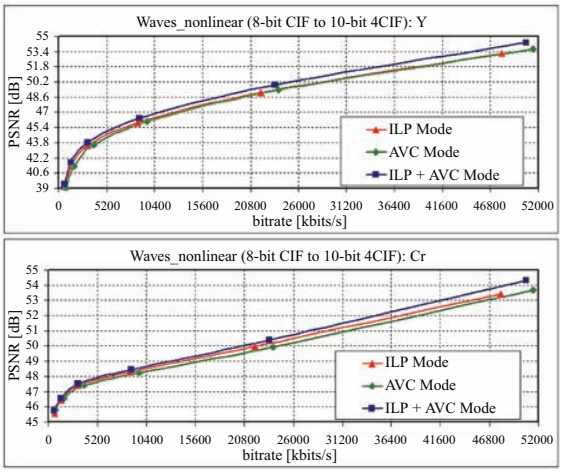


Fig. 21. Different prediction modes in combined bit-depth and spatial scalability for "Waves" with nonlinear tone mapping.

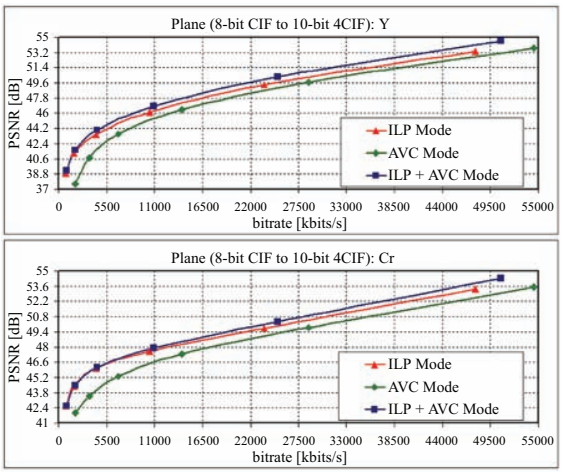


Fig. 22. Different prediction modes in combined bit-depth and spatial scalability for "Plane."

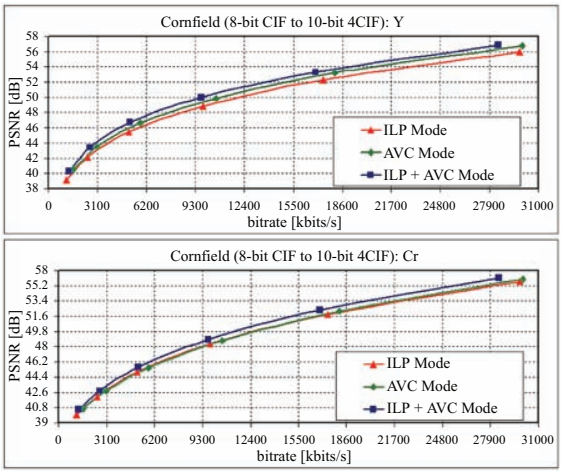


Fig. 23. Different prediction modes in combined bit-depth and spatial scalability for "Cornfield."

luma and chroma components, "AVC + ILP mode" shows superior performance than "AVC mode" and "ILP mode." These observations are consistent with the statement we made in the paragraph preceding Fig. 18 that by enabling the adaptive inter-layer prediction, the best prediction mode at the macroblock level is selected in terms of R-D performance optimization.

V. Conclusions and Future Work

Bit-depth scalable coding is capable of maintaining backward comparability to AVC 8-bit decoders as well as providing enhanced visual content with higher bit-depth, if more powerful decoders are available, and saving bandwidth for video delivery. This paper proposed an H.264/AVC backward-compatible bit-depth scalable coding solution based on macroblock-level inter-layer bit-depth prediction.

The proposed scheme is compatible with other types of scalability supported in current SVC standard. It also supports single-loop decoding, as required in SVC. The presented spatial-domain bit-depth upsampling for inter-layer intra and inter prediction significantly improves the coding efficiency according to the experimental results. In addition, it supports adaptive inter-layer prediction to determine whether or not the inter-layer bit-depth prediction shall be invoked. It is useful in cases where the relationship between the two input sequences is complicated, e.g., when they are related by nonlinear tone mapping or when they have different spatial resolutions. The experimental results reported show that adaptive inter-layer prediction performs better than when using only H.264/AVC modes or only inter-layer prediction modes.

The coding efficiency of the presented bit-depth scalable coding can be further improved by incorporating advanced inter-layer bit-depth prediction algorithms other than linear scaling. Inter-layer prediction techniques that require the availability of the base-layer reconstruction can be applied in our framework to encode these enhancement layer macroblocks of which the collocated base layer macroblocks are intra-coded. As for the case in which the collocated base layer macroblock is inter-coded, more advanced inter-layer prediction techniques are also an important aspect in the proposed bit-depth scalable coding and worth further study.

References

[1] *Advanced video coding for generic audiovisual services*, ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG4-AVC) Standards, v1, May 2003; v2, Jan. 2004; v3 (with FREext), Sep. 2004; v4, Jul. 2005.

[2] G. J. Sullivan, P. Topiwala, and A. Luthra, "The H.264/AVC advanced video coding standard: Overview and introduction to the fidelity extensions," in *Proc. SPIE*, Aug. 2004, vol. 5558, no. 2, pp. 454-474.

[3] D. Marpe, T. Wiegand, and S. Gordon, "H.264/MPEG4-AVC fidelity range extensions: tools, profiles, performance, and application areas," in *Proc. IEEE Int. Conf. Image Process*, Geneva, Italy, Sep. 11–14, 2005, vol. 1, pp. 593–596.

[4] Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, "Joint draft 6 of 'New profiles for professional applications' amendment to ITU-T Rec. H.264 & ISO/IEC 14496-10 (Amendment 2 to 2005 ed.)," Doc. JVT-V204, Marrakech, Morocco, Jan. 2007.

[5] *Motion JPEG 2000 derived from International Standardization Organization base media file format*, ISO/IEC Standard 15444-3:2002/Amd 2:2003, 2003.

[6] A. Segall, L. Kerofsky, and S. Lei, "Tone mapping SEI message," Joint Video Team, Doc. JVT-U049, Hangzhou, China, Oct. 2006.

[7] Y. Gao and Y. Wu, "Applications and requirement for color bit-depth scalability," Joint Video Team, Doc. JVT-U049, Hangzhou, China, Oct. 2006.
[8] Y. Wu, Y. Gao, and Y. Chen, "Bit-depth scalable coding," in *Proc. IEEE Int. Conf. Multimedia and Expo*, Jul. 2007, pp. 1139–1142.
[9] Y. Wu, Y. Gao, and Y. Chen, "Bit-depth scalable coding based on macroblock level inter-layer prediction," in *Proc. IEEE Int. Symp. Circuits and Sys.*, May 2008, pp. 3442–3445.
[10] M. Winken, D. Marpe, H. Schwarz, and T. Wiegand, "Bit-depth scalable video coding," in *Proc. IEEE Int. Conf. Image Process.*, vol. 1, Sep. 2007, pp. 5–8.
[11] A. Segall, "Scalable coding of high dynamic range video," in *Proc. IEEE Int. Conf. Image Process.* vol. 1, Sep. 2007, pp. 1–4.
[12] S. Liu, W.–S. Kim, and A. Vetro, "Bit-depth scalable coding for high dynamic range video," in *Proc. SPIE Conf. Visual Commun. and Image Process.*, Cambridge, Jan. 2008, pp. 1–14.
[13] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda, "Photographic tone reproduction for digital images," *ACM Trans. Graph.,* vol. 21, no. 3, pp. 267–276, 2002.
[14] H. Seetzen, *et al*., "High dynamic range display system," *ACM Trans. Graph. (SIGGRAPH2004)*, vol. 23, no. 3, 2004, pp. 760–768.
[15] G. Ward, "The LogLuv encoding for full gamut, high dynamic range images," *J. Graph. Tools*, vol. 3, no. 1, pp. 15–31, 1998.
[16] G. Ward and M. Simmons, "Subband encoding of high dynamic range imagery," in *Proc. ACM Int. Conf. Scenes*, 2004, vol. 73, pp. 83–90.
[17] P. Mantiuk, *et al*., "Perception-motivated high dynamic range video encoding," in *Proc. of SIGGRAPH 2004 (Spl. issue ACM Trans. Graph.)*.
[18] J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3445–3462, Dec. 1993.
[19] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical tree," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 243–250, Mar. 1996.
[20] D. Daubman, "High performance scalable image compression with EBCOT," *IEEE Trans. Image Process.*, vol. 9, no. 7, pp. 1158–1170, Jul. 2000.
[21] M. W. Marcellin, M. J. Gormish, A. Bilgin, and M. P. Boliek, "An overview of JPEG-2000," in *Proc. Conf. Data Compression*, 2000, pp. 523–541.
[22] S. Sun, "Quality scalability for FRExt," Joint Video Team, Doc. JVT-L015, Redmond, Jul. 2004.
[23] Generic coding of moving pictures and associated audio information–part 2: video, Standard ITU-T Rec. H.262 and ISO/IEC 13818-2 (MPEG-2 Video), ITU-T and ISO/IEC JTC 1, Nov. 1994.
[24] Coding of audio-visual objects–part: 2: visual, ISO/IEC Std. 14492-2 (MPEG-4 Visual), ISO/IEC JTC 1, Version 1: Apr. 1999; Version 2: Feb. 2000; Version 3: May 2004.
[25] Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, "Joint draft ITU-T Rec. H.264 | ISO/IEC 14496-10/Amd.3 scalable video coding," *Doc. JVT-X201-M*, Geneva, Switzerland, Jul. 2007.
[26] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC std.," *IEEE Trans. Circuits and Sys. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.
[27] H. Schwarz, D. Marpe, and T. Wiegand, "Analysis of hierarchical Bc pictures and MCTF," in *Proc. IEEE Int. Conf. Multimedia and Expo*, Jul. 2006, pp. 1929–1932.
[28] J. Reichel, H. Schwarz, and M. Wien, "Joint scalable video model JSVM-11," Joint Video Team, Doc. JVT-X202, Geneva, Switzerland, Jul. 2007.
[29] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals and systems*, 2nd ed., Englewood Cliffs, NJ: Prentice Hall, 1997.
[30] Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, "Joint scalable video model JSVM-9," *Doc. JVT-V202*, Marrakech, Morocco, Jan. 2007.
[31] A. Segall, "CE2: bit-depth scalability," Joint Video Team, Doc. JVT-W302, San Jose, Apr. 2007.
[32] Y. Gao, A. Segall, and T. Wiegand, "Report of AhG on bit-depth and chroma format scalability," Joint Video Team, Doc. JVT-W010, San Jose, April 2007.
[33] G. J. Sullivan, and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 74–90, Jun. 1998.

Yongying Gao received the B.S. and M.E. degrees from Tsinghua University, Beijing, China, in 1998 and 2000, respectively, and the Ph.D. degree from Michigan State University, East Lansing, in 2005.

From 2005–2007, she was a Research Engineer in Thomson Corporate Research Beijing, Beijing, China. She is currently a Researcher in the Advanced Technology Division of MediaTeck (Beijing) Corporation, Beijing, China. Her research interests include image/video processing and computer vision.

Yuwen Wu received the B.S. degree from Shandong University, ShanDong, China, in 1999. He received the Ph.D. degree from the National Laboratory on Machine Perception by Peking University, Beijing, China, in 2005.

He joined Thomson Corporate Research Beijing, Beijing, China, in 2005, where he is currently a Senior Research Engineer. His research interests include image/video processing and compression, computer vision, and graphics.

Ying Chen (M'08) received the B.S. and M.S. degrees at the School of Mathematical Sciences and School of Electronics Engineering and Computer Science, both from Peking University, in 2001 and 2004, respectively.

He is currently a Researcher in the Department of Signal Processing, Tampere University of Technology (TUT), Tampere, Finland. He is also an External Member of the research staff at the Nokia Research Center, Finland, since September 2006. Before joining TUT, he was a Research Engineer at the Thomson Corporate Research Beijing, Beijing, China, from August 2004. His research interests include image processing and video coding and transmission. He has been an active contributor to ITU-T JVT and ISO/IEC MPEG, focusing on the scalable video coding, and multiview video coding standards. He has coauthored over 60 technical standardization reports and published over 20 academic papers, and has over 20 issued or pending patents.

[P3] Y. Chen, Y. -K. Wang, K. Ugur, M. M. Hannuksela, J. Lainema, and M. Gabbouj, “The Emerging MVC Standard for 3D Video Services,” *EURASIP Journal on Advances in Signal Processing*, Volume 2009, Article ID 786015.

Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2009, Article ID 786015, 13 pages
doi:10.1155/2009/786015

Review Article

The Emerging MVC Standard for 3D Video Services

Ying Chen,[1] Ye-Kui Wang,[2] Kemal Ugur,[2] Miska M. Hannuksela,[2] Jani Lainema,[2] and Moncef Gabbouj[1]

[1] *Department of Signal Processing, Tampere University of Technology, 33720 Tampere, Finland*
[2] *Nokia Research Center, Visiokatu 1, 33720 Tampere, Finland*

Correspondence should be addressed to Ying Chen, ying.chen@tut.fi

Received 1 October 2007; Revised 7 February 2008; Accepted 5 March 2008

Recommended by Aljoscha Smolic

Multiview video has gained a wide interest recently. The huge amount of data needed to be processed by multiview applications is a heavy burden for both transmission and decoding. The joint video team has recently devoted part of its effort to extend the widely deployed H.264/AVC standard to handle multiview video coding (MVC). The MVC extension of H.264/AVC includes a number of new techniques for improved coding efficiency, reduced decoding complexity, and new functionalities for multiview operations. MVC takes advantage of some of the interfaces and transport mechanisms introduced for the scalable video coding (SVC) extension of H.264/AVC, but the system level integration of MVC is conceptually more challenging as the decoder output may contain more than one view and can consist of any combination of the views with any temporal level. The generation of all the output views also requires careful consideration and control of the available decoder resources. In this paper, multiview applications and solutions to support generic multiview as well as 3D services are introduced. The proposed solutions, which have been adopted to the draft MVC specification, cover a wide range of requirements for 3D video related to interface, transport of the MVC bitstreams, and MVC decoder resource management. The features that have been introduced in MVC to support these solutions include marking of reference pictures, supporting for efficient view switching, structuring of the bitstream, signalling of view scalability supplemental enhancement information (SEI) and parallel decoding SEI.

1. Introduction

Three-dimensional video has gained significant interest recently. Furthermore, with the advances in acquisition and display technologies, 3D video is becoming a reality in consumer domain with different application opportunities. Given a certain maturity of capture and display technologies and with the help of multiview video coding (MVC) techniques, a number of different envisioned 3D video applications are getting feasible [1]. 3D video applications can be grouped under three categories: free-viewpoint video, 3D TV, and immersive teleconferencing. The requirements of these applications are quite different and each category has its own challenges to be addressed.

1.1. Application Scenarios. To illustrate these challenges, consider Figure 1, where the end-to-end architecture of different applications is shown. In this illustration, a multiview video is first captured and then encoded by a multiview video coding (MVC) encoder. A server transmits the coded bitstream(s) to different clients with different capabilities, possibly through media gateways. The media gateway is an intelligent device, also referred to as a media-aware network element (MANE), which is in the signaling context and may manipulate the incoming video packets (rather than simply forward packets). At the final stage, coded video is decoded and rendered with different means according to the application scenario and capabilities of the receiver. To provide smoothly immersive experience when a user adjusting its viewing position, view synthesis [2, 3] may be required at the client to generate "virtual" views of a real-world scene. However, till now, this process is out of the scope of any existing coding standard.

In free-viewpoint video, the viewer can interactively choose his/her viewpoint in 3D space to observe a real-world scene from preferred perspectives [4]. It provides realistic impressions with interactivity, that is, the viewer can navigate freely in the scene within a certain range, and

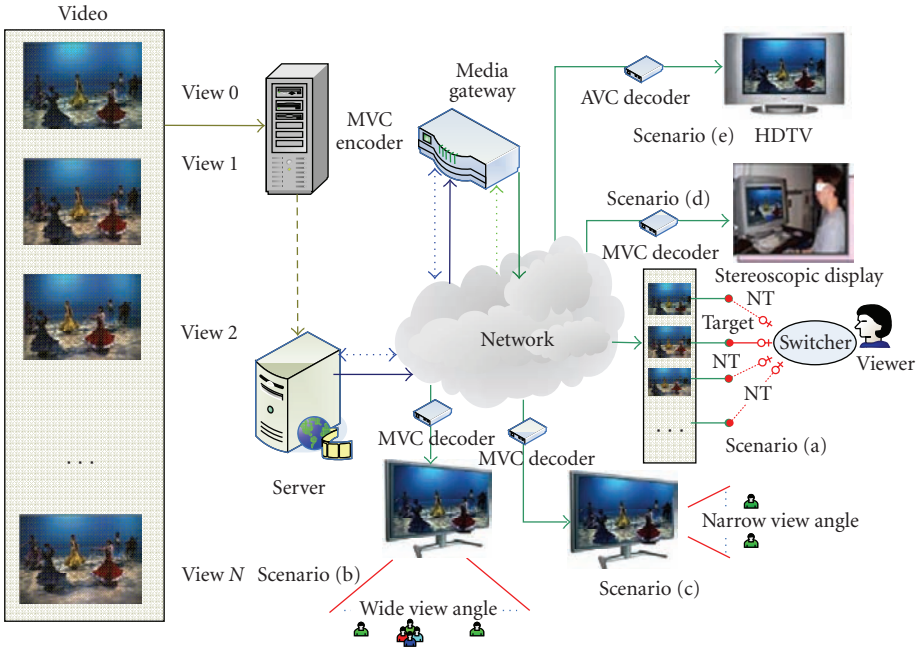


Figure 1: MVC system architecture.

analyze the 3D scene from different viewing angles. Such a video communication system has been reported in [5]. Unlike holography, which generates 3D representation and requires changing of the relative geometry position of a viewer to switch view point, this scenario is actually realized by switching between rendered view(s) using interface such as remote controller. In case the desired viewpoint is not available, interpolating a virtual view from other available views can be employed. Scenario (a), in Figure 1, illustrates this application, where there exist several candidate views for the viewer, and one of them is selected as the target view that is displayed (views that are not targeted and thus are not outputted are denoted as "NT" for simplicity in Figure 1). In this scenario, not all the candidate views are required to be decoded, thus the decoder can focus its resources only on decoding of the target view. For this purpose, the target view needs to be efficiently extracted from the bitstream and thus only the packets that are required for successfully decoding the desired views are transmitted. To enable navigation in a scene, important functionality to be achieved by the system is efficient switching between different views.

3D TV refers to the extension of traditional 2D TV displays-to-displays capable of 3D rendering. In this application, more than one view is decoded and displayed simultaneously [6]. A simple 3D TV application can be realized by stereoscopic video. Stereoscopic display can be achieved by using data glasses or other means. However, it is nicer for the user to get the 3D feeling directly through 3D appliances with added feature of rendering binocular depth cures [7], which can be realized by autostereoscopic displays. Advanced autostereoscopic displays can support head-motion parallax, by decoding and displaying multiple views from different viewpoints simultaneously. That is, a viewer without extra facilities like data glasses can move to different geometry angle ranges, each of which contains typically two views rendered and shed by 3D displays. 3D TV displays are discussed in [8]. The viewer then can experience a slightly different scene by moving his/her head (for example, user may look what is behind a certain object in the scene). In this scenario, multiple views need to be decoded simultaneously; therefore parallel processing of different views is very important to realize this application. In addition, displaying multiple views is important also to realize wide viewing angle as shown in Figuer 1(b). This scenario is also referred to as autostereoscopic 3D TV for multiple viewers [7]. However, if the decoder capability is limited or the transmission bandwidth decreases, the client at a receiver may simply decode and render just a subset of the views but still provide 3D display with a narrow view angle, as shown in Figure 1(c). The media gateway plays an important role to provide the adaptation functionality to support this use case. Such 3D TV broadcast or multicast system must then support flexible stream adaptation. Stream adaptation can be achieved at the server or media gateway, where only the sub-bitstreams, with less bandwidth and desired by the client are transmitted and other packets are discarded. After bitstream extraction, the sub-bitstream must be decodable for by MVC decoders.

Free-viewpoint video focuses on its functionality in free navigation while 3D TV emphasizes on 3D experience. In immersive teleconference, both interactivity and virtual reality may be preferred by the participants and thus free viewpoint or 3DTV style can be both supported. In the immersive teleconferencing, where there is interactivity among viewers, immersiveness can be achieved either in

a free-viewpoint video or 3D TV manner. So, the problems or requirements in free-viewpoint video or 3D TV are still existing and valid.

Typically, two mechanisms can make people perceptually feel immersed in a 3D environment. A typical technique, known as head-mounted display (HMD), needs a device worn on the head, as a helmet, which has a small display optic in front of each eye. This scenario is shown in Figure 1(d). Substitutions for HMD need to introduce head tracking [9] or gaze tracking [10] techniques, as shown in the solutions discussed in [7]. In 3D TV, however, each stereoscopic display can have effect on a certain small range of a view angle, thus, a viewer can change his/her viewing position when he/she is trying to view the scene in another viewpoint, as if there was a natural object.

For rendering of 3D TV content or view synthesis, depth information is needed. Depth-images storing the depth information as a monoscopic color video can be coded with existing coding standards, for example, as auxiliary pictures in H.264/AVC [11].

As the normal 2D TV or HDTV applications are still dominating the market, the MVC content will provide a way for those 2D decoders, for example, H.264/AVC decoder in the set-top box (STB) of digital TV to generate a display from an MVC bistream, as shown in Figure 1(e). This requires MVC bitstreams to be backward compatible, for example, to H.264/AVC.

1.2. Requirements of MVC. Due to the huge amount of data, particularly when the number of views to be decoded is large, transmission of multiview video applications relies heavily on the compression of the video captured by cameras. Therefore, efficient compression of multiview video contents is the primary challenge for realizing multiview video services.

A natural way to improve compression efficiency of multiview video content is to exploit the correlation between views, in addition to the use of inter prediction in monoview coding. This requires buffering of additional decoded pictures. When the number of views is large, the required memory buffer may be prohibitive. In order to make efficient implementations of MVC feasible, the codec design should include efficient memory management of decoded pictures.

The above challenges and requirements, among others [12], are the basis of the objectives for the emerging MVC standard, which is under development by the joint video team (JVT), and will become the multiview extension of H.264/AVC [11]. MVC standardization in the JVT started in July 2006 and is expected to be finalized in mid-2008. The most recent draft of MVC is available in [13].

In the MVC standard draft, redundancies among views are utilized to improve compression efficiency compared to independent coding of views. This is allowed with the so-called interview prediction, in which decoded pictures of other views can be used as reference pictures when coding a picture as long as they all share the same capturing or output time. View dependencies for interview prediction are defined for each coded video sequence.

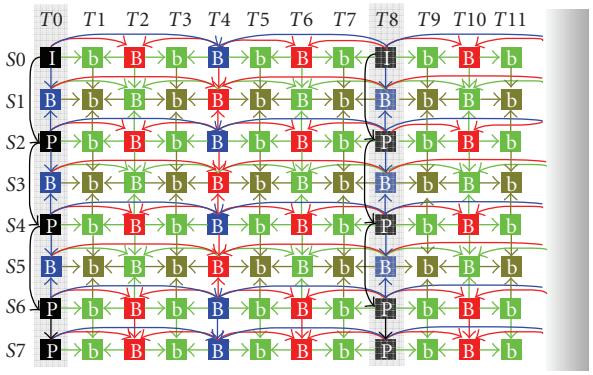


Figure 2: Typical MVC prediction structure.

With the exception of interview prediction, pictures of each view are coded with the tools supported by H.264/AVC. In particular, hierarchical temporal scalability was found to be efficient for multiview coding [14]. A typical prediction structure of MVC, utilizing both interview prediction and hierarchical temporal scalability, is shown in Figure 2. It is noted that the MVC standard provides a greater deal of flexibility than depicted in Figure 2 for arranging temporal or view prediction references [15].

Except the coding efficiency requirement, the following important aspects of the MVC requirements [12] for the design of the MVC standard are listed.

1.2.1. Scalabilities. View scalability and temporal scalability are considered in the MVC design for the adaptation of user preference, network bandwidth, and decoder complexity. View scalability is useful in the scenario shown in Figure 1(c), wherein some of the views are not transmitted and decoded.

1.2.2. Decoder Resource Consumption. In 3D TV scenarios, as shown in Figures 1(b) and 1(c), a number of views are to be decoded and displayed, an optimal decoder in terms of memory and complexity is of vital importance to make the real-time decoding of MVC bitstreams possible.

1.2.3. Parallel Processing. In the 3D TV scenarios, since multiple views need to be decoded simultaneously, parallel processing of different views is very important to realize this application and to reduce the computation time to achieve real-time decoding.

1.2.4. Random Access. Besides temporal random access, view random access is to be supported to enable accessing a frame in a given view with minimal decoding of frames in the view dimension. For example, free-viewpoint video described in Figure 1(a) needs advanced view random access functionality to support smooth navigation.

1.2.5. Robustness. When transmitted in a lossy channel, the MVC bitstream will have error resiliency capabilities. There are error resilient tools in H.264/AVC which can benefit the

MVC applications. Other techniques, which are designed only for MVC and discussed later, can also be utilized to improve error resilience of MVC bitstreams.

1.3. Contributions of this Paper. JVT has recently finalized the scalable extension of H.264/AVC, also known as scalable video coding (SVC) [16]. MVC shares some design principles with SVC, such as backward compatibility with H.264/AVC, temporal scalability, and network friendly adaptation, and many features in SVC have been reused in MVC.

However, new mechanisms are needed in MVC at least related to view scalability, interview prediction structure, coexisting of decoded pictures from multiple dimensions (i.e., both the temporal and view dimensions) in the decoded picture buffer, multiple representations in the display, and parallel decoding at the decoder.

These mechanisms cover the challenges and requirements, identified above, for 3D video services, except for the compression efficiency challenge. In this paper, we will describe how these mechanisms are realized in the existing draft MVC standard.

The main MVC features discussed in this paper include reference picture management to achieve optimal memory consumption at the decoder, time-first coding to support consistent system level design, SEI messages, and other features for view and scalability information provisioning, adaptation, random access, view switching, and reference picture list construction.

The rest of this paper is organized as follows. In Section 2, we discuss the MVC bitstream structure and the backward compatibility which is mentioned in Scenario (e). In Section 3, with a typical application scenario, we discuss how adaptation works when connectivity between server and client or decoder capacity varies. Then, view scalability information SEI message, which is designed to facilitate the storage, exaction, and adaptation of MVC bitstream, is reviewed. The features discussed in this section are of importance for efficient file composition, bitstream exaction, and stream adaptation in intermediate media gateways, which has been mentioned in Scenario (c). Random access and view switching functionalities are described in Section 4, which is desirable in Scenario (a). In Section 5, the decoded picture buffer management is discussed. This topic is crucial to enable a system to minimize the required memory for decoding MVC bitstreams. In Section 6, the parallel decoding SEI message, which is important for real-time MVC decoder solutions, is discussed. Other related issues are summarized in Section 7. Finally, Section 8 concludes the paper.

2. Structure of MVC Bitstreams

This section reviews the concept of network abstraction layer units (NAL units) and summarizes how the NAL unit types defined in H.264/AVC and SVC are reused for MVC. Syntax elements in the NAL unit header in the MVC context are also discussed.

In H.264/AVC, the coded video bits are organized into NAL units. NAL units can be categorized to video coding layer (VCL) NAL units and non-VCL NAL units. The supported VCL NAL unit types and non-VCL NAL units in H.264/AVC are defined in [11] and well categorized in [17].

In MVC, there is a base view, which is coded independently and is compliant with H.264/AVC, this meets the requirement in Scenario (e) of the MVC system architecture, as shown in Figure 1. Consequently, coded picture information for the base view is included in the VCL NAL units specified in H.264/AVC. A new NAL unit type, called coded slice of MVC extension, is used for containing coded picture information for nonbase views. When an MVC bitstream containing NAL units of the new NAL unit type is fed to an H.264/AVC decoder, NAL units of any new NAL unit type can be ignored and the decoder only decodes the bitstream subset containing NAL units of the existing NAL unit types defined in H.264/AVC.

There are useful properties of the coded pictures in the H.264/AVC-compliant base view, such as temporal level, which are not indicated in the VCL NAL units of H.264/AVC. To indicate those properties for the base view-coded pictures, the prefix NAL unit, of another new NAL unit type, has been introduced. Note that prefix NAL unit is also specified in SVC. A prefix NAL unit precedes each H.264/AVC VCL NAL unit and contains its essential characteristics in multiview context. As H.264/AVC decoders ignore prefix NAL units, the backward compatibility to H.264/AVC is still maintained.

Non-VCL NAL units include parameter set NAL units and SEI NAL units among others. Parameter sets contain the sequence-level header information (in sequence parameter sets (SPS)) and the infrequently changing picture-level header information (in picture parameter sets (PPS)). With parameter sets, this infrequently changing information needs not to be repeated for each sequence or picture, hence coding efficiency is improved. Furthermore, the use of parameter sets enables out-of-band transmission of the important header information, avoiding the need of redundant transmissions for error resilience. In "out-of-band" transmission, parameter set NAL units are transmitted in a more different channel than the ones for transmission of other NAL units. More discussions on parameter sets can be found in [18].

In MVC, coded pictures from different views may use different sequence parameter sets. An SPS in MVC can contain the view dependency information for interview prediction. This enables signaling-aware media gateways to construct the view dependency tree. Therefore, each view can be mapped to the view dependency tree and view scalability can be fulfilled, without any extra signaling inside NAL unit headers [19].

The scalable nesting SEI message [19], which was also introduced in SVC with the same name, is set apart from other SEI messages in that it contains one or more ordinary SEI messages, but in addition it indicates the scope of views or temporal levels for which the messages apply. In doing so, it enables the reuse of the syntax of H.264/AVC SEI messages for a specific set of views and temporal levels.

Some of the other SEI messages specified in MVC are related to the indication of output views, available operation points, and information for parallel decoding.

In H.264/AVC, an NAL unit consists of a 1-byte header and an NAL unit payload of varying size. In MVC, this structure is retained except for prefix NAL units and MVC-coded slice NAL units, which consist of a 4-byte header and the NAL unit payload. New syntax elements in MVC NAL unit header include priority_id, temporal_id, anchor_pic_flag, view_id, idr_flag and inter_view_flag.

anchor_pic_flag indicates whether a picture is an anchor picture or nonanchor picture. Anchor pictures and all the pictures succeeding in output order (i.e., display order) can be correctly decoded without decoding of previous pictures in decoding order (i.e., bitstream order) and thus can be used as random access points. Anchor pictures and nonanchor pictures can have different dependencies, both of which are signaled in the sequence parameter set.

More discussions on anchor pictures will be given in Section 4. idr_flag is introduced in Section 4, inter_view_flag is discussed in Section 5, and the other new MVC NAL unit header fields are introduced in Section 3.

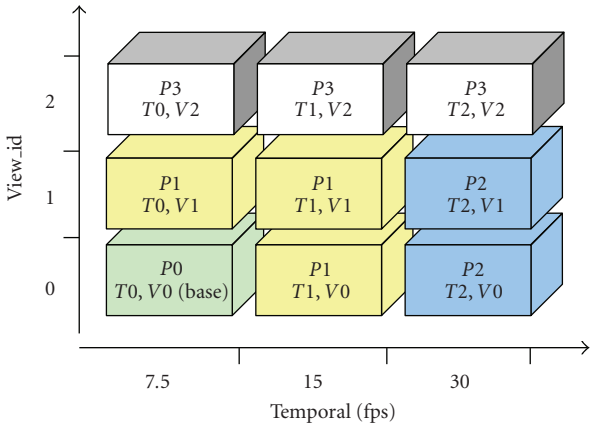
3. Extraction and Adaptation of MVC Bitstreams

MVC supports temporal scalability and view scalability. A portion of an MVC bitstream can correspond to an operation point that gives output representation for a certain frame rate and a number of target views. Data representing higher frame rate, views closer to the leaves of the dependency tree, or views that are not preferred by the client can be truncated during the stream bandwidth adaptation at the server or media gateway, or ignored at the decoder for complexity adaptation.

The bitstream structure defined in MVC is characterized by two syntax elements: view_id and temporal_id. The syntax element view_id indicates the identifier of each view. This indication in NAL unit header enables easy identification of NAL units at the decoder and quick access of the decoded views for display. The syntax element temporal_id indicates the temporal scalability hierarchy or, indirectly, the frame rate. An operation point including NAL units with a smaller maximum temporal_id value has a lower frame rate than an operation point with a larger maximum temporal_id value. Coded pictures with a higher temporal_id value typically depend on the coded pictures with lower temporal_id values within a view, but never depend on any coded picture with higher temporal_id.

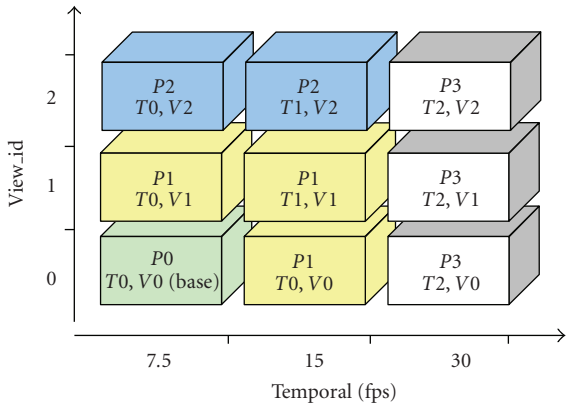
The syntax elements view_id and temporal_id in the NAL unit header are important for both bitstream extraction and adaptation. Another important syntax element in the NAL unit header is priority_id [19], which is mainly used for the simple one-path bitstream adaptation process.

Whenever the operation point contains only a subset of the entire MVC bitstream, such as in Scenario (a) and Scenario (c) shown in Figure 1, a bitstream extraction process is then needed to exact the required NAL units from the entire bitstream. The bitstream extraction process should be a lightweight process without heavy parsing of the bitstream. For this purpose, the mapping between each operation point (identified by the combination of required view_id values and temporal_id values) and the required NAL units is specified as part of the view scalability information SEI message (VSSEI) [20]. After the operation point is agreed upon, the server can simply extract the required bitstream subset by discarding nonrequired NAL units by checking the view_id and temporal_id values in the fixed-length coded NAL unit headers.



Path:
$P = 0$: view 0/7.5
$P = 1$: view 0, 1/15
$P = 2$: view 0, 1/30
$P = 3$: view 0, 1, 2/30

(a)



Path:
$P = 0$: view 0/7.5
$P = 1$: view 0, 1/15
$P = 2$: view 0, 1, 2/15
$P = 3$: view 0, 1, 2/30

(b)

Figure 3: Assignment of priority_id for NAL units of a 3-view bitstream with two levels of temporal resolution. *T: temporal level; V: view identifier; P: priority identifier. Temporal level equal to 0 corresponds to 7.5 fps (frame per second), it equal to 1 corresponds to 15 fps, and it equal to 2 corresponds to 30 fps.*

Media gateways can perform single-path adaptation by simply discarding NAL units with priority_id greater than a certain value. The priority_id has no normative effect on the decoding process. The only constraint to priority_id values is that any bitstream subset extracted based on any value of priority_id must be a conforming MVC bitstream. It is the encoder responsibility to set priority_id values for the NAL units and the values can be rewritten, for example, when the preference of the decoder changes.

Figure 3 depicts two examples of priority_id assignments which yield two different adaptation paths for the same MVC bitstream that contains 3 views with 3 temporal levels. In Figure 3(a), the priority_id is assigned such that the 7.5 Hz base view is with priority_id equal to 0, and then frame rate of 15 Hz including both view 0 and view 2 is with priority_id equal to 1, and then higher frame rate is preferred to more views. In Figure 3(b), the first two steps are the same as in Figure 3(a), while in the last two steps, more views are preferred to higher frame rate.

Although a simple media gateway may perform stream adaptation exclusively based on priority_id, more intelligent implementations may jointly employ the values of priority_id, view_id, and temporal_id, in order to perform combined adaptation. For example, for the bitstream discussed in Figure 3, there can be two adaptation steps, the first step is to have NAL units with temporal_id equal to 1 (15 Hz) and view_id through 0 to 1; the second step is to increase frame rate directly to 30 Hz and include all the NAL units in view 2. Note that in this case, the NAL units corresponding to each adaptation step can have different values of priority_id, for example, when the priority_id assignment follows Figure 3(a).

An MVC bitstream may contain a large number of views (the view_id in the current MVC draft specification is of 10 bits). This makes the possible number of combinations of view_id values and temporal_id values huge. However, in practical applications, typically only limited combinations, that is, operation points, would be used. The VSSEI has been designed to be flexible to signal any subset of all the possible operation points. Beside the mapping of operation points and NAL units, the following information for each indicated operation point is also included in the VSSEI, either to enable the establishment of the communication session or more efficient bitstream extraction or adaptation.

Profile and level: This information describes the capacity a decoder requires to decode a bitstream. Profile and level can be signaled in the SPS. However, the total number of SPS is limited to a certain value in the bitstream and it may happen that for all the operation points, many of them share the same SPS, the level inside which is not accurate enough to describe the minimum required capacity of the decoders for different operation points. Therefore, profile and level are signaled in the VSSEI for each operation point.

Bit rate: Similar as profile and level, this information is needed in the session negotiation process for the server and the client to agree upon a certain operation point. This information is also useful in rate adaptation by MANEs. For example, to better adapt the bandwidth, it is necessary for intelligent media gateways to know the bandwidth of a session when it switches to another operation point.

Operation point dependencies: In the VSSEI, each operation point is identified by the view_id values of the target views and the temporal_id values. The dependent views as well as the dependent pictures may be known from the active SPS which contains the view dependency information. However, within the view dependency, pictures may have more flexible relationship. For example, assume in a two-view bitstream with 30 fps, 4 temporal levels and according to the SPS MVC extension anchor pictures and nonanchor pictures in view 1 are, respectively, dependent on anchor and nonanchor pictures in view 0. And if we have two operation points (OPs), OP 0 has the pictures in view 0 with temporal level up to 3, that is, 15 fps and OP 1 has pictures with all the pictures in view 1, however, the pictures with the highest temporal level in view 1 do not really rely on interview pictures for reference. Then, OP 1 actually depends only on OP 0, which cantinas half of the pictures in view 0 and the highest temporal level pictures in view 0 can be neglected for transmission and decoding. However, with only the view dependency signaled in the SPS MVC extension, those pictures are still required to be transmitted and decoded. Thus, operation point dependency information included in the VSSEI would enable simply identification and discarding of the nonrequired NAL units that are not indicated by the view dependency information signaled in SPS.

In the following are some MVC stream adaptation examples in a broadcasting system (see Figure 1). Assume that the entire bitstream contains coded pictures of 8 views.

For Scenario (e), NAL units are filtered by the MANE so that only the NAL units that can be recognized by H.264/AVC decoders (by checking the NAL unit type) are fed to the STB of an HDTV.

For Scenario (d), an operation point containing, for example, only view 0 and view 1 is in use. The MANE controls the bitstream in a way that only allows the NAL units with view_id (by checking the view_id in the NAL unit header) equal to 0 or 1 to be sent to the client.

Depending on the bandwidth, a client with enough decoding capability for 3D TV may switch between Scenarios (b) and (c), wherein the sub-bitstream corresponding to Scenario (b) forms an operation point that contains only a subset of the views within a narrow view angle. The MANE filters out the views outside the view angle.

4. Random Access and View Switching

4.1. Random Access. Random access refers to starting decoding of a bitstream from a point other than the beginning. The support of random access is required for traditional trick play modes such as fast forward and fast backward. In streaming applications, random access is used to seek the desired playback position requested by the users. In broadcast and multicast applications, random access points are required to allow for newcomers to tune in or switching of program channels.

Random access with MVC for the above purposes is not much different from that with single-view coding, as

all the target views of an operation point are accessed simultaneously. The only difference is that there may be views dependent on by the target views; hence these dependent views need also to be accessed and decoded.

To access to a picture in a given view at a specified time, the decoder should first find the closest preceding temporal locations that are random access points to the specific target view and all the dependent views, collectively referred to as the required views. Then the decoder starts decoding the required views from a found location. In average, how many view pictures need to be decoded to access to a specific target picture is therefore proportional to the random access period (i.e., the length of the temporal dependency chain) and the number of dependent views (i.e., the length of the interview dependency chain).

Instantaneous decoding refresh (IDR) pictures are natural random access points. In an MVC bitstream, IDR pictures in the base view have NAL units of type 5. If the bitstream also contains NAL units that are unknown to plain H.264/AVC decoders, then the base view IDR picture NAL units are each preceded by a prefix NAL unit, which has idr_flag equal to 1. IDR pictures of nonbase views, also referred to as view-IDR (V-IDR) pictures in the draft MVC standard, all have idr_flag equal to 1. V-IDR pictures may rely on pictures from other views but only within the same access units though interview prediction [21].

An access unit contains all the NAL units pertaining to a certain time instance. According to the draft MVC standard, an IDR access unit is an access unit wherein the pictures of all the views are IDR pictures. Such an IDR access units provide random access support at the time instance to all the views. Note that the draft MVC standard allows for such access unit wherein pictures of some views are IDR pictures while pictures of other views are non-IDR pictures.

IDR pictures disallow any picture succeeding the IDR picture in decoding order (i.e., bitstream order) to be inter-predicted from earlier pictures in the same view. This leads to a reduced compression efficiency compared to the typical open GOP (group of pictures) coding structures such as the IBBP structure, where the B pictures after the I picture in decoding order precede the I picture in display order, and can use pictures before the I picture in decoding order for inter prediction. The I pictures in such open GOP coding structures are defined as anchor pictures in the draft MVC standard and are identified by the NAL unit header syntax element anchor_pic_flag equal to 1. Anchor pictures can therefore also be used as random access points, while application implementers must bear in mind that a few pictures after such random access points may not be correctly decoded when random access is carried out at these points. Actually, in this situation these pictures can be dropped from the bitstream sent to the user. Like V-IDR pictures, anchor pictures in nonbase views can also use interview prediction.

It is also possible to perform random access at non-intra pictures, for example, using gradual decoding refresh (GDR) based on the isolated regions technology [22]. In this case, the GDR random access points can be indicated by the recovery point SEI message as specified in H.264/AVC, but included in the scalable nesting SEI message that tells to which views the semantics apply.

4.2. View Switching. View switching refers to changing the target view(s). The number of target view(s) may be one or more. In case the number of target view(s) change or any of the target view is changed from one view to another, a view switching occurs. View switching must happen at view-switching points, after which the new target view(s) can be correctly decoded. A typical application for view switching is free-viewpoint video, which has been shown in Scenario (a) of Figure 1.

All random access points can also be used as view switching points. There is another type of switching points that are not random access points. For example, if at picture X the target views can be switched to view subset C from view subset A but not from view subset B, then picture X is a view-switching point from view subset A to view subset C. This type of switching points can be realized by specifically setting the interview prediction relationship, or by using the SP/SI coding technology [23].

5. Decoded Picture Buffer Management

In this section, we first introduce the decoding order arrangement of coded view pictures, which is closely related to decoded picture buffer management. After that, we present an analysis of the buffer requirement for decoding of MVC bitstreams, which has been discussed in more details in [24]. Finally, reference picture management methods both inside a view and related to interview pictures are discussed.

5.1. Decoding-Order Arrangement. In H.264/AVC, the order how NAL units are placed inside the bitstream is referred to as the decoding order. In multiview video, where two dimensions, time and view, are involved, prescription of the decoding order gets more complicated.

Two fundamentally different decoding order arrangements, view-first coding and time-first coding, have been considered by the JVT. In view-first coding [25], within each group of pictures (GOP), pictures of each view are contiguous in decoding order, as shown in Figure 4, where the horizontal direction denotes time (each time instance is represented by Tm), and the vertical direction denotes view (each view is represented by Sn). Pictures of each view are grouped into GOPs, for example, pictures T1 to T8 for any view in Figure 2 form a GOP.

View-first codingcauses a fundamental problem for storage of multiview video bitstreams in media container files based on ISO base media file format [26]. Coded pictures belonging to different views but with the same time instance are interleaved with pictures of other time instances in a bitstream, and thus cannot be in the same access unit. These different access units, when composed into a file according to the ISO base media file format, correspond to different samples. The ISO base media file format requires samples to be ordered in their decoding order. According to the ISO base media file format, the decoding time of a sample is an

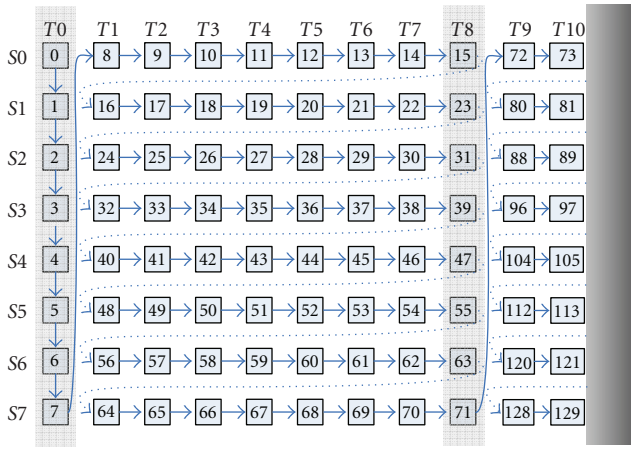


Figure 4: View-first coding.

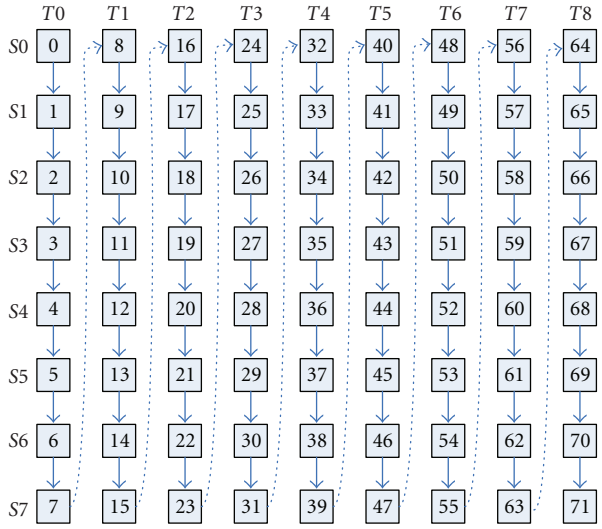


Figure 5: Time-first coding.

increasing function of sample number, and the composition time (also used as presentation time) of a sample is indicated as a nonnegative increment compared to its decoding time. Consequently, view-first coding would require a composition time offset proportional to the GOP size multiplied by the number of views, which would be perceived as significant initial buffering delay. Furthermore, possibility for parallel decoding would be hard to realize when view-first coded streams are included in files compliant with ISO base media file format, because the indicated decoding and composition times assumed single-processor operation.

To overcome the mentioned problems, time-first coding was introduced in MVC [27]. In time-first coding, pictures of any temporal location are contiguous in decoding order, as shown in Figure 5. In this case, we can define pictures of the same time instance but belonging to different views as one access unit. Note that the decoding order of access units may not be identical to the presentation order.

With time-first coding, an access unit contains NAL units continuous in decoding order. This definition is similar to the access unit definition in SVC. Therefore, many mechanisms designed in the SVC file format, such as extractors and aggregators, are useful for MVC too. Some design principle for MVC file format can be found in [28].

The following subsections on buffer requirement analysis and buffer management are all for time-first coding only.

5.2. Buffer Requirement Analysis. In MVC, pictures in the same time instance are assumed to be outputted simultaneously. Decoded pictures used for prediction or future output are buffered in the decoded picture buffer (DPB). To efficiently utilize the buffer memory, the DPB management processes have been specified, which include a storage process of decoded pictures into the DPB, a marking process of reference pictures, and an output and removal process of decoded pictures from the DPB.

Assume that we have a prediction structure similar to the one shown in Figure 2, where each GOP includes a number of views (nv) and in each view *gl* (GOP length) pictures.

The optimal DPB size, as discussed in [29], is TL + 1, where TL is the highest temporal level of all the pictures and $\mathrm{TL} = \lceil \log_2(gl) \rceil$.

The DPB sizes for time-first coding in different scenarios are summarized in the following, while more details can be found in [24, 30].

5.2.1. DPB When Output is not Taken into Consideration. In time-first coding, the pictures in the same time instance will be stored in the DPB longer and each view preserves the hierarchical B coding structure. So there are two steps to reach the maximum DPB size for time first:

(1) take the pictures in the same time instance as a whole and form a hierarchical B coding structure, the DPB size would then be $nv \cdot (\mathrm{TL} + 1)$;

(2) for the nonreference pictures in the highest temporal level, interview prediction requires them to be stored in the DPB.

These two steps are shown in Figure 6.

So, in the typical prediction structure, the maximum DPB size for time-first coding is $nv \cdot (\mathrm{TL} + 1) + 2$.

In both results, there is a "2", which actually means the maximum interview reference pictures in the typical prediction structure.

5.2.2. DPB When the Output is Taken into Consideration. When the output is considered, thecase is even worse for view-first coding, especially for 3D TV application scenario, which requires the display of all the views. The reason is that, in view-first coding, all the pictures of the already coded view in a GOP must be kept in the buffer at least till the last view starts decoding.

For simplicity, we give the DPB buffer sizes for view-first coding and time-first coding in both 3D TV and free-viewpoint video scenarios without detailed analysis, which can be found in [24].

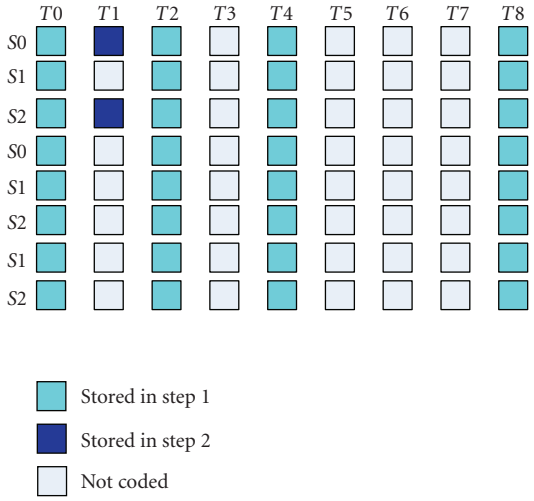


Figure 6: DPB status for time-first coding.

In 3D TV scenario, the total DPB sizes for time-first coding are $(nv-1)\,gl+\mathrm{TL}+1$ and $nv(2\cdot\mathrm{TL}-\log_2\lfloor \mathrm{TL}-1\rfloor)$, respectively.

In free-viewpoint video, the total DPB size for time-first coding is $(nv\cdot(\mathrm{TL}+1)+3)/2+\mathrm{TL}-1-\log_2\lfloor \mathrm{TL}-1\rfloor$.

Table 1 gives the example values for all the compared scenarios when the GOP length is 16 and number of views are 8. Time-first coding, as shown by the formula as well as the example values, requires less DPB size.

Note. Scenarios through (1) to (4) are the following scenarios, respectively: (1) DPB w/o output; (2) 3D TV DPB with output; (3) Free viewpoint video DPB with output, maximum; (4) Free viewpoint video DPB with output, average gl is 16 and nv is 8.

5.3. Buffer Management Inside a View. Because of the time-first coding structure, whether a picture is a reference picture or nonreference picture can be decided only by its temporal prediction structure. Because for any two pictures in a view, if picture A follows picture B, then in the whole bitstream, picture A also follows any picture with the same time instance as picture B. This is not the case in view-first coding, so it may require cross-view explicitly or implicit marking to make those pictures with the same time instance as B but with early decoding time as A as "unused for reference".

So, all the memory management control operation commands, if present, are effective inside a view. And the sliding window also takes effect inside a view, which was proposed into JVT in the same time in [30–32].

5.4. Buffer Management for Interview Reference Pictures. In each time instance, if dependency exists, for one current decoding picture, there can be one or more interview reference pictures. Those interview reference pictures, although there are not used for temporal prediction within a view, are required to be somehow stored in the decoded picture buffer.

Table 1: Comparison examples between view-first and time-first when different scenarios are utilized.

	(1)	(2)	(3)	(4)
view-first	44	117	44	32
time-first	44	56	56	23.5

However, whether to store these pictures as "used for reference" or "unused for reference" is still an issue. In the AVC specification, if a picture is not used as a reference picture for others, it is with a nal_ref_idc value equal to 0 and is a nonreference picture. Those pictures, however, in MVC context can be used for interview reference picture, for example, the highest temporal level pictures in view 0 when view 1 is decoded.

If there are stored as a reference picture, when only base view sub-bitstream is decoded, it is definitely an extra memory burden for the H.264/AVC decoder and the encoder may need to design extra memory management control operation (MMCO) commands. So, in [33], we proposed that those pictures are not required to be stored as a reference picture. This solution solves the problem we mentioned above and another question arises: how would those pictures used only for interview prediction be managed to reach the optimal buffer management. One argument is if an interview picture is not used for temporal prediction and is a nonreference picture, it may be not available in the DPB.

Because of the time-first coding structure and the assumption that pictures are outputted at the same time, the concern mentioned above is solved.

So there is no extra marking process for those pictures if all views are required for output. If some views are not required for output, those pictures can be implicitly removed from the DPB earlier. The implicit removal is based on the view dependency defined in the MVC SPS extension [19, 34]. The implicit removal is defined in the hypothetical reference decoder (HRD) part of the MVC specification. The current HRD design of MVC focuses mostly on output conformance.

Although the interview prediction structure is in the scope of MVC SPS extensions, for each time instance, a picture can be used as interview picture or not based on real uses. For example, pictures in higher temporal levels may be more helpful for the efficiency while picture in lower temporal levels may be less helpful. The decoding of those pictures, if they are nonreference pictures and belong to the views that are not required for output, can be avoided. So an inter_view_flag was proposed by [35, 36] and was introduced into the MVC specification.

6. Parallel Coding of Multiple Views

One of the key identified requirements for the MVC standard is its ability to support parallel processing of different views [11]. The parallel processing of different views is especially important for 3D broadcasting use cases, where the displays need to output many views simultaneously to support head-motion parallax. However, interview dependencies between

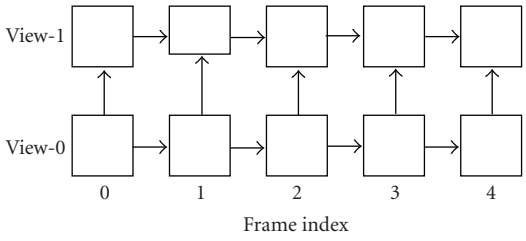


Figure 7: Sample prediction structure for two views.

pictures may impose serious parallelism issues to the video system, because two pictures at different views need to be decoded sequentially. Let us consider a 3DTV system displaying simultaneously two views, and views are coded with the coding structure as illustrated in Figure 7.

In order to decode a picture in view 1 at any temporal instant, the picture in view 0 at the same temporal instant will be decoded first. The only way to display two views at the same time is by having an MVC decoder running two times faster than a regular single-view decoder. Even though two independent decoders running on different platforms might be available, both decoders need to run twice faster than the single-view decoder because decoding has to be performed sequentially. The situation gets worse, with the increasing number of views that is supported by the 3D display. Currently there are commercial displays which can display 100 views simultaneously, and if all the views depend on each other, then the decoder must run 100 times faster, which is very challenging.

One way to increase the parallelism is to code each view independently. However, this kind of simulcast approach results in a significant penalty in coding efficiency as interview redundancies are not exploited at all. The draft MVC standard includes a more efficient method that allows parallel decoding/encoding operation of multiple views with high coding efficiency. This is achieved by utilizing the parallel decoding information SEI message that indicates that the views are encoded with systematic constraints, so that any macroblock in a certain view is allowed to depend only on reconstruction values of a subset of macroblocks in other views [37, 38].

In order to describe how parallel processing is achieved using parallel decoding information SEI message, let us consider an example, where two pictures from view 1 and view 0 are going to be decoded. Assume view 1 picture references view 0 picture as illustrated in Figure 8 (for simplicity, the sizes of the frames are five macroblocks both horizontally and vertically). Parallel decoding information SEI message indicates the video is encoded in a way that macroblocks in view 1 picture could only use reconstruction values of macroblocks that belong to certain rows in view 0 picture. For example, the macroblocks in the first macroblock row of view 1 picture could only use reconstruction values from the first two macroblock rows in view 0 picture. In other words, the available reference area for the first macroblock row of view 1 picture constitutes only data from the first

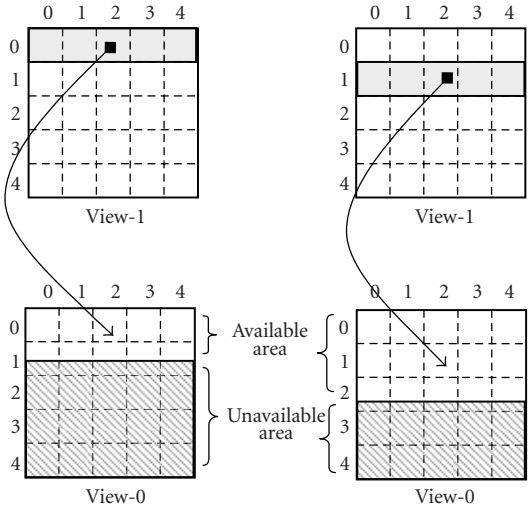


Figure 8: Systematic restriction of reference area.

two macroblock rows of view 0 picture (i.e., the motion vectors for the view 1 macroblocks are restricted). Similarly, the second macroblock row of view 1 picture only uses reconstruction values of the first three macroblock rows of the view 0 picture. This systematic restriction of reference area enables parallel decoding of first row of view 1 with any row below the second of view 0, as they are not referring each other.

In order to illustrate how this feature is used, let us assume an MVC decoder running on two processors (or processor cores) and decoding a bitstream containing two-views, where view 1 references view 0. Further assume that the bitstreams are coded with the restrictions as described above. The parallel decoding operation of these two views is illustrated in Figure 9, where processor P0 is decoding view 0 pictures and processor P1 is decoding view 1 pictures. The decoding operations for both views start simultaneously, but decodingf the first row of macroblocks in view 1 picture does not start before view 0 notifies the view 1 decoder. This notification is done after all the macroblocks in the first two macroblock rows in view 0 are decoded, and their reconstruction data are placed in the memory. This notification tells decoder of view 1 that all data required to decode first macroblock row in view 1 are ready. This way, the decoder of view 1 could start decoding the macroblocks of the first row, while the decoder of view 0 proceeds with decoding macroblocks in the third row and two decoders run in parallel. This parallel operation continues with two macroblock rows of delay between two views till the decoding of all the macroblocks is finished.

The benefit of using parallel decoding information SEI message is that significant coding gain is achieved over simulcast, while maintaining almost the same desirable parallelism characteristics. When compared to anchor method, where encoding happens without utilizing the SEI message and systematic restrictions, it is seen that parallel operation is achieved with almost no penalty on coding efficiency:

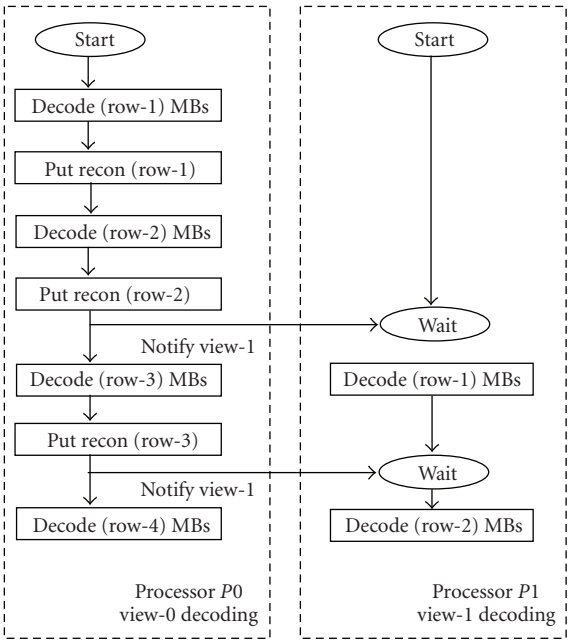


Figure 9: Sample parallel decoding process for two views.

maximum 0.08 and in average 0.03 dB loss for all the test sequences defined in the comment test condition for MVC [39]. Compared to simulcast, similar parallelism is achieved with 0.9 dB gain on coding efficiency as interview prediction is still utilized [37, 38].

In addition to using this SEI message, parallel processing could also be achieved by using simpler prediction structures. For example, consider the case where nonanchor pictures do not use interview prediction but only temporal prediction [40]. This approach achieves parallel operation as nonanchor pictures can be independently decoded without referencing other views. The parallelism of this structure could be further improved by using the parallel decoding information SEI message for anchor pictures.

It should be noted that parallel decoding information SEI message does not change the worst case complexity of MVC decoders. This means that the MVC decoders need to be designed to handle bitstreams where the encoding restrictions have not been applied, and parallel decoding information SEI message is not present. However, system standards such as DVB-H [41] can mandate the usage of this SEI message on their respective environments. This would ensure parallel operation for all the decoders operating on these services.

7. Other Related Techniques

Beside those discussed in earlier sections, the joint draft of MVC includes the following related techniques as summarized below.

7.1. Reference Picture List Construction. The reference picture list construction process can flexibly arrange temporal and view prediction references. This provides not only potential coding efficiency gain but also error resilience, since reference picture section and redundant picture mechanisms can then be extended to the view dimension [15]. This strengthens the error robustness of the MVC bitstreams.

7.2. Active View Information SEI Message. The decoder may prefer to display a subset of the views encoded in an MVC bitstream. If this preference can be known by the decoder, then only the output views and the dependent views need to be decoded and stored in the DPB. The active view information SEI message was introduced to indicate the views that are to be output [42].

7.3. Multiview Scene Information and Multiview Acquisition Information SEI Messages. Two SEI messages related to acquisition and rendering were introduced in MVC, namely multiview scene information SEI message and multiview acquisition information SEI message, to signal camera parameters, which are helpful in view interpolation by a renderer [43].

8. Concluding Remarks

In this paper, we reviewed the key aspects of the system, transport interface, and decoder designs of MVC. We also introduced techniques crucial in meeting the requirements of typical 3D services and system architectures. These solutions, as adopted to the draft MVC standard, focus on two parts: features to facilitate storage and transport of MVC bitstreams and features to achieve minimum decoder resource consumption.

For the MVC system and transport interface, bandwidth adaptation, decoder capability adaptation, view random access, and view switching are the main concerns of the design. For the MVC decoder, minimizing the memory consumption and computational complexities are addressed.

The following key points of the MVC design are highlighted.

(i) MVC shared the same network abstraction layer (NAL) unit types designed in SVC, while differs a little in some specific syntax elements.

(ii) The base view of an MVC bitstream was designed to be H.264/AVC compatible in a way that it can be reconstructed by a standard H.264/AVC decoder. At the same time, backward-compatible extensions allow to utilize MVC-specific features with an MVC-compliant decoder.

(iii) New supplemental enhancement information (SEI) messages have been introduced to signal operation points as well as their dependency information. It is designed for adaptation and bitstream extraction. In addition, a mechanism has been specified that allows reusing all the original H.264/AVC SEI messages.

(iv) Time-first coding order was introduced to facilitate the file format design of MVC. This coding order is essential to achieve optimal buffer management at the decoder.

(v) Parallel decoding information SEI message was introduced to enable parallel encoder/decoder operation

for different views. This is especially important for 3D broadcast systems that support head-motion parallax, where the receiving end needs to decode and display multiple views simultaneously.

Acknowledgment

This work was supported in part by Nokia and the Academy of Finland, Finnish Centre of Excellence Program 2006-2011 under Project 213462.

References

[1] A. Smolic, H. Kimata, and A. Vetro, "Development of MPEG standards for 3D and free viewpoint video," in *Three-Dimensional TV, Video, and Display IV*, vol. 6016 of *Proceedings of SPIE*, Boston, Mass, USA, October 2005.

[2] H.-Y. Shum, S. B. Kang, and S.-C. Chan, "Survey of image-based representations and compression techniques," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 11, pp. 1020–1037, 2003.

[3] C. Fehn, "Depth-image-based rendering (DIBR), compression and transmission for a new approach on 3D-TV," in *Stereoscopic Displays and Virtual Reality Systems XI*, vol. 5291 of *Proceedings of SPIE*, pp. 93–104, San Jose, Calif, USA, May 2004.

[4] H. Kimata, M. Kitahara, K. Kamikura, Y. Yashima, T. Fujii, and M. Tanimoto, "System design of free viewpoint video communication," in *Proceedings of the 4th International Conference on Computer and Information Technology (CIT '04)*, pp. 52–59, Wuhan, China, September 2004.

[5] A. Smolic and P. Kauff, "Interactive 3-D video representation and coding technologies," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 98–110, 2005.

[6] A. Vetro, W. Matusik, H. Pfister, and J. Xin, "Coding approaches for end-to-end 3D TV systems," in *Proceedings of the 23rd Picture Coding Symposium (PCS '04)*, pp. 319–324, San Francisco, Calif, USA, December 2004.

[7] C. Fehn, R. de la Barré, and S. Pastoor, "Interactive 3-DTV-concepts and key technologies," *Proceedings of the IEEE*, vol. 94, no. 3, pp. 524–538, 2006.

[8] J. G. Eden, "Information display early in the 21st century: overview of selected emissive display technologies," *Proceedings of the IEEE*, vol. 94, no. 3, pp. 567–574, 2006.

[9] B. Fröba and C. Küblbeck, "Face detection and tracking using edge orientation information," in *Visual Communications and Image Processing*, vol. 4310 of *Proceedings of SPIE*, pp. 583–594, San Jose, Calif, USA, January 2001.

[10] L. Young and D. Sheena, "Methods & designs: survey of eye movement recording methods," *Behavior Research Methods & Instrumentation*, vol. 7, no. 5, pp. 397–429, 1975.

[11] ITU-T Rec. H.264—ISO/IEC IS 14496-10, "Advanced video coding for generic audiovisual services," v3, 2005.

[12] ISO/IEC JTC1/SC29/WG11, "Requirements on multi-view video coding v.5," N7539, Nice, France, October 2005.

[13] "Joint draft 6.0 on multi-view video coding," JVT-Z209, Antalya, Turkey, January 2007.

[14] D. Tian, M. M. Hannuksela, and M. Gabbouj, "Sub-sequence video coding for improved temporal scalability," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '05)*, vol. 6, pp. 6074–6077, Kobe, Japan, May 2005.

[15] Y. Chen, Y.-K. Wang, and M. M. Hannuksela, "On MVC reference picture list construction," JVT-V043, Marrakech, Morocco, January 2007.

[16] T. Wiegand, G. Sullivan, J. Reichel, H. Schwarz, and M. Wien, Eds., "Joint draft 11 of SVC amendment," JVT-X201, Geneva, Switzerland, June-July 2007.

[17] Y.-K. Wang, M. M. Hannuksela, S. Pateux, A. Eleftheriadis, and S. Wenger, "System and transport interface of SVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1149–1163, 2007.

[18] S. Wenger, "H.264/AVC over IP," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 645–656, 2003.

[19] Y. Chen, Y.-K. Wang, and M. M. Hannuksela, "Comments on MVC JD 2.0," JVT-W035, San Jose, Calif, USA, April 2007.

[20] Y. Chen, Y.-K. Wang, and M. M. Hannuksela, "View scalability information SEI message for MVC," JVT-W037, San Jose, Calif, USA, April 2007.

[21] Y. Chen, Y.-K. Wang, and M. M. Hannuksela, "MVC comments on JD 3.0," Geneva, Switzerland, July 2007.

[22] M. M. Hannuksela, Y.-K. Wang, and M. Gabbouj, "Isolated regions in video coding," *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 259–267, 2004.

[23] M. Karczewicz and R. Kurceren, "The SP- and SI-frames design for H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 637–644, 2003.

[24] Y. Chen, Y.-K. Wang, and M. Gabbouj, "Buffer requirement analyses for multi-view video coding," in *Proceedings of the 26th Picture Coding Symposium (PCS '07)*, Lisbon, Portugal, November 2007.

[25] "Joint multiview video model (JMVM) 1.0," JVT-T208, Klagenfurt, Austria, July 2006.

[26] ISO/IEC IS 14496-2, "Information technology—coding of audio-visual objects—part 12: ISO base media file format," 2005.

[27] Y.-K. Wang, Y. Chen, and M. M. Hannuksela, "Time-first coding for multi-view video coding," JVT-U104, Hangzhou, China, October 2006.

[28] Y. Chen, Y.-K. Wang, and M. M. Hannuksela, "On MVC file format," M14634, Lausanne, Switzerland, July 2007. 1pt0.8pt

[29] H. Schwarz, D. Marpe, and T. Wiegand, "Analysis of hierarchical B pictures and MCTF," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '06)*, pp. 1929–1932, Toronto, Ontario, Canada, July 2006.

[30] Y. Chen, Y.-K. Wang, and M. M. Hannuksela, "MVC reference picture management," JVT-U105, Hangzhou, China, October 2006.

[31] A. Vetro and S. Yea, "Comments on MVC reference picture marking," JVT-U062, Hangzhou, China, October 2006.

[32] P. Pandit, Y. Su, P. Yin, and C. Gomila, "Comments on high-level syntax for MVC," JVT-U026, Hangzhou, China, October 2006.

[33] Y. Chen, Y.-K. Wang, M. M. Hannuksela, S. Liu, and H. Li, "On MVC reference picture marking," JVT-V044, Marrakech, Morocco, January 2007.

[34] A. Vetro and S. Yea, "MVC clarification of marking process," JVT-V085, Marrakech, Morocco, January 2007.

[35] Y.-K. Wang, Y. Chen, and M. M. Hannuksela, "Comments to JMVM 1.0," JVT-U103, Hangzhou, China, October 2006.

[36] J. Choi, W. Shim, H. Song, and Y. Moon, "Inter-view prediction reference picture marking," JVT-W056, San Jose, Calif, USA, April 2007.

[37] K. Ugur, H. Liu, J. Lainema, M. Gabbouj, and H. Li, "Parallel encoding-decoding operation for multi-view video coding with high coding efficiency," in *Proceedings of the Conference on True Vision, Capture, Transmission, and Display of 3D Video (3DTV '07)*, pp. 1–4, Kos Island, Greece, May 2007.

[38] K. Ugur, J. Lainema, H. Liu, and Y.-K. Wang, "Parallel decoding info SEI message for MVC," JVT-V098, Marrakech, Morocco, January 2007.

[39] "Common test conditions for multiview video coding," JVT-T207, Klagenfurt, Austria, July 2006.

[40] P. Merkle, A. Smolic, K. Mueller, and T. Wiegand, "Comparative study of MVC structures," JVT-V132, Marrakech, Morocco, January 2007.

[41] G. Faria, J. A. Henriksson, E. Stare, and P. Talmola, "DVB-H: digital broadcast services to handheld devices," *Proceedings of the IEEE*, vol. 94, no. 1, pp. 194–209, 2006.

[42] Y.-K. Wang, M. M. Hannuksela, and Y. Chen, "MVC output related conformance," JVT-W036, San Jose, Calif, USA, April 2007.

[43] A. Vetro, S. Yea, W. Matusik, H. Pfister, and M. Zwicker, "Anti-aliasing for 3D displays," JVT-W060, San Jose, Calif, USA, April 2007.

[P4] Y. Guo, Y. Chen, Y.-K. Wang, H. Li, M. M. Hannuksela, and M. Gabbouj, "Error Resilient Coding and Error Concealment in Scalable Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 6, pp. 781–795, June 2009.

Error Resilient Coding and Error Concealment in Scalable Video Coding

Yi Guo, Ying Chen, *Member, IEEE*, Ye-Kui Wang, *Member, IEEE*, Houqiang Li, Miska M. Hannuksela, *Member, IEEE*, and Moncef Gabbouj, *Senior Member, IEEE*

***Abstract*— Scalable video coding (SVC), which is the scalable extension of the H.264/AVC standard, was developed by the Joint Video Team (JVT) of ISO/IEC MPEG (Moving Picture Experts Group) and ITU-T VCEG (Video Coding Experts Group). SVC is designed to provide adaptation capability for heterogeneous network structures and different receiving devices with the help of temporal, spatial, and quality scalabilities. It is challenging to achieve graceful quality degradation in an error-prone environment, since channel errors can drastically deteriorate the quality of the video. Error resilient coding and error concealment techniques have been introduced into SVC to reduce the quality degradation impact of transmission errors. Some of the techniques are inherited from or applicable also to H.264/AVC, while some of them take advantage of the SVC coding structure and coding tools. In this paper, the error resilient coding and error concealment tools in SVC are first reviewed. Then, several important tools such as loss-aware rate-distortion optimized macroblock mode decision algorithm and error concealment methods in SVC are discussed and experimental results are provided to show the benefits from them. The results demonstrate that PSNR gains can be achieved for the conventional inter prediction (IPPP) coding structure or the hierarchical bi-predictive (B) picture coding structure with large group of pictures size, for all the tested sequences and under various combinations of packet loss rates, compared with the basic Joint Scalable Video Model (JSVM) design applying no error resilient tools at the encoder and only picture copy error concealment method at the decoder.**

***Index Terms*— Error concealment, error resilient coding, H.264/AVC, SVC.**

I. INTRODUCTION

SCALABLE VIDEO coding, also referred to as layered video coding, has been designed to facilitate video services using a single bit stream, from which appropriate sub-bit stream can be extracted to meet different preferences and requirements for a possibly large number of end users, over heterogeneous network structures with a wide range of quality of service (QoS). In scalable video coding (SVC), a video is coded into more than one layer: the base layer and enhancement layers, the latter of which usually can improve user experience with respect to picture rate, spatial resolution, and/or video quality. These enhancements are referred to as temporal, spatial, and SNR scalabilities, respectively, and can be used in a combined manner.

Manuscript received March 29, 2008; revised June 21, 2008. First version published March 16, 2009; current version published June 19, 2009. This paper is partially supported by National Natural Science Foundation of China (NSFC) General Program (Contract No. 60572067&60672161), and NSFC Key Program (Contract No. 60736043). It is also supported in part by Nokia and the Academy of Finland, Finnish Center of Excellence Program 2006-2011 under Project 213462. This paper was recommended by Associate Editor H. Sun.

Y. Guo and H. Li are with the Department of Electronic Engineering and Information Science at the University of Science and Technology of China, Hefei (e-mail: guoyi@mail.ustc.edu.cn; lihq@ustc.edu.cn).

Y. Chen and M. Gabbouj are with the Department of Signal Processing at Tampere University of Technology, Tampere, Finland (e-mail: ying.chen@tut.fi; moncef.gabbouj@tut.fi).

Y.-K. Wang and M. M. Hannuksela are with the Nokia Research Center, Tampere, Finland (e-mail: ye-kui.wang@nokia.com; miska.hannuksela@nokia.com).

Digital Object Identifier 10.1109/TCSVT.2009.2017311

A. Scalable Video Coding Over Heterogeneous Networks

Typical application scenarios for SVC are shown in Fig. 1. Note that, in this figure, only spatial and temporal scalabilities are shown. However, the scenarios for spatial scalability are also valid for SNR scalability. In practice, those scenarios may exist in different systems with different contents, network structures, and receiving devices.

Due to various levels of decoding capability, videos with different spatial resolutions, e.g., for a standard definition TV (SDTV) set and a high definition TV (HDTV) set, can be decoded as shown in scenario (a), or videos with different picture rates, e.g., for a mobile device and a laptop, can be decoded as shown in scenario (b).

The clients can be the same but within different sub-networks or with different connections, e.g. in scenario (c). The clients are connected with cable, local area network (LAN), digital subscriber line (DSL), and wireless LAN (WLAN). Clients can also be located in the same network but with different QoS, e.g., the different congestion control methods applied by the intermediate nodes. Therefore, the expected bandwidth for each client may be different, which will lead to various received videos combined with different picture rates, spatial resolutions, and/or quality levels.

Even for one client, owing to bandwidth fluctuation, the received video may change at any moment in picture rate, spatial resolution, and quality level.

B. Error Robust Requirement and Error Control

The number of packet-based video transmission channels, such as the Internet and packet-oriented wireless networks, has been increasing rapidly. One inherent problem of video transmitted in packet-oriented transport protocol is channel errors, as client 4 in scenario (c) of Fig. 1. Packet loss may be caused if a packet fails to reach the destination in a specific time. Another source of packet loss is bit errors caused by physical interference in any link of the transmission

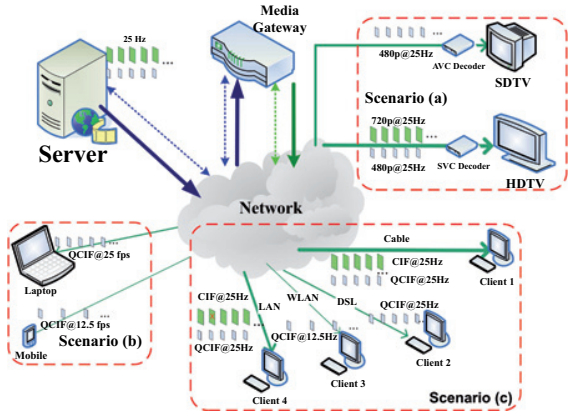


Fig. 1. Scalable video coding application scenarios.

path. Many video communication systems apply the user datagram protocol (UDP) [1]. Any bit error occurring in a UDP packet will result in the loss of the packet, as UDP discards packets with bit errors. Packet loss can damage one whole picture or an area of it. Unfortunately, because of the predictive coding techniques, a transmission error (after error concealment) will propagate both temporally and spatially, and sometimes can bring substantial deterioration to the subjective and objective quality of the reproduced video sequence until an instantaneous decoding refresh (IDR) picture. However, if the bit stream is protected by error control methods [2], the system may still maintain graceful degradation.

Various error control methods have been proposed. In [3], error control methods are classified into four types as follows: transport-level error control; source-level error resilient coding; interactive error control; and error concealment.

This paper will mainly focus on source-level error resilient coding and error concealment. Error resilient coding injects such redundancy into the bit stream, which helps receivers in recovery or concealment from potential channel errors. The objective of error resilient coding is to design a scheme that can achieve the minimum end-to-end distortion under a certain rate. The redundancy may be used to detect data losses, stop error propagation, and/or guide error concealment. Error concealment provides an estimation of lost picture areas based on the correctly decoded samples as well as any other helpful information. Error concealment is done only by the decoder, unlike other methods that require encoder actions.

C. Outline and Contribution of This Paper

In this paper, error resilient coding and error concealment techniques used in single-layer coding are reviewed first. Some of these techniques are included in or can be applied to SVC [4], which is the scalable extension of H.264/AVC [5]. Several new error resilient techniques in SVC, including some normative tools as well as the non-normative loss-aware rate-distortion optimized mode decision (LA-RDO) algorithm, are then discussed. Furthermore, error concealment algorithms, which are designed according to new characteristics of SVC, e.g., inter-layer texture, motion and residual prediction, are discussed. It is shown that techniques based on the inter-layer correlation can outperform the techniques inherited from single-layer coding, only based on the spatial/temporal correlations.

The rest of this paper is organized as follows. First, an overview of SVC is given in Section II in order to help understand the discussion of the error resilient coding and error concealment tools. In Section III, techniques for single-layer coding, especially for H.264/AVC, are introduced. The error resilient coding and error concealment tools, most of which were proposed by the authors of this paper, are discussed in Section IV. Simulation results are provided in Section V to show the benefits of the proposed algorithms. Finally, Section VI concludes the paper.

II. Overview of the Scalable Extension of H.264/AVC

This section reviews SVC (the scalable extension of H.264/AVC), which is important to understand the terminologies required for SVC error resilient coding and error concealment. SVC has been included in MPEG-2 video (also known as ITU-T H.262) [6], H.263 [7], MPEG-4 visual [8], and SVC, which all provide temporal, spatial, and SNR scalabilities.

A. Novel Features of SVC

Some functionalities of SVC are inherited from H.264/AVC. Compared to previous scalable standards, the most essential advantages, namely hierarchical temporal scalability, inter-layer prediction, single-loop decoding, and flexible transport interface, are reviewed below.

According to the SVC specification, the pictures with the lowest spatial and quality layer are compatible with H.264/AVC, and their pictures of the lowest temporal level form the temporal base layer, which can be enhanced with pictures of higher temporal levels. In addition to the H.264/AVC-compatible layer, several spatial and/or SNR enhancement layers can be added to provide spatial and/or quality scalabilities. SNR scalability is also referred to as quality scalability. Each spatial or SNR enhancement layer itself may be temporally scalable, with the same temporal scalability structure as the H.264/AVC-compatible layer. For one spatial or SNR enhancement layer, the lower layer it depends on is also referred to as the base layer of that specific spatial or SNR enhancement layer. In this paper, unless otherwise stated, the term "base layer" refers to a certain spatial or SNR layer, the information (texture, residue, and motion) of which may be used as inter-layer prediction by a higher spatial or SNR layer, and the term "enhancement layer" refers to the specific higher spatial or SNR layer.

1) Hierarchical Temporal Scalability: H.264/AVC provides a flexible hierarchical B picture coding structure, which enables it to realize advanced temporal scalability [9]. With this feature inherited from H.264/AVC, SVC supports temporal scalability for layers with different resolutions [10]. In SVC, a group of pictures (GOP) consists of a so-called key picture and all pictures that are located in output/display order between

this key picture and the previous key picture. A key picture is coded in regular or irregular intervals, which is either intra-coded or inter-coded using the previous key picture as reference for motion-compensated prediction. The non-key pictures are hierarchically predicted from the pictures with lower temporal levels. The temporal level of a picture is indicated by the syntax element temporal_id in the network abstraction layer (NAL) unit header SVC extension [4].

2) Inter-layer Prediction: SVC introduces inter-layer prediction for spatial and SNR scalabilities based on texture, residue, and motion. The spatial scalability in SVC has been generalized into any resolution ratio between two layers [10]. The SNR scalability can be realized by coarse granularity scalability (CGS) or medium granularity scalability (MGS) [10]. In SVC, two spatial or CGS layers belong to different dependency layers (indicated by dependency_id in NAL unit header [4]), while two MGS layers can be in the same dependency layer. One dependency layer includes quality layers with quality_id [4] from zero to higher values, which correspond to quality enhancement layers. In SVC, inter-layer prediction methods are utilized to reduce the inter-layer redundancy. They are briefly introduced in the following paragraphs.

1) Inter-layer texture prediction: The coding mode using inter-layer texture prediction is called "IntraBL" mode in SVC. To enable single-loop decoding [11], only the macroblocks (MBs) whose co-located MBs in the base layer are constrainedly intra-coded can use this mode. A constrainedly intra-coded macroblock (MB) is intra-coded without referring to any samples from the neighboring MBs that are inter-coded.
2) Inter-layer residual prediction: If an MB is indicated to use residual prediction, the co-located MB in the base layer for inter-layer prediction must be an inter MB and its residue may be upsampled according to the resolution ratio. The difference between the residue of the enhancement layer and that of the base layer is coded.
3) Inter-layer motion prediction: The co-located base layer motion vectors may be scaled to generate predictors for the motion vectors of MB or MB partition in the enhancement layer. In addition, there is one MB type named base mode, which sends one flag for each MB. If this flag is true and the corresponding base layer MB is not intra, then motion vectors, partitioning modes and reference indices are all derived from base layer.

3) Single-loop Decoding: The single-loop decoding scheme in SVC is revolutionary compared to earlier scalable coding techniques. In the single-loop decoding scheme, only the target layer needs to be motion-compensated and fully decoded [11]. Therefore, compared to the conventional multiple-loop decoding scheme, where motion compensation and full decoding are typically required for every spatial or SNR-scalable layer, decoding complexity as well as the decoded picture buffer (DPB) size can be greatly reduced.

4) Flexible Transport Interface: SVC provides flexible systems and transport interface designs that enable seamless integration of the codec to scalable multimedia application systems. Other than compression and scalability provisioning, systems and transport interface focuses on codec functionalities, such as, for video codec in general, interoperability and conformance, extensibility, random access, timing, buffer management, as well as error resilience, and for scalable coding in particular, backward compatibility, scalability information provisioning, and scalability adaptation. These mechanisms are augmented by the SVC file format extension to the International Standardization Organization (ISO) Base Media File Format [12] and Real-time Transport Protocol (RTP) payload formats [13]. Discussions of these SVC systems and transport interface designs can be found in [12], [13], and [14]. The error resilient coding and error concealment tools that are applicable to SVC are discussed in the following sections of this paper.

III. Overview of Error Resilient Coding and Error Concealment Tools for H.264/AVC

Earlier video coding standards (H.261/3, MPEG-1/2/4) support the following standard error resilient coding tools: 1) intra MB/picture refresh [15]; 2) slice coding [15]; 3) reference picture identification (see below); 4) reference picture selection (RPS) [15]; 5) data partitioning [15]; 6) header extension code and header repetition [15]; 7) spare picture signaling [16]; 8) intra block motion signaling [17]; 9) reversible variable length coding (RVLC) [15]; 10) resynchronization marker [15]; 11) source-coding-level FEC [18]; and redundant pictures (also known as sync pictures for video redundancy coding) [19].

Seven of the above tools, namely intra MB/picture refresh, slice coding, reference picture identification, RPS, data partitioning, spare picture signaling, and redundant slices/pictures, are also supported by H.264/AVC. In addition to the "old" standard tools, H.264/AVC includes some new standard tools: 1) parameter sets [20]; 2) Flexible MB Order (FMO) [20]; 3) Gradual Decoding Refresh (GDR) [21]; 4) scene information signaling [22]; 5) SP/SI pictures [23]; 6) constrained intra prediction (see below); and 7) reference picture marking repetition (RPMR, see below).

Nonstandard error control tools include error concealment [15], error tracking [24], [25], and multiple description coding (MDC) [26]. Basically, all the nonstandard tools can be used with any video codec, including H.264/AVC and SVC. However, only a subset of MDC methods, e.g., the one reported in [27], generates standard-compatible bit streams.

Among all the above-mentioned standard error resilient coding tools, reference picture identification, spare picture signaling, GDR, scene information signaling, constrained intra prediction, and RPMR have not been covered by the earlier review papers in [2], [15], [20], [23], and are supported by H.264/AVC or SVC. These tools are reviewed in the following section. In addition, intra refresh and redundant slices/pictures are also reviewed, as the former is the basis for the discussion of SVC LA-RDO algorithm in Section IV, and for the latter there have been considerable amount of new developments since the old review in [20]. For nonstandard error control tools, only error concealment is reviewed, to form the basis for the discussions of SVC error concealment methods in Section IV. Readers are referred to the corresponding references

listed above for those error resilient tools that are not covered by the following reviews and detailed discussions.

A. Standard Error Resilient Coding Tools in H.264/AVC

In this section, the standard error resilient coding tools in H.264/AVC are summarized.

1) Reference Picture Identification: In H.264/AVC, each reference pictures is with an incremental frame number. This design frame number enables decoders to detect loss of reference pictures and take proper actions when there are losses of reference pictures.

2) Gradual Decoding Refresh (GDR): GDR is enabled by the so-called isolated region technique [21]. An isolated region evolving over time can completely stop error propagation resulting from packet losses occurring before the starting point of the isolated region in a gradual manner, i.e., after the isolated region covers the entire picture area. It can also be used for other purposes such as gradual random access.

3) Redundant Slices/Pictures: Various usages of redundant slices/pictures are proposed in [27]–[29]. Furthermore, H.264/AVC-compatible redundant picture coding in combination with RPS, reference picture list reordering (RPLR), and adaptive redundant picture allocation was reported in [30].

4) Reference Picture Marking Repetition (RPMR): RPMR, using the decoded reference picture marking repetition SEI message, can be used to repeat the decoded reference picture marking syntax structures in the earlier decoded pictures. Consequently, even if earlier reference pictures were lost, the decoder can still maintain correct status of the reference picture buffer and reference picture lists.

5) Spare Picture Signaling: The spare picture SEI message, which signals the similarity between a reference picture and other pictures, tells the decoder which picture can be used as a substituted reference picture or can be used to better conceal the lost reference picture [16].

6) Scene Information Signaling: The scene information SEI message provides a mechanism to select a proper error concealment method for intra pictures, scene-cut pictures, and gradual scene transition pictures at the decoder [22].

7) Constrained Intra Prediction: In the constrained intra prediction mode, samples from inter coded blocks are not used for intra prediction. Consequently, temporal error propagation can be efficiently stopped.

8) Intra MB/Picture Refresh: Intra refresh intentionally inserts intra pictures or intra MBs into the bit stream. It can achieve better RD performance on certain packet loss conditions. Several methods for insertion of intra MBs have been reported, e.g., random intra refresh (RIR) [31], cyclic intra refresh (CIR) [32], recursive optimal per-pixel estimate (ROPE) [33], sub-pixels ROPE [34], LA-RDO algorithm in H.264/AVC [35], and 4×4 block-based error propagation map method [36].

B. Error Concealment for H.264/AVC

Error concealment is a decoder-only technique. Typically, the spatial, temporal, and spectral redundancy can be made use of to mask the effect of channel errors at the decoder.

If the picture is partially corrupted, e.g., the picture is split into multiple slices, spatial error concealment method, e.g., as in [37], can be used. For low bit rate video transmission such as 3G wireless systems, usually one picture is coded into only one packet, and loss of a packet implies that the entire picture must be recovered from the previously decoded pictures. The simplest way to solve this problem is by copying the previously decoded picture to replace the lost one. However, if the sequence is with smooth motion, motion copy [38] can be used to improve the performance.

IV. Error Resilient Coding and Error Concealment Tools for SVC

All the standard error resilient video coding tools supported by H.264/AVC are inherited to SVC. However, data partitioning and SP/SI pictures are not included in the currently specified SVC profiles. All the nonstandard error control tools are supported by SVC, in the same manner as H.264/AVC. Some of these tools that are inherited from H.264/AVC are supported in the SVC reference software, namely the joint scalable video model (JSVM). These tools are briefly summarized in Section IV-A.

Besides the tools inherited from H.264/AVC, SVC includes three new standard error resilient coding tools, namely quality layer integrity check signaling, redundant picture property signaling, and temporal level zero index signaling. These tools are discussed in Section IV-B.

The conventional error resilient coding and error concealment tools for single-layer coding can certainly be applied to the SVC enhancement layers. However, these methods do not utilize the correlations between different layers, which are high in many cases. Improved performance can be expected if inter-layer correlations are utilized. In Sections IV-C and IV-D, we discuss LA-RDO-based intra MB refresh and error concealment algorithms, respectively, that utilize inter-layer correlations in SVC bit streams.

A. Error Control Tools Inherited from H.264/AVC and Supported in the JSVM

The JSVM software include the support of FMO [39], redundant pictures [40], [41], slice coding [42], LA-RDO-based intra MB refresh [43], as well as some error concealment methods [44], [45].

The simplest exact-copy redundant coding for each picture was proposed to the JSVM by [40]. An unequal error protection (UEP) like method, which only codes redundant representations for key pictures of enhancement layers, was proposed in [41]. The LA-RDO-based intra MB refresh algorithm, which was proposed in [43], was extended from the single-layer method reported in [36]. Four error concealment methods were proposed in [44] according to the inter-layer prediction characteristics of SVC. Another improved error concealment method using motion copy for key picture was proposed in [45]. It has also been agreed to include it in the JSVM software, but at the time of writing the feature has not yet been integrated. By applying some of these error concealment methods in a combined manner, significant PSNR gain compared to single layer error concealment algorithms can be observed.

B. New Standard Error Resilient Coding Tools in SVC

1) Quality Layer Integrity Check Signaling: The quality layer integrity check SEI message includes a cyclic redundancy check (CRC) code calculated from all the quality enhancement NAL units (with the syntax element quality_id larger than 0) of a dependency representation (all NAL units in one access unit and with the same value for the syntax element depencency_id). This information can be used to verify whether all quality NAL units of a dependency representation are received by the decoder. If loss is detected, the decoder can inform the loss to the encoder, which in turn decides the use of the error-free base quality layer as reference for encoding subsequent access units. Therefore the drift error by using the erroneous highest quality layer as reference can be avoided. When no loss is detected, the encoder is free to use the highest quality layer as reference for improved coding efficiency. More details can be found in [46].

2) Redundant Picture Property Signaling: The redundant picture property SEI message can be used to indicate the correlations between a redundant layer representation and the corresponding primary layer representation. A layer representation consists of all NAL units in one dependency representation and with the same value for the syntax element quality_id. Indicated information includes, when a primary picture is lost, whether redundant representation can completely replace the primary representation:

1) for inter prediction or inter-layer prediction;
2) for inter-layer mode prediction (part of inter-layer motion prediction);
3) for inter-layer motion prediction;
4) for inter-layer residual prediction;
5) for inter-layer texture prediction.

More details can be found in [41].

3) Temporal Level Zero Index Signaling: The temporal level zero dependency representation index SEI message provides a mechanism to detect whether a dependency representation at the lowest temporal level (i.e., with temporal_id equal to 0) needed for decoding the current access unit is available when NAL unit losses are expected during transport. Decoders can use the SEI message to determine whether to transmit a feedback message or a retransmission request concerning a lost dependency representation at the lowest temporal level. More details can be found in [47]–[49].

C. LA-RDO-Based Intra MB Refresh for SVC

In SVC, when encoding an MB in an enhancement layer picture, the traditional MB coding modes in single-layer coding as well as new inter-layer prediction mode can be used. Similar as in single-layer coding, MB mode selection in SVC also affects the error resilient performance of the encoded bit stream. In the following, a method that is extended from the single-layer method in [36] to multilayer coding is presented. In this method, given the target packet loss rate (PLR), the 4×4 block-based error propagation maps for a picture is calculated, and the map is taken into account to perform mode decision for pictures in the latter.

In order to understand the multilayer method better, we first discuss the generic LA-RDO process and the particular single-layer method in [36].

1) Mode Decision: The MB mode selection is decided according to the following steps.

1) Loop over all the candidate modes, and for each candidate mode, estimate the distortion of the reconstructed MB resulting from both packet losses and source coding, and the coding rate (e.g., the number of bits for representing the MB).
2) Calculate each mode's cost, which is represented by the following equation, and choose the mode that gives the smallest cost

$$C = D + \lambda R. \tag{1}$$

In (1), C denotes the cost, D denotes the estimated distortion, R denotes the estimated coding rate, and λ is the Lagrange multiplier.

2) Single-layer Method: Assume that the PLR is p_l. The overall distortion of the mth MB in the nth picture with the candidate coding option o is represented by

$$\begin{aligned} D(n,m,o) = (1-p_l)(D_s(n,m,o) + D_{ep\_ref}(n,m,o)) \\ + p_l D_{ec}(n,m) \end{aligned} \tag{2}$$

where $D_s(n,m,o)$ and $D_{ep\_ref}(n,m,o)$ denote the source coding distortion and the error propagation distortion, respectively; and $D_{ec}(n,m)$ denotes the error concealment distortion in case the MB is lost. Obviously, $D_{ec}(n,m)$ is independent of the MBs coding mode. The source coding distortion $D_s(n,m,o)$ is the distortion between the original signal and the error-free reconstructed signal.

Source coding distortion $D_s(n,m,o)$ is the distortion between the original signal and the error-free reconstructed signal. It can be calculated as the mean square error (MSE), sum of absolute difference (SAD), or sum of square error (SSE). The error concealment distortion $D_{ec}(n,m)$ can be calculated as the MSE, SAD, or SSE between the original signal and the error concealed signal. The used norm, i.e., MSE, SAD or SSE, shall be aligned for $D_s(n,m,o)$ and $D_{ec}(n,m)$.

For the calculation of the error propagation distortion $D_{ep\_ref}(n,m,o)$, a distortion map D_{ep} for each picture on a block basis (e.g., 4×4 luminance samples) is defined. Given the distortion map, $D_{ep\_ref}(n,m,o)$ is calculated as

$$\begin{aligned} D_{ep\_ref}(n,m,o) &= \sum_{k=1}^{K} D_{ep\_ref}(n,m,k,o) \\ &= \sum_{k=1}^{K} \sum_{l=1}^{4} w_l D_{ep}(n_l, m_l, k_l, o) \end{aligned} \tag{3}$$

where K is the number of blocks in one MB, and $D_{ep\_ref}(n,m,k,o)$ denotes the error propagation distortion of the kth block in the current MB. $D_{ep\_ref}(n,m,k,o)$ is calculated as the weighted average of the error propagation distortion $\{D_{ep}(n_l, m_l, k_l, o_l)\}$ of the blocks $\{k_l\}$ that are referenced by the current block. The weight w_l of each reference block is proportional to the area that is used for reference.

The distortion map with the optimal coding mode o^* is defined as follows.

For an inter-coded block wherein bi-prediction is not used, i.e., there is only one reference picture used

$$D_{ep}(n,m,k) = (1-p_l)D_{ep\_ref}(n,m,k,o^*) + p_l(D_{ec\_rec}(n,m,k,o^*) + D_{ec\_ep}(n,m,k)) \quad (4)$$

where $D_{ec\_rec}(n,m,k,o^*)$ is the distortion between the error-concealed block and the reconstructed block, and $D_{ec\_ep}(n,m,k)$ is the distortion due to error concealment and the error propagation distortion in the reference picture that is used for error concealment. Equation (3) is used to calculate $D_{ec\_ep}(n,m,k)$ assuming that the error concealment method is known, i.e., $D_{ec\_ep}(n,m,k)$ is calculated as the weighted average of the error propagation distortion of the blocks that are used for concealing the current block, and the weight w_l of each reference block is proportional to the area that is used for error concealment.

For an inter-coded block wherein bi-prediction is used, i.e., there are two reference pictures used

$$\begin{aligned} D_{ep}(n,m,k) = {} & w_{r0} \times ((1-p_l)D_{ep\_ref\_r0}(n,m,k,o^*) \\ & + p_l(D_{ec\_rec}(n,m,k,o^*) + D_{ec\_ep}(n,m,k))) \\ & + w_{r1} \times ((1-p_l)D_{ep\_ref\_r1}(n,m,k,o^*) \\ & + p_l(D_{ec\_rec}(n,m,k,o^*) + D_{ec\_ep}(n,m,k))) \end{aligned} \quad (5)$$

where w_{r0} and w_{r1} are, respectively, the weights of the two reference pictures used for bi-prediction.

For an intra-coded block, no error propagation distortion is transmitted, and only error concealment distortion is considered

$$D_{ep}(n,m,k) = p_l(D_{ec\_rec}(n,m,k,o^*) + D_{ec\_ep}(n,m,k)) \quad (6)$$

According to [50] the error-free Lagrange multiplier is represented by

$$\lambda_{ef} = -\frac{dD_s}{dR}. \quad (7)$$

However, when transmission error exists, a different Lagrange multiplier may be needed.

Combining (1) and (2), we get

$$C = (1-p_l)(D_s(n,m,o) + D_{ep\_ref}(n,m,o)) + p_l D_{ec}(n,m) + \lambda R. \quad (8)$$

Let the derivative of C to R be zero, and we get

$$\lambda = -(1-p_l)\frac{dD_s(n,m,o)}{dR} = (1-p_l)\lambda_{ef}. \quad (9)$$

Consequently, (1) becomes

$$C = (1-p_l)(D_s(n,m,o) + D_{ef\_ref}(n,m,o)) + p_l D_{ec}(n,m) + (1-p_l)\lambda_{ef} R. \quad (10)$$

Since $D_{ec}(n,m)$ is independent of the coding mode, it can be removed. After $D_{ec}(n,m)$ is removed, the common coefficient $(1-p_l)$ can also be removed, which finally results in

$$C = D_s(n,m,o) + D_{ep\_ref}(n,m,o) + \lambda_{ef} R. \quad (11)$$

3) Multilayer Method: In scalable coding with multiple layers, the MB mode decision for the base layer pictures is exactly the same as in the single-layer method. For a slice in an enhancement layer picture, if no inter-layer prediction is used, the single-layer method is used, with the used PLR being the PLR of the current layer. Otherwise (if inter-layer prediction is used), the distortion estimation and the Lagrange multiplier selection processes are presented below.

Let the current layer contain the current MB be l_c, the lower layer contain the co-located MB used for inter-layer prediction by the current MB be $l_{c\text{-}1}$, the further lower layer containing the MB used for inter-layer prediction of the co-located MB in $l_{c\text{-}1}$ be $l_{c\text{-}2}, \ldots$, and the lowest layer containing an inter-layer-dependent block for the current MB be l_0, and let the PLRs be $p_{l,c}, p_{l,c\text{-}1}, \ldots, p_{l,0}$, respectively. For a current slice that may use inter-layer prediction, it is assumed that a contained MB would be decoded only if the MB and all the dependent lower-layer blocks are received; otherwise the slice is concealed. For a slice that does not use inter-layer prediction, a contained MB would be decoded as long as it is received.

The overall distortion of the mth MB in the nth picture in layer l_c with the candidate coding option o is represented by

$$D(n,m,o) = \left(\prod_{i=0}^{c}(1-p_{l,i})\right)(D_s(n,m,o) + D_{ep\_ref}(n,m,o)) + \left(1-\prod_{i=0}^{c}(1-p_{l,i})\right)D_{ec}(n,m) \quad (12)$$

where $D_s(n,m,o)$ is calculated the same way as in the single-layer method. $D_{ec}(n,m)$ is determined by the chosen error concealment method. Given the distortion map of the reference picture in the same layer or in the lower layer (for inter-layer texture prediction), $D_{ep\_ref}(n,m,o)$ is calculated using (3).

The distortion map is derived as presented in below. When the current layer is of a higher spatial resolution, the distortion map of the lower layer $l_{c\text{-}1}$ is first upsampled. For example, if the resolution is changed by a factor of two for both the width and the height, then each value in the distortion map is simply upsampled to be a 2×2 block of identical values.

1) Texture prediction: In this mode, distortion can be propagated from the lower layer. Then the distortion map of the kth block in the current MB is as in (13). Note that here $D_{ep\_ref}(n,m,k,o^*)$ is the distortion map of the kth block in the co-located MB in the lower layer l_{n-1}. $D_{ec\_rec}(n,m,k,o^*)$ and $D_{ec\_ep}(n,m,k)$ are calculated the same as in the single-layer method

$$D_{ep}(n,m,k) = \left(\prod_{i=0}^{c}(1-p_{l,i})\right)D_{ep\_ref}(n,m,k,o^*) + \left(1-\prod_{i=0}^{c}(1-p_{l,i})\right) \times (D_{ec\_rec}(n,m,k,o^*) + D_{ec\_ep}(n,m,k)). \quad (13)$$

2) Motion prediction: Since the motion prediction in JSVM use the motion vector field, reference indices and MB partitioning of the lower layer are for the corresponding MB in the current layer. The inter prediction process still uses the reference pictures in the same layer. For a block that uses inter-layer motion prediction and does not use bi-prediction, the distortion map of the kth block is

$$D_{ep}(n,m,k) = \left(\prod_{i=0}^{c}(1-p_{l,i})\right) D_{ep\_ref}(n,m,k,o^*) + \left(1-\prod_{i=0}^{c}(1-p_{l,i})\right)(D_{ec\_rec}(n,m,k,o^*) + D_{ec\_ep}(n,m,k)). \quad (14)$$

For a block that uses inter-layer motion prediction and also uses bi-prediction, the distortion map of the kth block is

$$\begin{aligned} &D_{ep}(n,m,k) \\ &= w_{r0}\times\left(\left(\prod_{i=0}^{c}(1-p_{l,i})\right) D_{ep\_ref\_r0}(n,m,k,o^*) \right. \\ &+\left(1-\prod_{i=0}^{c}(1-p_{l,i})\right)(D_{ec\_rec}(n,m,k,o^*) \\ &+ D_{ec\_ep}(n,m,k))) + w_{r1}\times\left(\left(\prod_{i=0}^{c}(1-p_{l,i})\right)\right. \\ &\times D_{ep\_ref\_r1}(n,m,k,o^*) + \left(1-\prod_{i=0}^{c}(1-p_{l,i})\right) \\ &\left.\left.\times (D_{ec\_rec}(n,m,k,o^*) + D_{ec\_ep}(n,m,k))\right)\right). \end{aligned} \quad (15)$$

Note that here $D_{ep\_ref}(n,m,k,o^*)$ in (14) and $D_{ep\_ref\_r0}(n,m,k,o^*)$ and $D_{ep\_ref\_r1}(n,m,k,o^*)$ in (15) are the distortion map of the kth block calculated from reference pictures in the same layer. $D_{ep\_ec}(n,m,k,o^*)$ and $D_{ec\_ep}(n,m,k,o^*)$ are calculated the same as in the single-layer method.

1) Residual prediction: If the low layer is received, and residue of the low layer can be decoded correctly, then there is no error propagation. Otherwise, the error concealment is performed. Therefore, (14) and (15) can also be used to derive the distortion map for an MB mode using inter-layer residual prediction.
2) No inter-layer prediction: For an inter-coded block, (14) and (15) are used to generate the distortion map, while for an intra-coded block

$$D_{ep}(n,m,k) = \left(1-\prod_{i=0}^{c}(1-p_{l,i})\right) \times (D_{ec\_rec}(n,m,k,o^*) + D_{ec\_ep}(n,m,k)). \quad (16)$$

The calculation process of $D_{ep}(n,m,k)$ can be seen from Fig. 2 clearly.

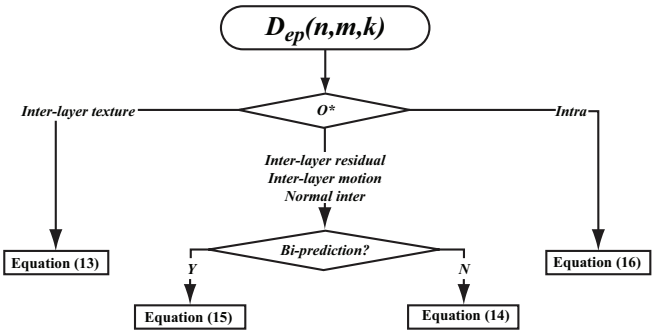


Fig. 2. Calculation of the distortion map $D_{epc}(n,m,k)$.

Combining (1) and (12), we get

$$C = \left(\prod_{i=0}^{c}(1-p_{l,i})\right)(D_s(n,m,o) + D_{ep\_ref}(n,m,o)) \times \left(1-\prod_{i=0}^{c}(1-p_{l,i})\right) D_{ec}(n,m) + \lambda R. \quad (17)$$

Let the derivative of C to R be zero, and then we get

$$\lambda = -\left(\prod_{i=0}^{c}(1-p_{l,i})\right)\left(\frac{dD_s(n,m,o)}{dR}\right) = \left(\prod_{i=0}^{c}(1-p_{l,i})\right)\lambda_{\mathrm{ef}}. \quad (18)$$

Consequently, (1) becomes

$$C = \left(\prod_{i=0}^{c}(1-p_{l,i})\right)(D_s(n,m,o) + D_{ep\_ref}(n,m,o)) \times \left(1-\prod_{i=0}^{c}(1-p_{l,i})\right) D_{ec}(n,m) + \left(\prod_{i=0}^{c}(1-p_{l,i})\right)\lambda_{\mathrm{ef}} R. \quad (19)$$

Here, $D_{ec}(n,m)$ may be dependent on the coding mode, since the MB may be concealed even it is received, while the decoder may utilize the known coding mode to use a better error concealment method. Therefore, the $D_{ec}(n,m)$ term should be retained. Consequently, the coefficient $\prod_{i=0}^{c}(1-p_{l,i})$ that is not common for all the items should also be retained. The final mode decision process becomes

$$C = D_s(n,m,o) + D_{ep\_ref}(n,m,o) + \lambda_{ef} R. \quad (20)$$

Note that the difference between (20) and (11) is that $D_{ep\_ref}(n,m)$ may come from the base layer distortion map if the checked mode o is inter-layer texture prediction and base layer MB is reconstructed. The mode decision process for multilayer is depicted in Fig. 3.

D. Error Concealment Algorithms for SVC

1) Reference Picture Management for Lost Pictures: Upon detection of a lost picture, a key picture is concealed as a lost P picture, and the necessary RPLR commands and

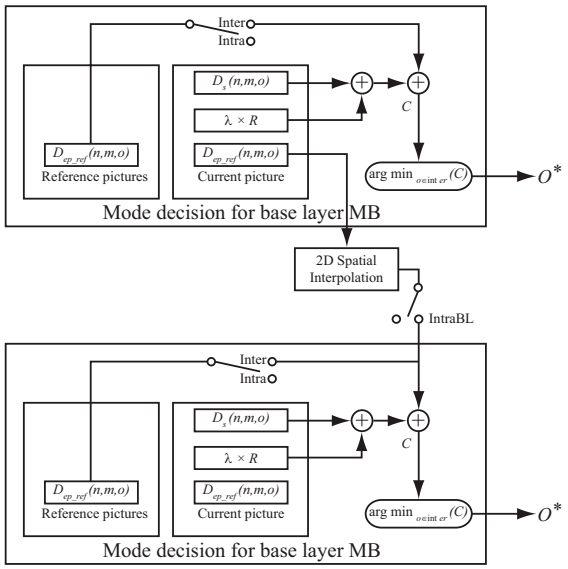


Fig. 3. Mode decision algorithm for the multilayer method.

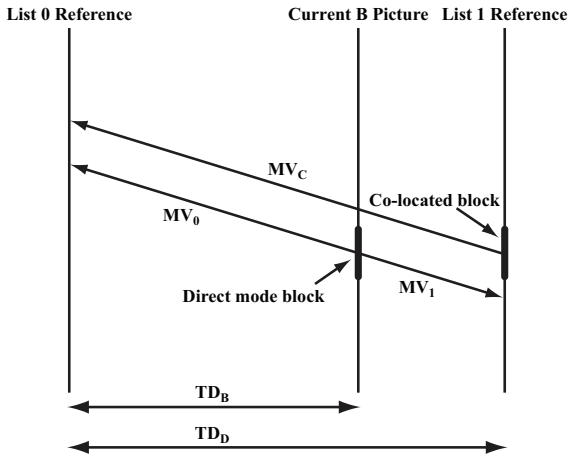


Fig. 4. Example for temporal direct-mode motion vector inference.

memory management control operation (MMCO) commands are set as follows. The RPLR commands are to guarantee the current picture to be predicted from the previous key picture. The MMCO commands are to mark the unnecessary decoded pictures in the previous GOP so as to guarantee the minimum DPB even when packet losses occur. How to conceal a lost key picture is to be discussed in the following sections.

If a lost picture is not a key picture, usually the RPLR commands can be constructed based on those of the pictures in the previous GOPs or on those of the base layer picture if the lost picture is in the enhancement layer.

On the basis of the current design of SVC, the corresponding enhancement layer picture will not be decodable if the base layer picture is lost unless two layers are independently encoded. So base layer picture loss leads to the "loss" of the whole access unit, and one picture of a certain layer leads to the "loss" of the pictures in all the higher layers of the same access unit.

Two types of error concealment algorithms are implemented by us in the current JSVM software. They are summarized as intra-layer error concealment and inter-layer error concealment. One of those methods, if used, is applied to the whole picture, although it is possible that different MBs can selectively use different methods.

2) Intra-layer Error Concealment Algorithms: Intra-layer error concealment is defined as the method that uses the information of the same spatial or quality layer to conceal a lost picture. Three methods are introduced.

1) Picture copy (PC): In this algorithm, each pixel value of the concealed picture is copied from the corresponding pixel of the first picture in the reference picture list 0. If multiple-loop decoding is supported for an error concealment method, this algorithm can be invoked for both the base layer and enhancement layers. Otherwise, only the highest layer in the current access unit can be used for concealment.
2) Temporal direct (TD) for B pictures: The TD mode specified in H.264/AVC is generated as follows. As can be seen in Fig. 4, we assume that an MB or MB partition in the current B picture is coded in temporal direct mode, and then its motion vectors are inferred from its neighboring reference pictures. If the co-located MB or MB partition (belongs to List 1 Reference as shown in Fig. 4) in the reference picture list (namely list for simplicity) one uses a picture (named in Fig. 4 as List 0 Reference) as a reference in list 0 and that picture is also in the list 0 of the current B picture, then the List 0 Reference and List 1 Reference are chosen to bi-predict the being processed MB or MB partition of the current picture. The list 0 and list 1 motion vectors MV_0 and MV_1 are scaled from MV_c using the picture order count (POC, i.e., display order) differences. The detailed deriving process can be seen in [51].

The temporal direct mode specified in H.264/AVC standards cannot be used for any spatial or SNR enhancement layer. However, the concealment of the B picture in SVC can still be applicable for both base layer and enhancement layer. Using the calculated MVs including list 0 and list 1 motion vectors, motion compensation from two specific reference pictures is utilized to predict the MB in the lost picture, assuming zero residue.

In the current SVC design, the necessary motion vectors are stored for each layer. This makes it possible to apply TD at the decoder without introducing extra memory requirement.

1) Motion copy (MC) for key pictures**:** The MC algorithm is applicable for the lost key pictures. Key pictures are concealed as P pictures no mater whether they are originally I or P pictures, since TD is not applicable for this picture and PC may not be efficient because the gap of two key pictures may be large (depending on the GOP size). To get a more accurately concealed picture for the lost key picture, motion vectors are re-generated by copying the motion field of the previous key picture.

3) Inter-layer Error Concealment Algorithms: Two methods are introduced: one works for single-loop decoding; and the other works for multiple-loop decoding.

1) Base layer skip (BLSkip): This method operates as follows. If the base layer is an intra MB, then texture prediction is used. If the base layer is an inter MB, then motion prediction as well as residual prediction are used to generate information for an MB in a lost picture at the enhancement layer. In this case, motion compensation is done at the enhancement layer using the possibly upsampled motion vectors. This algorithm can directly be used for the enhancement layer if there is no picture loss in the base layer. If base layer picture is also lost, the motion vectors for base layer picture are generated using the TD method first. We call this method as BLSkip+TD, but for simplicity we will use BLSkip to represent this method throughout this paper.
2) Reconstruction base layer and possibly upsampling (RU): In the RU algorithm, the base layer picture is reconstructed, and may be upsampled, for the lost picture at the enhancement layer, which is dependent on the spatial ratio between the enhancement layer and the base layer. This requires full decoding of a base layer and thus leads to the requirement of multiple-loop decoding. This method is helpful when there are continuous picture losses only in the enhancement layer and may be competitive for low motion sequences compared with BLSkip.

4) The Improved Error Concealment Algorithm: The improved error concealment method which combines BLSkip with MC is proposed. MC is used to repair the loss of the base layer key picture or those key pictures of the enhancement layer whose base layer pictures are lost. Meanwhile BLSkip is used for the other pictures with losses.

The applicability of these methods is as follows. PC works for all pictures; TD works for all non-key pictures; BLSkip and RU work only for enhancement layer pictures; MC work for key pictures. The RU method can be only used when the decoder adopts multiloop decoding.

V. Simulation

A. Test Conditions

To demonstrate performance of the proposed algorithms, the Bus, Football, Foreman, and News sequences (YUV 4:2:0, 30 frames/s, and progressive) were tested. The tested sequences can be categorized according to their motion characteristics. Bus sequence has high but very regular motion; Foreman sequence has medium but irregular motion; Football sequence has high and irregular motion, while the News sequence has slow motion. The simulation conditions are as follows.

1) JSVM 9.7.
2) Low delay application (IPPP coding structure) and high delay application (hierarchical B picture coding structure with GOP size equal to 16) were tested separately.
3) 4001 pictures were encoded and decoded.
4) Intra picture period: 32 for low delay application and 128 for high delay application.
5) Two layers: base layer was QCIF@30 Hz; enhancement layer was CIF@30 Hz.
6) QP: 28, 32, 36, 40. Base layer and enhancement layer had the same QPs.
7) Multiple slice structure was not used.
8) The error patterns included in [52] are used, and PLRs were as in the following table:

TABLE I
Tested PLRs

Base layer PLR (%)	0	3	3	5	5	10	10	20
Enhancement layer PLR (%)	3	3	5	5	10	10	20	20

The PLR pair at the encoder for LA-RDO was the same as that of the target PLR pair of the decoder.

The bit stream through packet loss was generated by [53] with two modifications as follows.

1) The base layer was defined as the spatial base layer.
2) Error patterns that determine the packet losses of enhancement layer and base layer packets do not overlap.

The comparisons in the following aspects are considered:

1) with/without LA-RDO;
2) with/without MC.

As it can be concluded from the experimental results of [44] that the BLSkip error concealment method is a good error concealment tool and PC method is preliminary, both of them are considered here as basic algorithms for comparisons. RU requires multiple-loop decoding, thus the results are not reported here but can be found in [44].

Given different choices, there are various combinations in terms of configurations. However, each of them is compared with the PC case without LA-RDO, which is named as "Anchor" in this section, and the Y-PSNR (luma) differences are calculated by the Bjontegaard measurement [54].

B. Simulation Results for Low-Delay Application

The results are shown in Fig. 5, and we could see that the BLSkip method outperforms the PC method for all tested sequences, with an average PSNR gain of around 2.3 dB over all sequences and all PLR pairs, as summarized in Fig. 6. A further 1 dB gain on average can be achieved by MC, as shown in Fig. 6. LA-RDO provides nearly 5 dB gain on average when PC is utilized. If LA-RDO and BLSkip+MC are combined, an average of more than 5.5 dB gain can be obtained, which outperforms any other methods. However, there may be a few losses in regard to several low PLR pairs compared with "Anchor," which may be caused by some excessive intra MBs introduced by LA-RDO algorithm.

It is also clear that, for low motion sequences (e.g., News), the gains between PC and other advanced error concealment methods are relatively small no matter whether LA-RDO is on or off. Furthermore, the benefits when LA-RDO is off are far from those when LA-RDO is on for the low motion sequence, as shown in Fig. 6.

The gains of the above methods, especially when the best configuration is adopted, increase when the PLR pair increases. However, when the PLR pair is very high, e.g.,

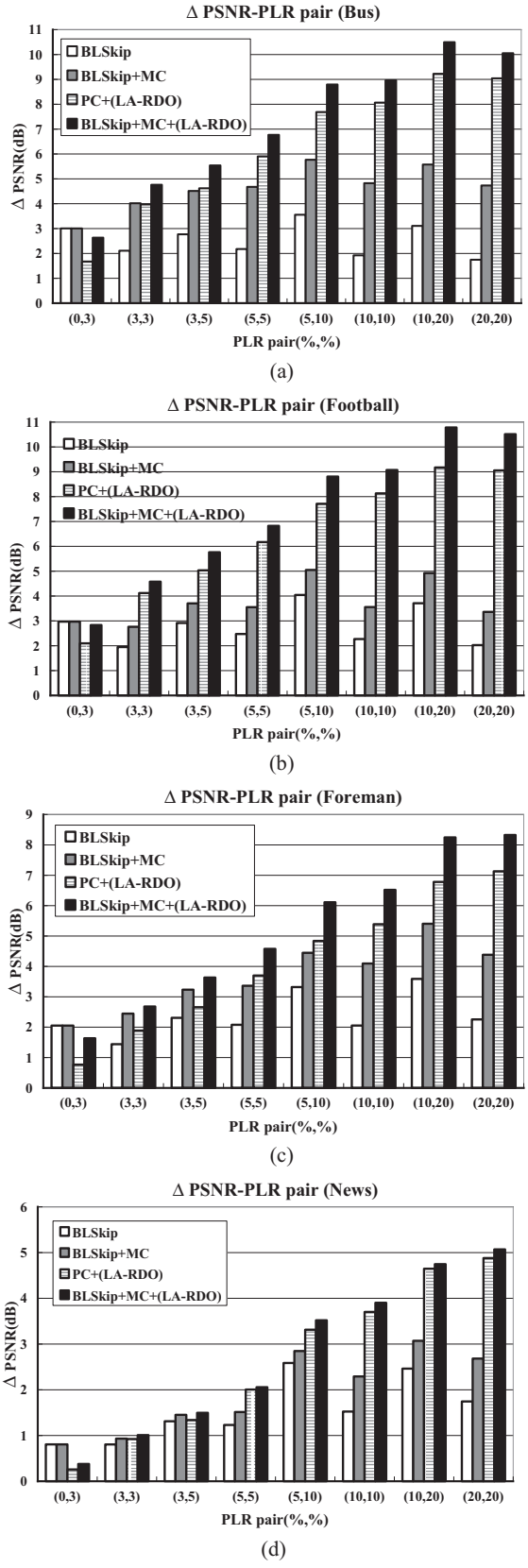


Fig. 5. △ PSNR (dB) for low delay application.

(20%, 20%), the gains might decrease a little for the high motion sequences (i.e., Bus and Football).

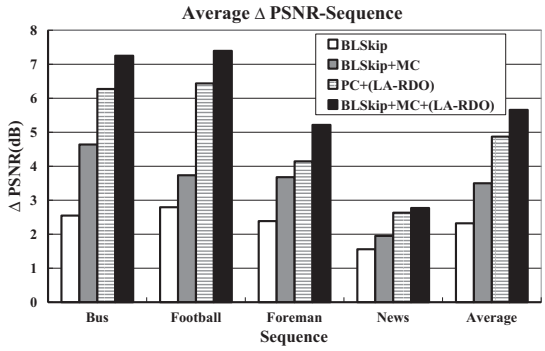


Fig. 6. Average △ PSNR(dB) for low delay application.

Some selective RD curves are plotted in Fig. 7 in order to show the error resilient performance of different methods clearly. It should be noted that, in these figures, if LA-RDO is on, the bit rate will increase much more than in other methods by reason of increasing intra MBs under the same QP setting, so the bit rate ranges for curves with LA-RDO and those for curves without LA-RDO are different. However, there are still some overlapped bit rate ranges, and the trends of these curves are obvious; therefore it is easy to determine which one is the best among different curves.

As can be seen, BLSkip+MC with LA-RDO is the best method among all the methods, while BLSkip+MC is the best when LA-RDO is off.

C. Simulation Results for High-Delay Application

The results are shown in Fig. 8, and the average values are given in Fig. 9. Compared with low-delay application, the results of TD method in high-delay application were provided by extra bars.

From these two figures, we can see that the BLSkip method outperforms the PC method, with an average PSNR gain of around 2.8 dB over all sequences and all PLR pairs. But only a small gain on average can be achieved by MC. LA-RDO provides smaller gains than those of low-delay application, while the average gain is about 2.8 dB when PC is utilized.

Also, the average gain provided by LA-RDO decreases to around 4.0 dB when BLSkip+MC is adopted compared to 5.5 dB in low-delay application. TD method outperforms PC by only about 0.3 dB on average, which is much worse than BLSkip. In conclusion, inter-layer information is of crucial importance for the error concealment algorithm in SVC and is much better than making use of only intralayer information.

Compared to the results shown in Fig. 5 and Fig. 6, most of the observations are still valid, and we skip the detailed analysis of them in this section. However, the differences of the performances in high-delay application are discussed.

The most significant difference is that the average gain for the BLSkip method is higher in high-delay application (about 2.8 dB) than in the low-delay application (about 2.3 dB). The main reason is that in the high-delay application, hierarchical B picture coding structure is used, and therefore the distances from pictures are farther and motion information turns out to be more important. In this case also, PC turns out worse

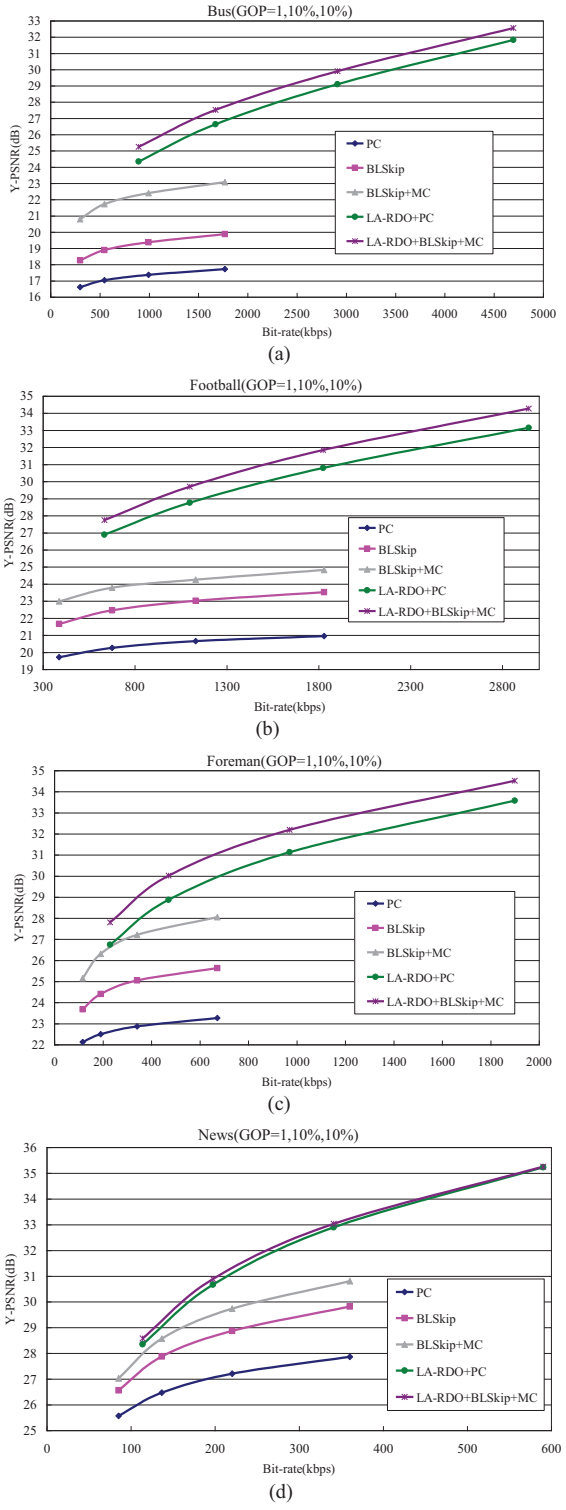


Fig. 7. RD curves of all sequences for the (10%, 10%) PLR pair in low delay application.

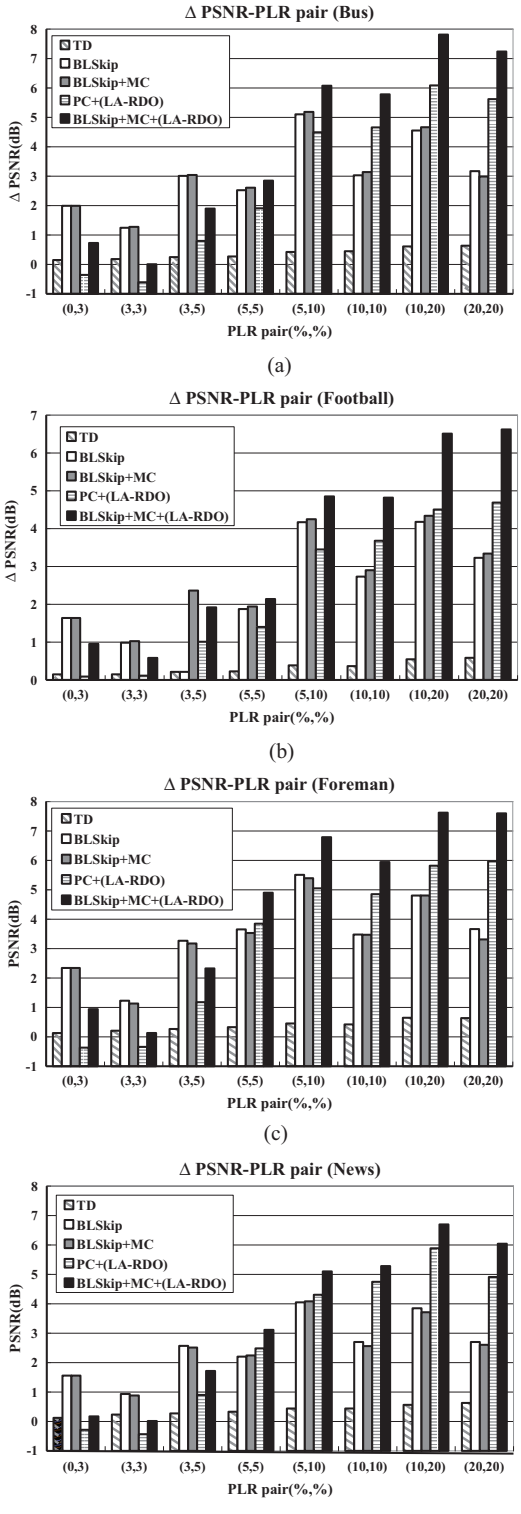


Fig. 8. △ PSNR(dB) for high-delay application.

because those pictures can be used for copying with a larger distance to the lost picture. Temporal motion prediction gets weaker because of the same reason. But this does not affect inter-layer motion prediction used in BLSkip. However, the average gain of Football decreases to about 2.3 dB, which demonstrates that the utilization of inter-layer information will be less effective in the high-delay application for some fast and irregular motion sequences.

The second difference is that the MC method performs much worse, mainly because there is only one key picture in every 16 pictures, and the motion information copied form last

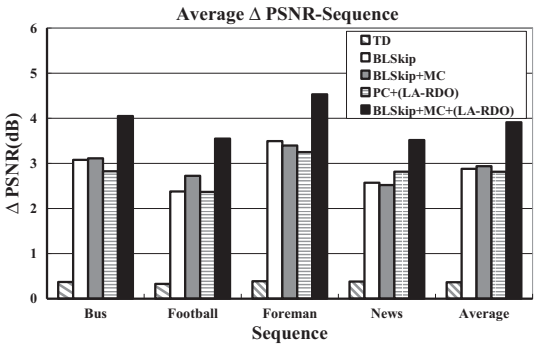


Fig. 9. Average △ PSNR(dB) for high-delay application.

key picture for current key picture will be futile if these two pictures are in different motion speeds and directions. While in low-delay application, every picture is a key picture, and the correlation of motion information between two consecutive pictures is very strong, so considerable gains can be achieved compared with PC method.

The third difference is that the performance of LA-RDO decreases in hierarchical B picture coding structure. In high-delay application, the end-to-end distortion, especially those of B pictures, will be harder to estimate than in low-delay application, and inaccuracy of estimation can sometimes decrease the coding efficiency. In high-delay application, the channel distortion of one B picture may be referred by many other pictures that have higher temporal level, whereas in low-delay application only the latter picture will refer the distortion of the current key picture.

The fourth difference is that, for Bus and Foreman sequences, there are some slight losses which are less than 0.6 dB for several low PLR pairs in high-delay application, at the same time the corresponding gains of these low PLR pairs will be inferior to those gains without LA-RDO. It seems that when LA-RDO is on, there may be some excessive intra MBs in low PLR pairs, which greatly degrade the RD performance. However, for the fast and irregular motion sequence (i.e., Football), the excessive intra MBs can intentionally truncate the channel distortion, so there are no losses in the low PLR pairs.

In summary, BLSkip is much more important in the high-delay application; however, other methods, such as MC and LA-RDO are also helpful, the latter being able to provide about 1 dB extra average gain.

Some selective RD curves are plotted in Figs. 10 and 11 in order to show the error resilient performance of different methods clearly. As can be seen in Fig. 10, BLSkip+MC with LA-RDO is the best method among all the methods, while TD almost gives the same results as those of PC without LA-RDO. In Fig. 11, the RD curves of (3%, 3%) PLR pair is specially given to show that BLSkip or BLSkip+MC without LA-RDO can be suitable for some very low PLR pairs in the high-delay application. The gains are about 1–2 dB compared to other methods.

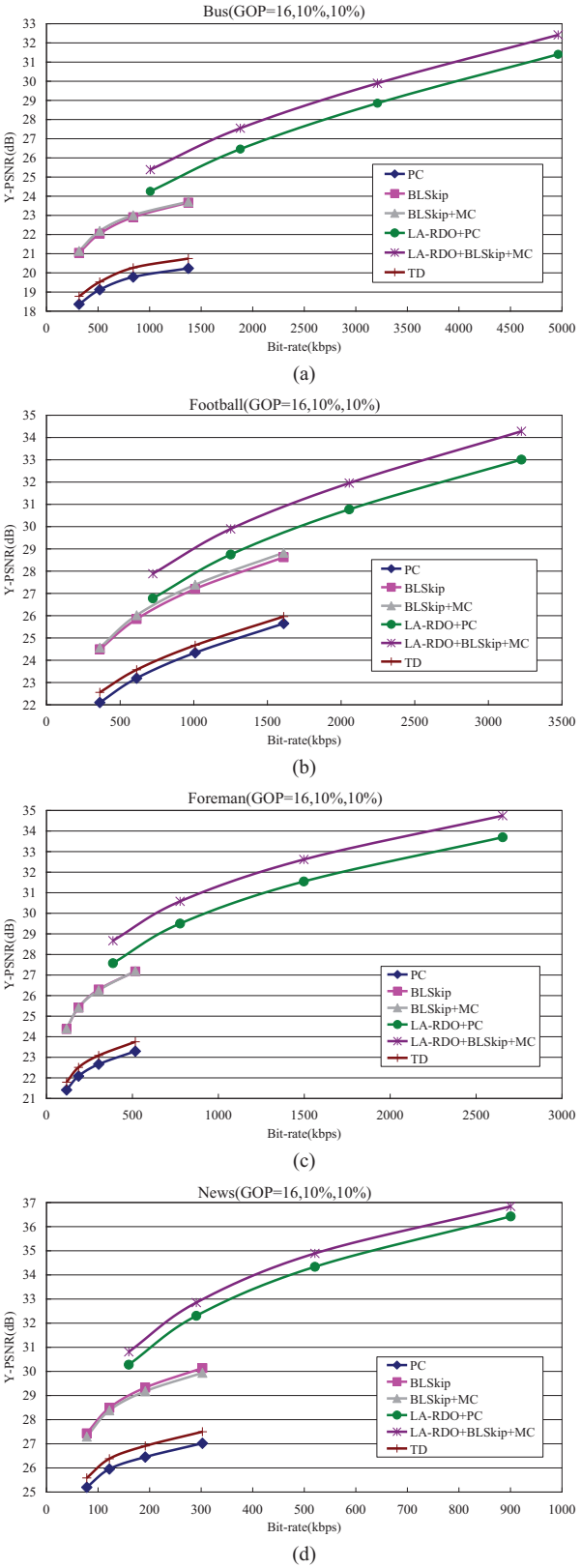


Fig. 10. RD curves of all sequences for (10%, 10%) PLR pair in high-delay application.

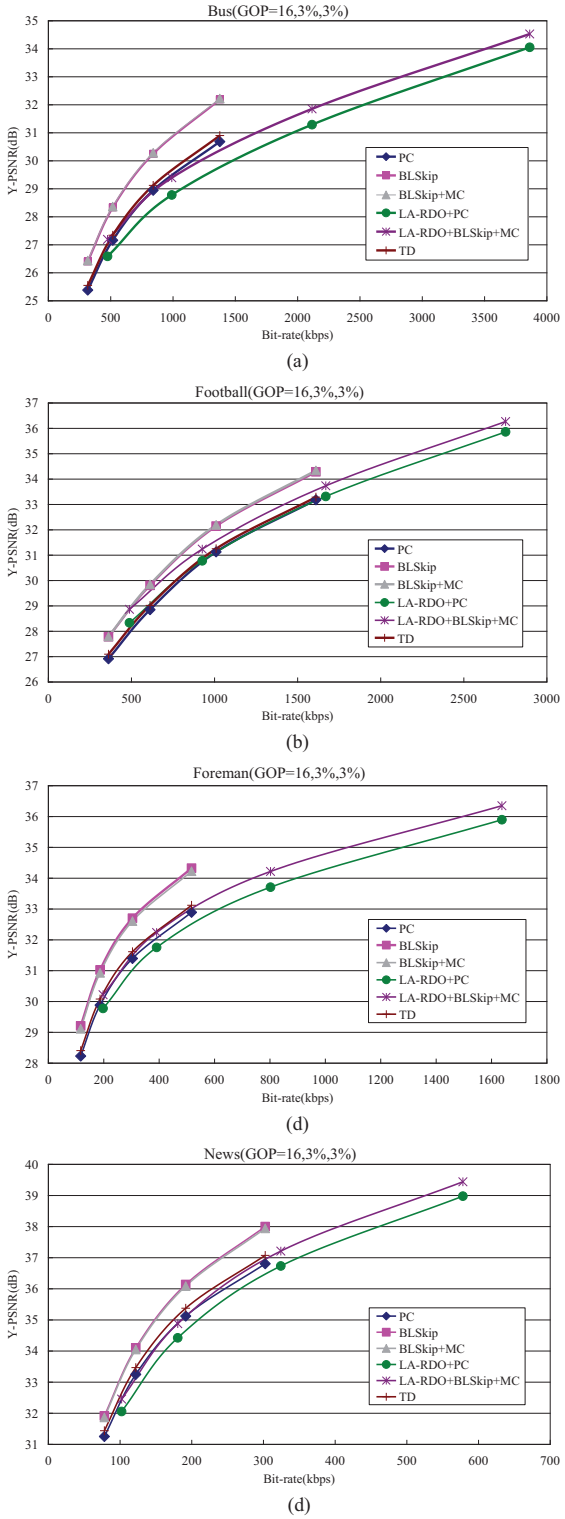


Fig. 11. RD curves of all sequences for (3%, 3%) PLR pair in high-delay application.

VI. Conclusion

SVC has been recently approved as an international standard. Apart from better coding efficiency, it provides improved adaptation capability to heterogeneous network compared to earlier SVC standards. Error resilient coding and error concealment are highly desired for the robustness and flexibility of SVC-based applications. In this paper, we reviewed error resilient coding and error concealment algorithms in H.264/AVC and SVC. LA-RDO algorithm for SVC was presented in detail. Moreover, five error concealment methods for SVC were proposed and analyzed. Simulation results showed that LA-RDO for SVC, the proposed error concealment methods, and their combination improve the average picture quality under erroneous channel conditions when compared to the design applying no error-resilient tools at the encoder and only picture copy error-concealment method at the decoder.

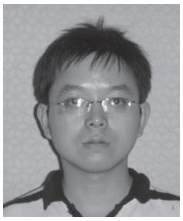
Acknowledgment

The authors thank the experts of ITU-T VCEG, ISO/IEC MPEG, and the Joint Video Team (JVT) for their contributions and Kai Xie, Jill Boyce, Purvin Pandit, and Feng Zhang from Thomson for their contributions to the SVC error concealment methods discussed in this paper.

References

[1] J. Postel, *User Datagram Protocol*, IETF STD 6 (RFC 0768), Aug. 1980.
[2] Y. Wang and Q. Zhu, "Error control and concealment for video communication: A Review," *Proc. IEEE*, vol. 86, no. 5, pp. 974–997, May 1998.
[3] Y. Wang, J. Ostermann, Y.-Q. Zhang, *Video Process. and Commun.* Englewood Cliffs, NJ: Prentice Hall, 2002.
[4] T.Wiegand, G. Sullivan, J. Reichel, H. Schwarz, and M.Wien, *Joint Draft 11 of SVC Amendment*, Joint Video Team, Doc. JVT-X201, Jun.–Jul. 2007.
[5] *Advanced Video Coding Generic Audiovisual Services*, ITU-T Rec.H.264|ISO/IEC IS 14496-10 v3, 2005.
[6] *Generic Coding Moving Pictures and Associated Audio Inform.-Part 2: Video*, ITU-T Rec.H.262|ISO/IEC 13818-2 (MPEG-2 Video), Nov. 1994.
[7] ITU-T Rec. H.263, *Video coding for low bit rate communication*, v3: Nov. 2000.
[8] ISO/IEC 14492-2 (MPEG-4 Visual), *Coding of audio-visual objects-Part 2: Visual*, v3: May 2004.
[9] D. Tian, M. M. Hannuksela, and M. Gabbouj, "Sub-sequence video coding for improved temporal scalability," in *Proc. ISCAS '05*, vol. 6. Kobe, Japan, May 2005, pp. 6074–6077.
[10] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of scalable video coding extension of H.264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.
[11] H. Schwarz, T. Hinz, D. Marpe, and T. Wiegand, "Constrained inter-layer prediction for single-loop decoding in spatial scalability," in *Proc. ICIP '05*, vol. 2. Genova, Ital, Sep. 2005, pp. II-870–II-3.
[12] P. Amon, T. Rathgen, and D. Singer, "File format for scalable video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1174–1185, Sep. 2007.
[13] S. Wenger, Y.-K. Wang, and T. Schierl, "Transport and signaling of SVC in IP networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1164–1173, Sep. 2007.
[14] Y.-K. Wang, M. M. Hannuksela, S. Pateux, A. Eleftheriadis, and S. Wenger, "System and transport interface of SVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1149–1163, Sep. 2007.
[15] Y. Wang, S. Wenger, J. Wen, and A. K. Katsaggelos, "Error resilient video coding techniques," *IEEE Signal Process. Mag.*, vol. 17, no. 4, pp. 61–82, Jul. 2000.
[16] D. Tian, M. M. Hannuksela, Y.-K. Wang, and M. Gabbouj, "Error resilient video coding techniques using spare pictures," in *Proc. Packet Video Workshop '03*, Nantes, France, Apr. 2003.
[17] S. Cen and P. Cosman, "Comparison of error concealment strategies for MPEG video," in *Proc. IEEE Wireless Commun. Networking Conf. (WCNC)*, vol. 1. New Orleans, LA, Sep. 1999, pp. 329–333.
[18] *Video Coding Low Bit Rate Commun.—Annex H: Forward Error Correction Coded Video Signal*, ITU-T Rec. H.263 Annex H, Feb. 1998.

[19] S. Wenger, "Video redundancy coding in H.263+," in *Proc. Int. Workshop Audio-Visual Services Over Packet Networks*, Sep. 1997.
[20] S. Wenger, "H.264/AVC over IP," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 645–656, Jul. 2003.
[21] M. M. Hannuksela, Y.-K. Wang, and M. Gabbouj, "Isolated regions in video coding," *IEEE Trans. Multimedia*, vol. 6, no. 2, pp. 259–267, Apr. 2004.
[22] Y.-K. Wang, M.M. Hannuksela, K. Caglar, and M. Gabbouj, "Improved error concealment using scene information," in *Proc. 2003 Intern. Workshop Very Low Bitrate Video (VLBV'03),* pp. 283–289, Madrid, Spain, Sep. 2003.
[23] S. Kumar, L. Xu, M. K. Mandal, and S. Panchanathan, "Error resiliency schemes in H.264/AVC standard," *J. Vis. Comm. Image Represent.*, vol. 17, no. 2, pp. 425–450, Apr. 2006.
[24] Y.-K. Wang, C. Zhu, and H. Li, "Error resilient video coding using flexible reference frames," in *Proc. SPIE VCIP '05*, pp. 691–702, Beijing, China, Jul. 2005.
[25] B. Girod and N. Farber, "Feedback-based error control for mobile video transmission," *Proc. IEEE*, vol. 87, no. 10, pp. 1707–1723, Oct. 1999.
[26] Y. Wang, A. R. Reibman, and S. Lin, "Multiple description coding for video delivery," *Proc. IEEE*, vol. 93, no. 1, pp. 57–70, Jan. 2005.
[27] I. Radulovic, Y.-K. Wang, S. Wenger, A. Hallapuro, M. M. Hannuksela, and P. Frossard, "Multiple description H.264 video coding with redundant pictures," in *Proc. Mobile Video Workshop, ACM Multimedia '07*, pp. 37–42, Augsburg, Germany, Sep. 2007.
[28] Y.-K. Wang, M. M. Hannuksela, and M. Gabbouj, "Error resilient video coding using unequally protected key pictures," in *Proc. 2003 Int. Workshop Very Low Bitrate Video (VLBV '03)*, pp. 290–297, Madrid, Spain, Sep. 2003.
[29] S. Rane, P. Baccichet, and B. Girod, "Modeling and optimization of a systematic lossy error protection system based on H.264/AVC redundant slices," in *Proc. Picture Coding Symp. (PCS '06)*, Beijing, China, Apr. 2006.
[30] C. Zhu, Y.-K. Wang, and H. Li, "Adaptive redundant picture for error resilient video coding," in *Proc. ICIP '07*, vol. 4. San Antonio, TX, Sep. 2007, pp. IV-253–IV-256.
[31] G. Cote and F. Kossentini, "Optimal intra coding of blocks for robust video communication over the Internet," *Signal Process. Image Commun.*, vol. 15, no. 1, pp. 25–34, Sep. 1999.
[32] Q. Zhu and L. Kerofsky, "Joint source coding, transport processing and error concealment for H.323-based packet video," in *Proc. SPIE VCIP'99*, pp. 52–62, San Jose, Jan. 1999.
[33] R. Zhang, S. L. Regunathan, and K. Rose, "Video coding with optimal inter/intra-mode switching for packet loss resilience," *IEEE J. Select. Areas Commun.*, vol. 18, no. 6, pp. 966–976, Jun. 2000.
[34] H. Yang and K. Rose, "Recursive end-to-end distortion estimation with model-based cross-correlation approximation," in *Proc. ICIP '03*, vol. 2. Barcelona, Spain, Sep. 2003, pp. III-469–III-72.
[35] T. Stockhammer, D. Kontopodis, and T. Wiegand, "Rate-distortion optimization for JVT/H.26L coding in packet loss environment," in *Proc. Packet Video Workshop '02*, Pittsburgh, PA, Apr. 2002.
[36] Y. Zhang, W. Gao, H. Sun, Q. Huang, and Y. Lu. "Error resilience video coding in H.264 encoder with potential distortion tracking," in *Proc. ICIP '04*, vol. 1. Singapore, Oct. 2004, pp. 163–166.
[37] Y.-K. Wang, M. M. Hannuksela, V. Varsa, A. Hourunranta, and M. Gabbouj, "The error concealment feature in the H.26L test model," in *Proc ICIP '02*, vol. 2. Rochester, NY, Sep. 2002, pp. II-729–II-732.
[38] Z. Wu and J. Boyce, "An error concealment scheme for entire frame losses based on H.264/AVC," in *Proc. ISCAS'06*, pp. 4463–4466, Island of Kos, Greece, May 2006.
[39] T. Bae, T. Thang, D. Kim, Y. Ro, J. Kang, J. Kim, and J. Hong, "FMO implementation in JSVM," Poznan, Porland, Doc. JVT-P043, Jul. 2005.
[40] J. Jia, H. Kim, and H. Choi etc., "Implementation of redundant pictures in JSVM," Sejong Univ. and ETRI, Doc. JVT-Q054, Nice, France, Oct. 2005.
[41] C. He, H. Liu, H. Li, Y.-K. Wang, and M.M. Hannuksela, "Redundant picture for SVC," USTC and Nokia Corporation, Doc. JVT-W049, San Jose, Apr. 2007.
[42] S. Tao, H. Liu, H. Li, and Y.-K. Wang, "SVC slice implementation to JSVM," USTC and Nokia Corporation, Doc. JVT-X046, Geneva, Switzerland, Jun. 2007.
[43] Y. Guo, Y.-K. Wang, and H. Li, "Error resilience mode decision in scalable video coding," in *Proc. ICIP'06*, pp. 2225–2228, Atlanta, Oct. 2006.
[44] Y. Chen, K. Xie, F. Zhang, P. Pandit, and J. Boyce, "Frame loss error concealment for SVC," *Journal of Zhejiang University SCIENCE A, also in Proc. Packet Video Workshop'06*, pp. 677–683, Hangzhou, China, Apr. 2006.
[45] Y. Guo, Y.-K. Wang, and H. Li, "Motion-copy error concealment for key pictures," USTC and Nokia Corporation, Doc. JVT-Y047, Shenzhen, China, Oct. 2007.
[46] Y.-K. Wang and M.M. Hannuksela, "SVC feedback based coding," Nokia Corporation, Doc. JVT-W052, San Jose, Apr. 2007.
[47] A. Eleftheriadis, S. Cipolli, and J. Lennox, "Improved error resilience using frame index in NAL header extension for SVC," Layered Media, Inc., Doc. JVT-V088, Marrakech, Morocco, Jan. 2007.
[48] Y.-K. Wang and M.M. Hannuksela, "On tl0_pic_idx in SVC," Nokia Corporation, Doc. JVT-W050, San Jose, Apr. 2007.
[49] A. Eleftheriadis, S. Cipolli, and J. Lennox, "Improved error resilience using temporal level 0 picture index," Layered Media, Inc., Doc. JVT-W062, San Jose, Apr. 2007.
[50] T. Wiegand and B. Girod, "Lagrangian multiplier selection in hybrid video coder control," in *Proc. ICIP '01*, vol. 3. Thessaloniki, Greece, Oct. 2001, pp. 542–545.
[51] M. Flierl and B. Girod, "Generalized B pictures and the draft H.264/AVC video compression standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 587–597, Jul. 2003.
[52] S. Wenger, "Error patterns for Internet experiments," TU Berlin, Doc. VCEG-Q15-I-16r1, New Jersey, Oct. 1999.
[53] Y. Guo, Y.-K. Wang, and H. Li "SVC/AVC loss simulator donation," USTC and Nokia Corporation, Doc. JVT-Q069, Bangkok, Thailand, Jan. 2006.
[54] S. Pateux and J. Jung, "An Excel add-in for computing Bjontegaard metric and additional performance analysis," Orange-France Telecom Research and Development, Doc. VCEG-AE07, Marrakech, Morocco, Jan. 2007.



Yi Guo received the B.S. degree in electronic information engineering from the Department of Electronic Engineering and Information Science at the University of Science and Technology of China, Hefei in 2004. He is currently working toward the Ph.D. degree in signal and information processing at the same university.

During April 2008–June 2008, he was working as an intern at Microsoft Research Asia, Beijing, China. His research interests include image/video processing, image/video coding, error control, and video adaptation.

Ying Chen (M'05) received the B.S. and M.S, degrees in mathematical sciences and electronics engineering and computer science from Peking University, Beijing, China in 2001 and 2004, respectively.

He is currently a Researcher with the Department of Signal Processing at Tampere University of Technology, Tampere, Finland. Before joining Tampere University of Technology, he worked as a Research Engineer at the Thomson Corporate Research, Beijing, China. His research interests include image processing and video coding and transmission. He has been an active contributor to ITU-T JVT and ISO/IEC MPEG, focusing on scalable video coding and multiview video coding standards. He has coauthored over 60 technical standardization reports and over 20 academic papers, and has over 20 issued and pending patents.

Mr. Chen is an external member of Research Staff at the Nokia Research Center, Finland, since September 2006.

Ye-Kui Wang (M'02) received the B.S. degree in industrial automation in 1995 from Beijing Institute of Technology, Beijing, China, and the Ph.D. degree in electrical engineering in 2001 from the Graduate School at Beijing, China, University of Science and Technology of China, Beijing.

From February 2003 to April 2004, he was a Senior Design Engineer at Nokia Mobile Phone. Before joining Nokia, he worked as a Senior Researcher from June 2001 to January 2003 at the Tampere International Center for Signal Processing, Tampere University of Technology, Finland. His research interests include video coding and transport, particularly in an error resilient and scalable manner. He has been an active contributor to different standardization organizations, including ITU-T VCEG, ISO/IEC MPEG, JVT, 3GPP SA4, IETF and AVS. He has been an editor for several (rdraft) standard specifications, including ITU-T Rec. H.271, and the MPEG file format and the IETF RTP payload format for the scalable video coding (SVC) standard. He has also been in the chair of the Special Session of Scalable Video Transport at the 15th International Packet Video Workshop in 2006. He has coauthored over 200 technical standardization contributions and about 40 academic papers. In addition, he has to his credit over 60 issued and pending patents in the fields of multimedia coding, transport, and application systems.

Dr. Wang is currently a Principal Member of the Research Staff with the Department of Signal Processing at Tampere University of Technology, Tampere, Finland.

Miska M. Hannuksela (M'02) received his M.S. and Ph.D. degrees in engineering from Tampere University of Technology, Tampere, Finland, in 1997 and 2009, respectively.

He is currently a Research Leader and the head of the Media Systems and Transport Team in Nokia Research Center, Tampere, Finland. He has more than ten years of experience in video compression and multimedia communication systems. He has been an active delegate in international standardization organizations, such as the Joint Video Team, the Digital Video Broadcasting Project, and the 3rd Generation Partnership Project. His research interests include scalable and error-resilient video coding, real-time multimedia broadcast systems, and human perception of audiovisual quality. He has authored more than 20 international patents and several tens of academic papers.

Houqiang Li received the B.S., M.S. and Ph.D. degrees in 1992, 1997, and 2000, respectively, all in electronic engineering and information science from the University of Science and Technology of China (USTC), Hefei.

From November 2000 to November 2002, he was a Postdoctoral Fellow at the Signal Detection Lab, USTC. Since December 2002, he has been on the faculty and currently he is the Professor with the Department of Electronic Engineering and Information Science at USTC. His current research interests include image and video coding, image processing, and computer vision.

Moncef Gabbouj (M'85–SM'95) received the B.S. degree in electrical engineering in 1985 from Oklahoma State University, Stillwater, and the M.S. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, in 1986 and 1989, respectively.

He is currently a Professor with the Department of Signal Processing at Tampere University of Technology, Tampere, Finland. He was Head of the Department during 2002–2007. His research interests include multimedia content-based analysis, indexing, and retrieval; nonlinear signal and image processing and analysis; and video processing and coding. He is currently on sabbatical leave at the American University of Sharjah, UAE, and Senior Research Fellow of the Academy of Finland.

Dr. Gabbouj has served as Distinguished Lecturer for the IEEE Circuits and Systems Society in 2004–2005. He served as Associate Editor of IEEE TRANSACTIONS ON IMAGE PROCESSING, and was Guest Editor of MULTIMEDIA TOOLS AND APPLICATIONS, the European journal of Applied Signal Processing. He is the past Chairman of the IEEE Finland Section, the IEEE CAS Society, Technical Committee on DSP, and the IEEE SP/CAS Finland Chapter. He was the recipient of the 2005 Nokia Foundation Recognition Award and co-recipient of the Myril B. Reed Best Paper Award from the 32nd Midwest Symposium on Circuits and Systems and co-recipient of the NORSIG 94 Best Paper Award from the 1994 Nordic Signal Processing Symposium.

[P5] S. Liu, Y. Chen, Y. -K. Wang,. M. Gabbouj, M.M. Hannuksela, and H. Li, “Frame Loss Error Concealment for Multiview Video Coding,” *IEEE International Symposium on Circuits and Systems, ISCAS’*08, Seattle, Washington, USA, May 18-21, 2008, pp. 3470–3473.

Frame Loss Error Concealment For Multiview Video Coding

Shujie Liu[1], Ying Chen[2], Ye-Kui Wang[3], Moncef Gabbouj[2], Miska M. Hannuksela[3], Houqiang Li[1]

[1]University of Science and Technology of China
Hefei, China
{lsjxbd@mail., lihq@}ustc.edu.cn

[2]Institute of Signal Processing
Tampere University of Technology
Tampere Finland
{ying.chen, moncef.gabbouj}@tut.fi

[3]Nokia Research Center
Tampere Finland
ye-kui.wang@nokia.com
miska.hannuksela@nokia.com

***Abstract*—The Multiview Video Coding (MVC) standard is currently under development by the Joint Video Team as an extension of the Advanced Video Coding (H.264/AVC) standard. An MVC encoder compresses more than one viewpoint of a scene captured by different cameras. Redundancies between views can be used for inter-view prediction in encoding as well as error concealment in decoding. In this paper, a new algorithm utilizing motion information of pictures from other views to conceal a lost picture is proposed. The algorithm first derives motion information for a lost picture based on motion fields of pictures in adjacent views. Then, traditional motion compensation is invoked within the view containing the lost picture to derive a concealed frame. Experimental results show that the proposed algorithm can improve video quality with a negligible computational complexity overhead compared to simple temporal error concealment algorithms.**

I. INTRODUCTION

Multiview video technologies have gained significant interest recently. In multiview video coding (MVC), the original video content is a group of video sequences captured by multiple cameras at the same time for the same scene but from different viewpoints. Typical multiview applications include free-viewpoint video, where the viewer can interactively choose his/her viewpoint in a three-dimensional (3D) space to observe a scene from a preferred perspective [1], and 3D television (TV), where multiple views are displayed simultaneously [2]. Due to the huge amount of original video data, the transmission part of multiview video systems relies heavily on compression of the video captured by multiple cameras. Fortunately, sophisticated coding tools can utilize correlations among different views to compress multiview content better than coding each view independently. One of the goals for the development of the MVC standard [3] is to discover such tools. Moreover, redundancies between views also provide opportunities for improved error concealment.

During transmission of video data, packet losses may occur. This may lead to undesirable effects such as system instability, unacceptable video quality and unpredictable decoder behavior. Therefore, techniques to control the impacts of transmission errors are highly desirable. A number of error control algorithms operating in the source coding layer, often referred to as error concealment methods, have been proposed in the literature [4][5]. They can be summarized into three categories: encoder and decoder interactive error concealment, error resilient encoding, and decoder error concealment [6]. Interactive error concealment typically utilizes the feedback from the receiver to adjust the encoding strategy to minimize error propagation. Error resilient encoding mainly utilizes some redundant information added at the encoder side. In this paper, we mainly focus on decoder error concealment. Some algorithms based on decoder error concealment have been proposed in [7][8][9]. They make use of temporal or spatial correlation between the macroblocks (MBs) in damaged area and its adjacent MBs in the same or previous frame. These algorithms assume that if either a single MB or a slice consisting of several consecutive MBs is lost, information from the neighboring available MBs or MBs in the adjacent frames can be used to estimate both motion vectors and texture of the missing MB. However, in some applications, a coded picture typically fits in one packet, and a transmission error will lead to a loss of a whole slice or frame.

Algorithms handling frame loss have also been proposed. For example, Belfiore et al. [10] addressed a pixel-level algorithm based on the optical flow theory, assuming that the motion between two consecutive pictures does not vary in a dramatic way. It exploited motion information in a few past frames to estimate forward motion of a lost frame. An error concealment algorithm called "BLSkip" was introduced for scalable video coding [11]. This algorithm gets motion vectors for the lost frame from the co-located MBs of the lower layers and outperforms simple temporal error concealment methods, such as "Frame Copy", where a lost frame is reconstructed by copying the reconstruction of the previous frame.

Getting motion information for a lost frame is important for the performance of an error concealment algorithm. In an MVC bitstream, there is usually a high correlation between two views. This correlation, such as motion field similarity, can be used to get a more accurate estimation of coding modes and motion vectors. In this paper, an algorithm utilizing this correlation is proposed. Experimental results show that the proposed algorithm can improve video quality comparing to low complexity temporal error concealment algorithms.

II. BACKGROUND

In this section, a brief overview of MVC is included first. Then, two error concealment methods, which are used for comparison, are discussed separately: (1) Frame Copy (FC); (2) Temporal Direct motion vector generation (TD). FC method can be used to conceal a lost picture of any type, while TD method conceals lost B pictures.

978-1-4244-1684-4/08/$25.00 ©2008 IEEE

A. Overview of MVC

An H.264/AVC bitstream consists of NAL units. NAL units can be categorized into video coding layer (VCL) NAL units and non-VCL NAL units. Coded pictures are contained in VCL NAL units. MVC, as an extension of H.264/AVC, has the same concepts for NAL and VCL.

The current coding scheme of MVC has a structure shown in Fig. 1. For each view, a prediction structure with hierarchical B pictures is used. There is a base view which is H.264/AVC compliant and independently coded. A view V can rely on other views for decoding. For view V, they are dependent views. Typically each view with an even view identifier (called even view for simplicity) has the previous even view as the dependent view and an odd view has the previous and next even views as dependent views. Normally, each GOP (Group of Pictures) contains an anchor picture and the rest pictures in the GOP are non-anchor pictures. An anchor picture is a picture that can be correctly decoded without the decoding of previous pictures in decoding order. Anchor pictures can serve as random access points.

B. Error concealment of frame copy (FC)

Frame copy is a simple error concealment method. In this algorithm, a lost picture is handled as P picture and a reference picture list construction process (with or without reference picture list reordering) is invoked to construct a reference list in a way that the first reference picture in the list is the previous picture of this lost picture in output order . Then, each pixel value of the lost picture is copied from the corresponding pixel of that first reference picture.

C. Temporal direct motion vector generation (TD)

TD is also an error concealment method that usually uses information in the same view [12]. In this algorithm, for each lost MB, the motion vectors (MVs) and references are generated as if it is coded using the "temporal direct mode", which is illustrated in Fig. 2.

As shown in Fig. 2, the motion vectors of the direct-mode are found in the first picture of Reference Picture List 1 (RefPicList1). The motion vector of the co-located block MV_C is scaled to obtain MV_0 and MV_1 based on the temporal distance TD_B and TD_D. Furthermore, in simulation, FC method is used as alternation for lost P pictures and I pictures.

III. Proposed Algorithm

In this section, the relationship between motion fields of different views is discussed first based on a geometric model. After that, the proposed algorithm, called Motion Prediction (MP), is described.

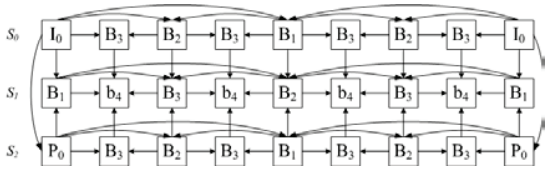


Figure 1. Inter-view temporal prediction structure

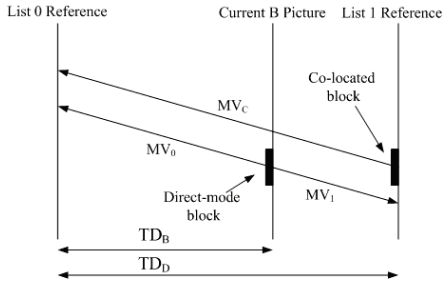


Figure 2. Example of temporal direct mode (MV generation)

A. Similarity of Motion Fields

Without loss of generality, we take the case of two-view video content as an example. Let P_t^0 denote the location of one pixel in view 0 at the time of t, while P_t^1 denote the location of the corresponding pixel in view 1. Moreover, P_{t-1}^i is the corresponding pixel of P_t^i at the time of t-1 shifted according to the temporal motion vectors MV_t^i. We can obtain the following equation:

$$P_t^i = P_{t-1}^i + MV_t^i \tag{1}$$

Normally, a six-parameter affine model is used to describe the global disparity between two views. The model includes a 2x2 transformation matrix $\boldsymbol{A}$ and a 2x1 displacement matrix $\boldsymbol{B}$. $\boldsymbol{A_t}$ and $\boldsymbol{B_t}$ are supposed to represent the affine model between two views at the time of t.

In this paper, the global disparity between two views is simplified to a two parameter model, which contains only the displacement between the two views, and the matrix $\boldsymbol{A}$ is simplified to an identity matrix. Furthermore, the geometrical locations and angles of the cameras are assumed to be identical during a short interval between a picture and its reference picture. Combining this assumption and (1), we can obtain:

$$MV_t^1 = MV_t^0 \tag{2}$$

If extended to a block, this equation means that the motion fields of corresponding blocks in two views do have similarity. And this similarity is used in the proposed algorithm, which is described in the following part.

B. Algorithm Description

In the encoder side, the global disparity motion regarding the inter-view reference picture relative to the base view is sent for every anchor picture. In the decoder, the MP algorithm is described as follows.

When a picture is detected as lost, each MB of the lost frame is processed as follows. First, the corresponding MB (CMB) in a dependent view is found according to the global disparity motion. As indicated by (2), the mode and motion vectors of CMB are then copied for the lost MB. Finally, motion compensation is used to generate the concealed picture. There are cases when the forward method is not applicable for

some special MBs or even a whole slice. They are concealed by other methods in MP algorithm:

If the CMB does not contain motion information, e.g., it is Intra coded, spatial error concealment for this MB is utilized. If the slice is a P slice, the current MB is set to skip mode in P picture; if the slice is a B slice, spatial direct mode is utilized.

If an anchor picture in one view that is not the base view is lost (it is detected to be an anchor picture if its corresponding picture in the base view with the same time instance is an anchor picture), it is concealed by copying sample values from that corresponding anchor picture of its dependent view.

IV. SIMULATIONS

A. Simulation conditions

In our simulations, the packet loss model in [13] was used, and we only focused on the cases of two views: view 0 and view 1, where view 0 is the dependent view of view 1. Two simulation models are used: simulation 1 and simulation 2. In simulation 1, each NAL unit was assumed to be contained in one packet. While in the other, one NAL unit was segmented into several 1400 bytes packets, if one of the segmentations is lost, the NAL unit is thought lost, and anchor pictures are assumed to be error free.

Only entire frame losses were considered, i.e., one slice per frame was encoded, which could be simply extended to slice loss cases. Different loss conditions were used in the simulation, with lower loss rate for view 0 than for view 1, which are all listed in TABLE I. The video sequences for the MVC common test conditions, *Akko&Kayo*, *Ballroom*, *Breakdancers*, *Exit*, *Race1*, *Rena*, and *Flamenco2*, were used but with only two views. Our implementation was based on the reference software of JMVM with a version of 4, and the common encoding settings were followed [14].

B. Results

The proposed error concealment method, FC, and TD were all tested. The performance comparison of the former two methods is reported in TABLE I. Fig. 3 shows the results of the three methods for the *Akko&Kayo, Exit* and *Rena* sequences at some conditions with no loss of view 0. It is clear that the proposed algorithm has a better performance than FC and TD both at the low packet loss rate and the high packet loss rate for *Akko&Kayo*, and has a little loss compared to TD for *Exit* in some conditions. Decoded pictures with degradation because of error propagation are shown in Fig. 4. They are the 162nd frame of *race1* and correspond to the three different error concealment algorithms. As shown in Fig.4, the proposed algorithm introduces less degradation.

The experimental results show that the proposed algorithm obtained significant gains compared to the FC and TD error concealment algorithms, up to more than 2.6 dB and on average about 0.97 dB compared with FC algorithm in terms of average luma peak signal-to-noise ratio (PSNR).

C. Discussion

There are cases when MP is close to FC or even a little worse than TD, mainly because most of the motion is local and the estimated disparity is not accurate enough for every MB. For simplicity, we used a simple displacement to model the transformation between two views. However, in practice, the optimal transformation between two views maybe non-linear and objects with different depths and different location may need different disparities. Those effects, however, can be compensated by local disparity motion vectors, e.g. in MB level. A potential improvement can use algorithms to estimate disparity motion vectors for each MB. Based on the assumption that the disparity remains the same during a short interval, we can estimate the disparity of each MB according to the adjacent correct decoded pictures in other views.

In the case discussed above, temporal motion field similarity can be better extracted and can predict the motion field well for the lost frame. That is why TD has close performance as MP, e.g. for *Exit*. Another improvement can adaptively choose the best motion prediction from either temporal or view dimension.

V. ACKNOWLEDGMENT

The work of Shujie Liu and Houqiang Li are partially supported by NSFC General Program under contract No. 60572067 and NSFC Key Program under contract No. 60736043.

VI. CONCLUSION

As multiview technologies become more and more mature, it is necessary to consider error concealment algorithms for coded multiview bitstreams. Error concealment can take advantage of inter-view redundancies. In this paper, a fast error concealment algorithm was proposed for multiview video coding. Motion field similarity between different views was utilized to generate the motion vectors for each MB of a lost picture, by predicting them from the motion vectors of the corresponding MB in another view. The corresponding MB can be estimated by several means, while in this paper we used the global motion disparity. The proposed method can conceal entire missing frame of a video sequence with low complexity and high quality. Simulation results showed that this method has significant gain over the frame copy method.

REFERENCES

[1] "Requirements on Multi-view Video Coding v.5," document N7539, MPEG, Nice, France, Oct. 2005.

[2] A. Vetro, W. Matusik, H. Pfister, J. Xin, "Coding Approaches for End-to-End 3D TV Systems," Picture Coding Symposium, 2004.

[3] "Joint Draft 4.0 on Multiview Video Coding", JVT-X209, Geneva, Swizerland, Jun.-Jul. 2007.

[4] B. W. Wah, X. Su, and D.Lin, "A survey of error-concealment schemes for real-time audio and video transmissions over the internet," Proc. Int. Symp. Mutimedia Software Engineering, Dec. 2000.

[5] P. Cuenca, L. Orozco-Barbosa, A. Garrido, F. Quiles, and T. Olivares, "A survey of error concealment schemes for MPEG-2 video communications over ATM networks," Proc. IEEE 1997 Can. Conf. Elect.and Comput, Eng., May 1997.

[6] Y. Wang, S. Wenger, J. Wen, and A.K. Katsaggelos, "Error resilient video coding techniques," IEEE Signal Processing Magazine, vol. 17, issue 4, pp. 61 - 82, Jul. 2000.

[7] J. Zheng and L. P. Chau, "A temporal error concealment algorithm for H.264 using Lagrange interpolation," International Symposium on Circuits and Systems, ISCAS 2004

[8] L. W. Kang and J. J. Leou, "A hybrid error concealment scheme for MPEG-2 video transmission based on best neighborhood matching algorithm," J. Visual Communication and Image Representation, vol. 16, issue 3, pp.288-310, 2005.

[9] Y-K. Wang, M. M. Hannuksela, V. Varsa, A. Hourunranta, and M. Gabbouj, "The error concealment feature in the H.26L test model," ICIP 2002, Sep, 2002.

[10] S. Belfiore, M. Grangetto, E. Magli and G. Olmo, "Concealment of whole-frame losses for wireless low bit-rate video based on multiframe optical flow estimation," IEEE Trans. on Multimedia, vol 7, issue 2, pp. 316-329, Apr. 2005.

[11] Y. Chen, K. Xie, F. Zhang, P. Pandit, J. Boyce, "Frame Loss Error Concealment for SVC", Journal of Zhejiang University SCIENCE A, also in Proc. Packet Video Apr. 2006.

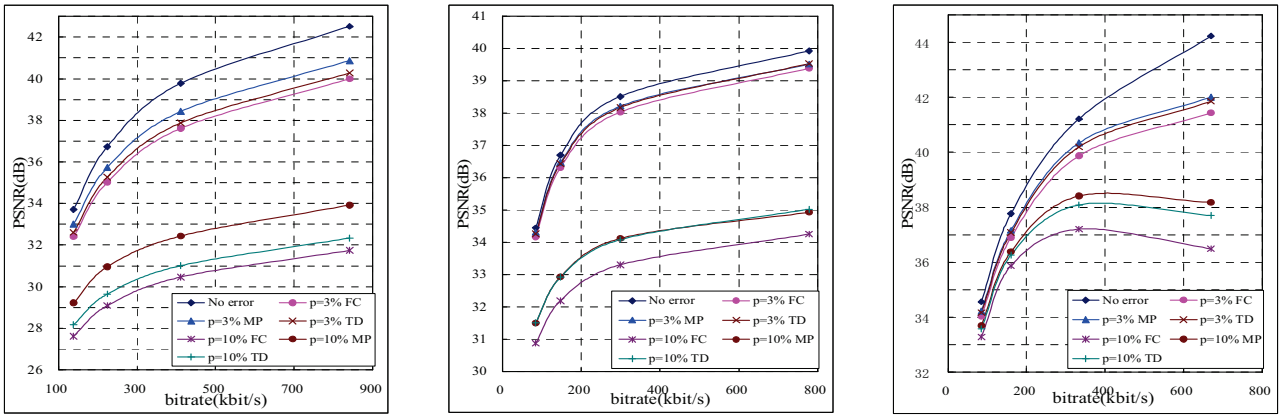
[12] ITU-T Rec. H.264 | ISO/IEC 14496-10, "Advanced video coding for generic audiovisual services, v3: 2005".

[13] S. Wenger, "Error Patterns for Internet Experiments," VCEG Q15-I-16r1, 2002.

[14] "Common Test Conditions for Multiview Video Coding," JVT-T207, Klagenfurt, Austria, Jul. 2006.

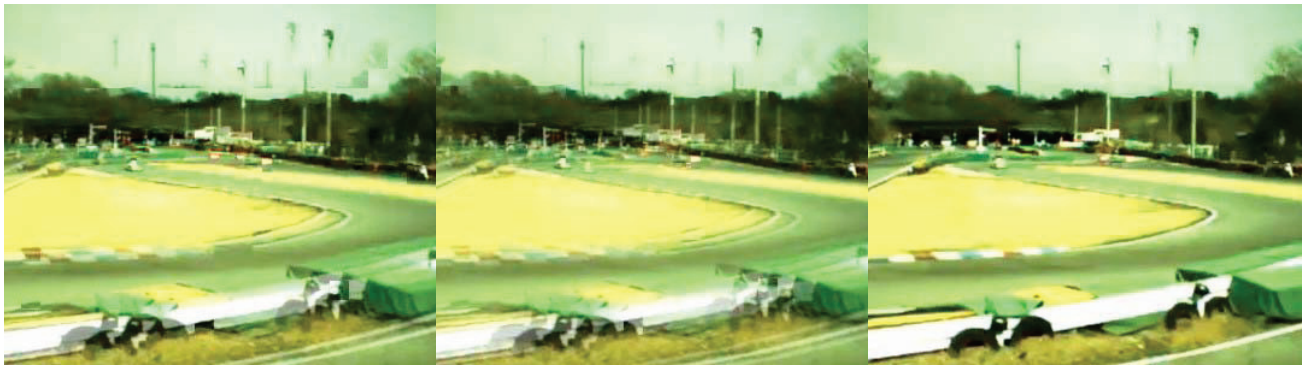
TABLE I. PSNR BETWEEN MP AND FC WHEN DIFFERENT PACKET LOSS RATES FOR VIEW 0 (Q) AND VIEW 1 (P) ARE APPLIED

Sequence	△PSNR(dB) (MP vs FC) (Simulation 1)						△PSNR(dB) (MP vs FC) (Simulation 2)					
	q = 0% p = 3%	*q = 0% p = 5%*	*q = 0% p = 10%*	*q = 3% p = 5%*	*q = 3% p = 10%*	*q = 5% p = 10%*	*q = 0% p = 3%*	*q = 0% p = 5%*	*q = 0% p = 10%*	*q = 3% p = 5%*	*q = 3% p = 10%*	*q = 5% p = 10%*
Akko&Kayo	0.77	1.15	1.93	1.16	1.43	1.16	1.20	1.91	2.63	1.53	2.13	1.47
ballroom	0.12	0.71	0.97	0.77	0.93	0.92	0.50	1.29	1.57	1.09	1.44	1.26
Breakdancers	0.14	0.56	0.53	0.66	0.56	0.67	0.47	1.04	1.01	1.05	1.06	1.11
exit	0.17	0.59	0.76	0.53	0.75	0.74	0.36	0.64	0.61	0.63	0.63	0.59
Flamenco2	0.24	0.31	0.48	0.44	0.62	0.62	0.40	0.79	1.07	0.92	1.18	1.15
race1	0.67	1.33	2.30	1.17	1.51	0.93	1.46	2.23	2.65	1.36	1.98	1.55
rena	0.34	0.45	0.76	0.57	0.76	0.71	0.37	0.67	0.93	0.61	0.96	0.89
(Average)	0.35	0.73	1.10	0.76	0.94	0.82	0.68	1.22	1.50	1.03	1.34	1.15



(a) Akko&Kayo p=3%, 10% (Simulation 1) (b) Exit p=3%, 10% (Simulation 1) (c) Rena p=3%, 10% (Simulation 2)

Figure 3. Comparison MP, TD, FC for Akko&Kayo, Rena and Exit sequences with different packet loss rates (p) in view 1



(a) Cropped picture corresponds to FC (b) Cropped picture corresponds to TD (c) Cropped picture corresponds to MP

Figure 4 Subjective quality comparison of pictures for "race1", with no loss in view 0 and 10%, loss rate in view 1, QP = 37.

[P6] Y. Chen, M. M. Hannuksela, L. Zhu, A. Hallapuro, M. Gabbouj and H. Li, "Coding Techniques in Multiview Video Coding and Joint Multiview video Model," *Picture Coding Symposium, PCS*'09, Chicago, Illinois, USA, May 6-8, 2009.

CODING TECHNIQUES IN MULTIVIEW VIDEO CODING AND JOINT MULTIVIEW VIDEO MODEL

Ying Chen[1], Miska M. Hannuksela[2], Ling Zhu[3], Antti Hallapuro[2], Moncef Gabbouj[1], and Houqiang Li[3]

[1]Department of Signal Processing, Tampere University of Technology
[2]Nokia Research Center
[3]University of Science and Technology of China

ABSTRACT

Since early 2006, Joint Video Team has been devoting on the development of Multiview Video Coding (MVC) standard as an extension of H.264/AVC. This MVC standard has been finalized in 2008. During the standardization of MVC, there was also a project namely Joint Multiview Video Model (JMVM), which focused on the advanced coding tools that are potentially useful. Those coding tools adopted into JMVM, including illumination compensation and motion skip, have not been added into MVC specification. In this paper, coding techniques in MVC as well as the tools in JMVM are described and discussed, focusing on the coding efficiency.

Index Terms— Multiview Video Coding, Joint Multiview Video Model, Coding Tool, Inter-view Prediction

1. INTRODUCTION

With the advances in acquisition and display technologies, 3D video is becoming a reality in the consumer domain with different application opportunities. 3D video applications can be grouped into two categories: free-viewpoint video [1] and 3D TV [2]. In free-viewpoint video, the viewer can interactively choose his/her viewpoint in 3D space to observe a real-world scene from preferred perspectives [1]. 3D TV refers to the extension of traditional 2D TV displays to displays capable of 3D rendering. Advanced auto-stereoscopic displays can support head-motion parallax, by decoding and displaying multiple views from different view-points simultaneously [2]. The content in above scenarios can be represented by multiple views, which of each is a traditional 2D sequence that was captured by a camera and can be used for 2D digital TV. Multiview video coding (MVC) is a key technology to enable a 3D video transmission system adopting multiview representation with efficient design for both transmission bandwidth and decoder resource consumption.

MPEG-2 and H.264/AVC can code two views by interleaving the left and right views in the temporal domain. In MPEG-2 Multiview Profile, one view, e.g., the left view can be coded in a reduced frame rate and the other view is coded as a temporal enhancement layer [3]. In H.264/AVC, the stereo video information supplemental enhancement information (SEI) message was adopted to indicate how two views are arranged in one bitstream [4]. The two views can be alternating frames or complementary field pairs.

Recent interests and advances in 3D video display technologies have also driven the requirements of using 3D content of more than two views. To meet this request as well as other requirements described in [5], MPEG issued its "call for proposals" [6] and started the standardization of MVC. Among all the proposed MVC solutions, the one based on H.264/AVC hierarchical B-pictures coding was selected as the basis of the MVC codec [7]. Since early 2006, the standardization efforts of MVC have been continued in Joint Video Team (JVT).

JVT maintains a joint draft of the MVC specification as well as a Joint Multiview Video Model (JMVM). MVC usually refers to the joint draft, which has the latest version in [8]. However, JMVM is a superset of MVC. Specific coding tools that have been demonstrated as useful could be adopted by JVT into JMVM [9]. In MVC, inter-view prediction is realized so-called disparity motion compensation, which follows the H.264/AVC motion compensation processes.

Multiview coding techniques have been summarized in [10] and most of them have also been proposed to JVT. Those techniques include Illumination Compensation (IC), Motion Skip (MS), view interpolation prediction, adaptive reference filtering and asymmetric coding. IC and MS will be further discussed in Section 3. The basic idea of view interpolation prediction is to estimate depth/disparity for synthesize predictions [11][12]. The adaptive reference filtering algorithm filters the integer pixels of the inter-view reference frames, to compromise the potential problem of focus mismatch among views [13]. Asymmetric coding targets on the scenario wherein one view of a stereo pair is coded with a quarter resolution of the other [14][15]. It requires substantially less bandwidth and complexity without noticeably scarifying the subjective quality. IC and MS have been adopted into JMVM. Others were not adopted because they either provide marginal compression efficiency gain or can only be applied for limited use cases.

In this paper, the coding techniques in JMVM are introduced. These techniques, when combined in different

scenarios, are compared with MVC and simulcast AVC under MVC common test condition. It can be concluded that those tools provides a relatively low coding efficiency increase. The rest of the paper is organized as follows. In Section 2, background of MVC is given, including the bitstream structure, prediction relationship and illustration of inter-view prediction in. JMVM Coding tools are presented in Section 3. Rate distortion (RD) results and decoding complexity analysis are given in Section 4. Section 5 concludes the paper.

2. MVC CODING STRUCTURE

In an MVC bitstream, a picture in a specific time instance of a specific view is called a view component. The view components are coded into Network Abstraction Layer (NAL) units which are ordered in a time-first coding order. In time-first coding, view components of any temporal location are contiguous in decoding order. All the coded NAL units of one time instance forms an access unit, also referred to as an (coded) picture. Note that the coded view components of an access unit follow the view order, which may not follow the ascending order of view identifiers. All the first view components of access units form a base view, which is decodable by H.264/AVC.

In MVC, view dependencies for inter-view prediction are defined for each coded video sequence. Inter-view prediction is enabled within the view dependencies to remove redundancies among views. With the exception of inter-view prediction, view components of each view are coded with the tools supported by H.264/AVC. In particular, hierarchical temporal scalability was found to be efficient for multiview coding. A typical prediction structure of MVC is shown in Fig. 1. It is noted that the MVC standard provides a greater deal of flexibility than depicted in Fig. 1 for arranging temporal or view prediction references.

There are anchor pictures wherein all view components are anchor view components. Anchor view components and all the view components (in the same view) succeeding in output order (i.e. display order) can be correctly decoded without decoding of previous view components in decoding order (i.e. bitstream order) and thus can be used as random access points. Anchor pictures (e.g., for T0 and T8 in Fig. 1), can be set as required. In MVC common test condition [16], there is an anchor picture for every 0.5 second.

An anchor picture and a non-anchor picture may have different view dependencies. While in a coded video sequence, all anchor or non-anchor pictures have the same view dependency respectively. A view dependency specifies the directly dependent views for each view, and is signaled in the sequence parameter set (SPS). Dependent views are signaled separately for the views that may be used as reference pictures in the two reference picture lists, namely list 0 and list 1. The dependent views corresponding to list 0/list 1 are also called forward/backward dependent views. A view that has both forward and backward dependent views is called a “B-view”, e.g., view 1, 3, 5 in Fig. 1. A view that has only forward dependent views is called a “P-view”, e.g., view 2, 4, 6, 7 in Fig. 1.

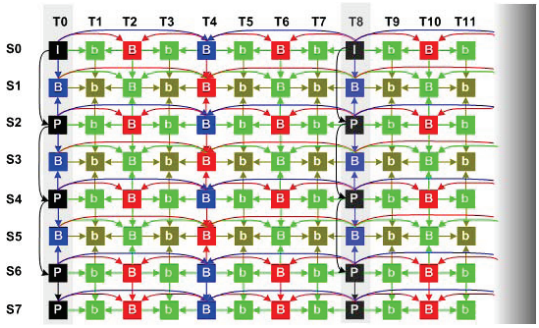


Fig. 1. Typical MVC prediction structure.

In MVC, view components in the same time instance can be used for inter-view prediction. It is utilized by putting the dependent view components into the reference picture lists of the view component that is being coded. In MVC, flexible reference picture list construction enables adding the view components of the time instance into list 0/1 and modifying the order of the inter-view prediction pictures and inter prediction pictures (view components in the same view) [17]. After the reference picture lists are constructed, the decoding processes of an MVC decoder are exactly the same as the processes of an H.264/AVC decoder.

3. CODING TOOLS IN JMVM

In JMVM, two coding tools are adopted on top of the MVC features. Illumination and color inconsistencies can happen due to different lighting conditions for multiple views. Although proper condition settings or preprocessing can also solve the problem, those are not always guaranteed. IC is the technique to solve this problem in the codec level, by subtracting the difference of the means of the reference block and the original block during motion compensation. This difference, namely locally illumination change are signaled for those blocks that use IC mode. At the decoder, an IC block is reconstructed by adding the motion compensation predictor, the residual and the illumination change, as shown in the following equation:

$$I(i, j) = R(i, j) + r(i + x, j + y) + C,$$

wherein $I(i, j)$ represents the reconstructed block, $R(i, j)$ is the residual signal, $r(i + x, j + y)$ is the reference block and C is the illumination change value for this block. At the encoder, each motion search needs an extra calculation of the means of the reference block.

MS is a motion vector (MV) derivation tool to enable reusing of the MVs from a view component in the same time instance but of other views by a given disparity for each macroblock (MB) that utilizes motion skip [19]. It is similar to SVC inter-layer motion prediction while does not enable motion refinement. In JMVM, a global disparity is

maintained and an offset is given for an MB with MS mode to calculate the local disparity, which is of 8-pixel accuracy. An example of motion reuse for a MB using MS mode is shown in Fig. 2. The current MB has a disparity points to four 8x8 blocks in an inter-view reference. The MVs of those blocks are reused for inter prediction motion compensation within the current view. The complexity increase at the decoder is minor, although the increase at encoder is substantially due to local disparity search.

4. SIMULATIONS

Our simulation followed the MVC common test condition defined in [16] and was based on the latest JMVM software [20]. The following scenarios are tested: Simulcast (each view is coded independently), MVC, IC (MVC inter-view prediction plus IC), MS (MVC inter-view prediction plus MS), IC+MS (MVC inter-view prediction plus two tools). For "P-views", inter-view prediction only applies to anchor pictures in the common tests. Since MS applies only to non-anchor pictures with inter-view prediction enabled, MS is not enabled for the "P-views". For IC, on the contrary, it works not only for inter-view prediction, but also for inter prediction. So, only the base view is not allowed to use IC, and a view component in a non-base view can be illumination compensated from its inter-view or inter prediction references.

The RD comparison results are shown in Table I, based on Bjontegaard measurement [21]. For each sequence, the compared Peak Signal-to-Noise Ratio (PSNR) and bitrate values are the average ones among all the views. MVC outperforms simulcast coding by an average bit-rate saving of 20%. MS and IC provide respectively 4% and 5% bitrate reduction over MVC, and in total, JMVM coding tools can achieve around 8.5% bit-rate saving, which is equivalent to 0.36 dB PSNR gain. Some typical RD curves are shown in Fig. 3, 4 and 5. It can be concluded that the coding efficiency improvements from simulcast to JMVM are mostly from the inter-view prediction tools defined in MVC. The coding tools, MS and IC, are potentially useful.

For the decoder, IC requires pixel level processing: adding a value to each block; MS requires only parsing and derivation for the MVs. Altogether, JMVM tools do not require significant complexity increase, although for hardware solutions, completely new modules need to be realized. MVC only requires slice header or higher syntax changes based on H.264/AVC, thus needs no new hardware implementations for major processing modules.

5. CONCLUSIONS

During the standardization of MVC, Joint Multiview Video Model has been maintained and it contains coding tools that are not included in the MVC specification. In this paper, those coding tools are reviewed and compared with the inter-view prediction in MVC. The JMVM coding tools, illumination compensation and motion skip can provide an average bitrate saving of about 8.5%, under MVC common test condition, which is much less than the gain of MVC achieved on top of H.264/AVC simulcast coding. There is neither decoding complexity increase nor extra hardware efforts for a MVC decoder compared to H.264/AVC decoders. The decoder complexity increase of JMVM is minor, but new hardware implementations are required.

6. REFERENCES

[1] A. Vetro, W. Matusik, H. Pfister, and J. Xin, "Coding approaches for end-to-end 3D TV systems," in *Proceedings of the 23rd Picture Coding Symposium (PCS '04)*, pp. 319–324, San Francisco, Calif, USA, December 2004.

[2] A. Smolic and P. Kauff, "Interactive 3-D video representation and coding technologies," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 98–110, 2005.

[3] J.R. Ohm, "Stereo/Multiview Encoding Using the MPEG Family of Standards," *In Proceedings of Electronic Imaging*, San Diego, USA, Jan. 1999.

[4] ITU-T Rec. H.264 | ISO/IEC IS 14496-10, "Advanced video coding for generic audiovisual services, v3: 2005."

[5] ISO/IEC JTC1/SC29/WG11, "Requirements on Multi-view Video Coding v.5," Doc. *N7539*, Nice, France, October 2005.

[6] ISO/IEC JTC1/SC29/WG11, "Call for Proposals on Multi-view Video Coding", Doc. *N7327*, Poznan, Poland, July 2005.

[7] P. Merkle, K. Mueller, A. Smolic, and T. Wiegand, "Efficient Compression of Multi-view Video Exploiting Inter-view Dependencies Based on H.264/MPEG4-AVC", *IEEE, International Conference on Multimedia and Expo (ICME) 2006*, Toronto, Canada, July 9-12 2006.

[8] "Joint draft 9.0 on multi-view video coding," *JVT-AB204*, Hannover, Germany, July 2008.

[9] "Joint multiview video model (JMVM) 8.0," *JVT-AA207*, Geneva, Switzerland, April 2008.

[10] A. Smolic, K. Mueller, N. Stefanoski, J. Ostermann, A. Gotchev, G.B. Akar, G.A. Triantafyllidis and A.Koz: "Coding Algorithms for 3DTV — A Survey," I*EEE Trans. on Circuits and Systems for Video Technology*, Vol 7, Issue 11, pp. 1606-1621, November 2007.

[11] S. Yea, A. Vetro, "Report on CE10 on View Synthesis Prediction," *JVT-U063*, Hanzhou, China, Oct. 2006.

[12] H. Kimata, S. Shimizu, M. Tanimoto, T. Fuji, K. Yamamoto, "CE10: Proposal on View Interpolation Prediction for MVC," *JVT-U093*, Hanzhou, China, Oct. 2006.

[13] J. H. Kim, P. Lai, J. Lopez, A. Ortega, Y. Su, P. Yin, and C. Gomila, "New Coding Tools for Illumination and Focus Mismatch Compensation in Multiview Video Coding," *Trans. Circuits Syst. Video Tech.*, vol. 17, no. 11, pp. 1519–1535, Nov 2007.

[14] C. Fehn, P. Kauff, S. Cho, H. Kwon, N. Hur, J. Kim, "Asymmetric coding of stereoscopic video for transmission over T-DMB," *Proc. 3DTV-CON 2007*, Kos Island, Greece, May 2007.

[15] Y. Chen, Y.-K. Wang, M. M. Hannuksela, M. Gabbouj, "Single-Loop Decoding for Multiview Video Coding," *IEEE ICME*, Hannover Germany, June 2008.

[16] "Common Test Conditions for Multiview Video Coding," *JVT-T207*, Klagenfurt, Austria, Jul. 2006.

[17] Y. Chen, Y. -K. Wang, K. Ugur, M. M. Hannuksela, J. Lainema, M. Gabbouj, "The Emerging MVC Standard for 3D

Video Services," *EURASIP Journal on Advances in Signal Processing*, Volume 2009, Article ID 786015, 2008.
[18] Y.-L. Lee, J.-H. Hur, Y.-K. Lee, K.-H. Han, S. Cho, N. Hur, J. Kim, J.-H. Kim, P.-L. Lai, A. Ortega, Y. Su, P. Yin, and C. Gomila, "CE11: Illumination Compensation," *JVT-U052*, Hangzhou, China, October, 2006.
[19] H. Yang, Y. Chang, J. Huo, S. Lin, S. Gao, L. Xiong "CE1, Fine motion matching for motion skip mode in MVC," *JVT-Z021*, Antalya, Turkey, January, 2008.
[20] P. Pandit, A. Vetro, Y. Chen, "JMVM 8 software," *JVT-AA208*, Geneva, Apr. 2008
[21] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," *VCEG-M33*, March, 2001.

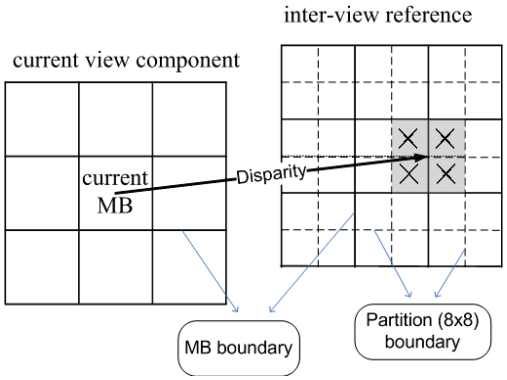


Fig. 2. Motion Vector Derivation in Motion Skip.

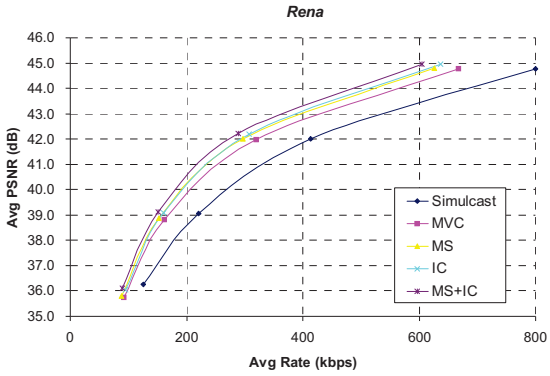


Fig. 3. RD Curves for "Rena".

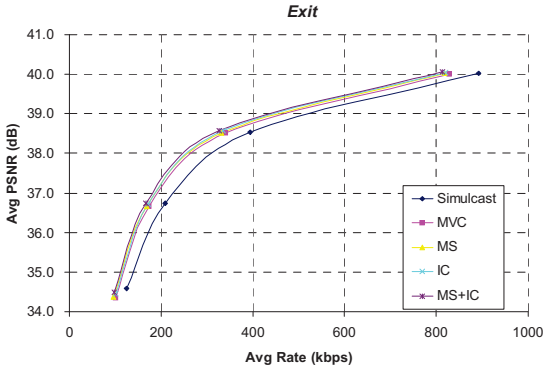


Fig. 4. RD Curves for "Exit".

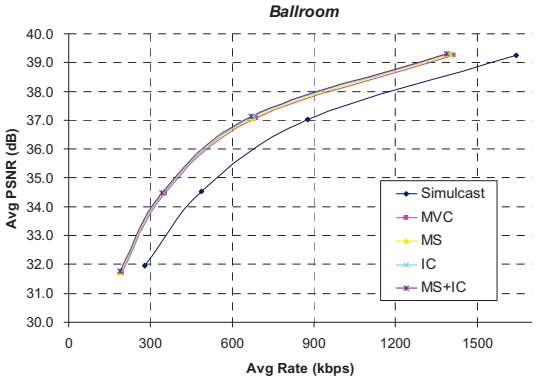


Fig. 5. RD Curves for "Ballroom".

TABLE I. Comparison between MVC and various scenarios

	MVC *vs* Simulcast		MVC *vs* MS		MVC *vs* IC		MVC *vs* MS+IC	
Sequence	Bit-rate	△PSNR	Bit-rate	△PSNR	Bit-rate	△PSNR	Bit-rate	△PSNR
Akyo&Kayo	34.23%	-1.430	-5.74%	0.294	-6.17%	0.320	-11.48%	0.598
Ballroom	32.29%	-1.105	-1.86%	0.073	-1.90%	0.072	-3.68%	0.144
Exit	16.05%	-0.407	-3.13%	0.087	-2.95%	0.079	-6.12%	0.168
Race1	27.00%	-1.015	-4.88%	0.208	-10.67%	0.462	-14.58%	0.652
Rena	26.94%	-1.051	-7.17%	0.339	-7.07%	0.341	-12.89%	0.635
Breakdancers	12.80%	-0.279	-3.06%	0.066	-6.43%	0.141	-8.69%	0.191
Flamenco2	12.42%	-0.526	-4.53%	0.211	-2.89%	0.135	-7.33%	0.354
Uli	-0.54%	0.022	-1.26%	0.050	-1.43%	0.056	-2.88%	0.114
Average	20.15%	-0.724	-3.96%	0.166	-4.94%	0.201	-8.46%	0.357

[P7] Y. Chen, Y.-K. Wang, M. M. Hannuksela, and M. Gabbouj, "Single-Loop Decoding for Multiview Video Coding," *IEEE International Conference on Multimedia and Expo, ICME*'08, Hannover, Germany, June 23-26, 2008, pp. 605–608.

SINGLE-LOOP DECODING FOR MULTIVIEW VIDEO CODING

Ying Chen[1], Ye-Kui Wang[2], Miska. M. Hannuksela[2], Moncef Gabbouj[1]

[1]Department of Signal Processing, Tampere University of Technology
[2]Nokia Research Center

ABSTRACT

Multiview video coding (MVC) is currently being standardized by the Joint Video Team as an extension of H264/AVC. When an MVC bitstream is decoded, some views (named target views) are to be displayed; some other views (named dependent views) may not be displayed but are needed for inter-view prediction of the target views. The original MVC design requires pictures of the dependent views to be fully decoded and stored. This entails both high decoding complexity and high memory consumption for the pictures in the views which are not intended for display, particularly when the number of dependent views is large. In this paper, a single-loop decoding (SLD) scheme is introduced to address these disadvantages. SLD requires only partial decoding of pictures in dependent views and thus significantly reduces decoding complexity and memory consumption. The proposed method is based on the so-called motion skip, wherein inter-view motion and coding mode prediction is exploited. Experimental results show that compared to coding schemes that require comparable complexity, significant compression gain can be achieved. For example, 25% bit-rate saving on average can be obtained compared to simulcast. Simulation results also show that the proposed SLD scheme provides a substantial reduction of complexity and memory size, at the expense of only a minor compression efficiency loss, compared with multiple-loop decoding MVC schemes.

Index Terms— Multiview Video Coding, MVC, Single-loop Decoding, Motion Skip, H.264/AVC

1. INTRODUCTION

Multiview video technologies have gained increasing interest recently. In multiview applications, the original video content is a group of video sequences captured by multiple cameras at the same time from the same scene through different viewpoints. The joint video team (JVT) is developing a multiview video coding (MVC) standard [1], which will become an extension of H.264/AVC. MVC takes advantage of the inter-view correlation to improve coding efficiency and also provides bit-rate and view scalability, error robustness, and control of decoding complexity [2].

Among typical MVC applications, such as free-viewpoint video [3], 3D TV [4], and immersive teleconferencing, there are cases that display only a subset of the encoded views. In this paper, the views that are displayed are referred to as the target views or output views, while the remaining views are referred to as the dependent views. Target views are related to dependent views through inter-view prediction. For example, in free-viewpoint video, only one view needs to be displayed at a certain time instance and that view is the target view.

In the latest joint draft (JD) of MVC [2], inter-view prediction is realized by utilizing pictures from other views as reference pictures for motion compensation. This inter-view prediction is referred to as inter-view sample prediction. In addition to the JD, JVT also maintains a joint multiview video model (JMVM) [5] for MVC. JMVM documents some additional tools that were shown to be potentially useful but not mature enough to be included into the JD. In the JMVM, there is a tool called motion skip, which enables motion prediction between views.

Single-loop decoding (SLD) [6] is supported in the scalable extension of H.264/AVC, also know as SVC [7]. The basic idea of SLD in SVC is as follows. To decode a target layer that depends on a number of lower layers, only the target layer itself needs to be fully decoded, while for the lower layers only parsing and decoding of Intra macroblocks (MBs) are needed. Essentially, SLD in SVC requires motion compensation only at the target layer. Consequently, SLD provides a substantial complexity reduction. Furthermore, since the lower layers do not need motion compensation and no sample values need to be stored in the decoded picture buffer (DPB), the decoder memory requirement is greatly reduced compared to conventional multiple-loop decoding (MLD), where motion compensation and full decoding is needed in every layer, as in the scalable profiles of earlier video coding standards.

In this paper, SLD is applied to MVC similarly as it is applied to SVC. In the proposed SLD for MVC, only the target views are fully decoded, while the non-anchor pictures of the dependent views need only to be parsed to obtain information required for inter-view prediction. We proposed the SLD scheme in [8] to be included in MVC, and a similar idea was also proposed in [9], with differences on high level syntax design. The proposed SLD scheme was adopted into the JMVM.

This paper is organized as follows. In Section 2, we explain how the motion prediction can be realized in MVC. In Section 3, illustrations are given for the proposed single-loop decoding scheme. Simulation results are provided in Section 4 and Section 5 concludes the paper.

2. MOTION PREDICTION IN SVC AND MVC

2.1. View dependency

In the draft MVC specification, an access unit consists of pictures of all the views pertaining to a certain display or output time. An anchor picture is a picture in a view that can be correctly decoded without the decoding of any earlier access unit in decoding order (i.e. bitstream order). Anchor pictures can serve as random access points. Other pictures are non-anchor pictures. View dependency is specified in the sequence parameter set (SPS) MVC extension, separately for anchor pictures and non-anchor pictures. A typical prediction structure of MVC is shown in Fig. 1, wherein anchor pictures are those of the time instances T0 and T8. Pictures within each view form a hierarchical temporal prediction structure. Prediction across views is also enabled, but is constrained in the same time instance.

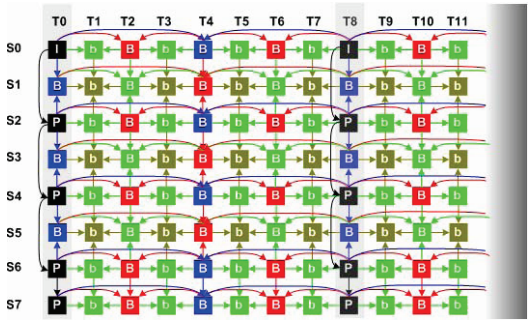


Fig. 1. Typical MVC prediction structure.

For a certain view, dependent pictures in other views are referred to as inter-view pictures. According to MVC JD, the texture (i.e. reconstructed sample values) of those pictures is required since inter-view pictures are used as reference pictures for motion compensation.

For bi-predictive (B) pictures, dependent views can be signaled in two prediction directions, corresponding to the reference picture list (we use list for simplicity) 0, and list 1. The views corresponding to list 0 (or list 1) are named forward (or backward) dependent views. For example, in Fig. 1, view 0 is the forward dependent view of view 1, while view 2 is the backward dependent view of view 1. A view that has both forward dependent views and backward dependent views is called a B-view. If a view has only forward dependent views, it is called a P-view.

2.2. Inter-layer motion prediction in SVC

In SVC, pictures in different layers of the same access unit coincide in time and thus the corresponding motion vectors have strong correlation. So, motion vectors of an MB are derived from the co-located MB(s) in the base layer. In spatial scalability, wherein resolutions of layers are different, base layer motion vectors are scaled. In CGS (Coarse Granularity Scalability), wherein layers have the same resolution, the derived motion vectors are the same as those of the co-located MB at the base layer. In SVC, the derived motion vectors can be further refined, and the refinement is coded into the bitstream.

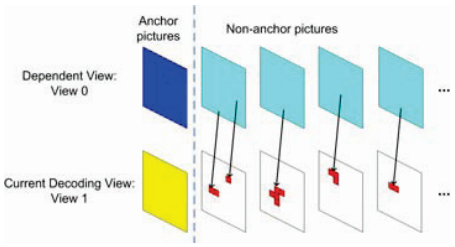


Fig. 2. Motion skip using disparity motion vector.

2.3. Motion skip in JMVM

In MVC, different views are captured by different cameras. This makes inter-view motion prediction in MVC harder and less efficient compared to inter-layer prediction of SVC.

As a coding tool in JMVM, motion skip predicts modes and motion vectors from the inter-view pictures and it applies to non-anchor pictures only. During encoding, a global disparity motion vector (GDMV) is estimated when encoding an anchor picture, and then GDMVs for non-anchor pictures are derived so that the GDMVs for a non-anchor picture is a weighted average from the GDMVs of the two neighboring anchor pictures. A GDMV is of 16-pel accuracy, i.e., for any MB in the current picture (i.e. the picture being encoded or decoded), the corresponding region shifted in an inter-view picture according to the GDMV covers exactly one MB in the inter-view picture. Normally, the first forward dependent view is considered for motion skip. However, if the corresponding MB in this picture is Intra coded, then the other candidate, which can be the first backward dependent view, if present, is considered. For an MB, if the motion skip mode is selected, motion vectors are derived and only residual between the original signal and the motion compensated signal are transmitted.

An example of motion skip is shown in Fig. 2, wherein view 0 is the dependent view and view 1 is the target view. With the disparity motion, when decoding MBs in view 1, the corresponding MBs in view 0 are located and their modes and motion vectors are reused as the MB modes and motion vectors for the MBs in view 1.

Unlike inter-view sample prediction, the use of which entails MLD since it requires motion compensation for

inter-view pictures, the use of motion skip does not require motion compensation of inter-view pictures. Thus, motion skip can be used for SLD.

3. SINGLE-LOOP DECODING FOR MVC

To achieve SLD, for coding of non-anchor pictures, only motion skip is applied for inter-view prediction. Inter-view sample prediction is only used for coding of anchor pictures. A flow chart for SLD at the decoder is shown in Fig. 3. When decoding an anchor picture, inter-view sample prediction can be used and the picture is fully decoded and stored into the DPB. When decoding a non-anchor picture, inter prediction and motion skip can be used, if this non-anchor picture belongs to a target view, it is motion compensated, reconstructed and stored in the DPB, otherwise, it is only parsed and only its coding modes and motion field are constructed.

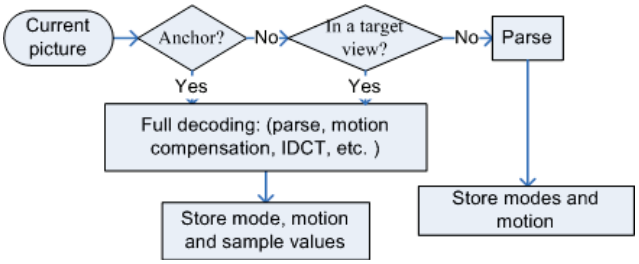


Fig. 3. SLD scheme in MVC.

An example of decoding complexity reduction and memory saving in the free-viewpoint video case is given for the prediction structure shown in Fig. 1.. It is assumed that view 5 is the only target view. When the MLD scheme is in use, all the pictures in views 0, 2, 4, 5 are to be fully decoded and stored in the DPB in order to successfully decode view 5. However, if the SLD scheme is in use and a non-anchor target picture is decoded, the dependent pictures in the same access unit need only to be parsed for generation of the coding modes and motion vectors, while their sample values do not need to be constructed. Consequently, approximately 60% of the decoding complexity and 50% of the memory for decoded pictures can be saved.

4. SIMULATION RESULTS

The proposed method was implemented based on the reference software of JMVM, version 5, and compared to three other coding scenarios: the first uses inter-view prediction only at the anchor pictures (AP), the second codes the views completely independently and is referred to as simulcast (SI), and the third is the MVC MLD. Sequences defined in the MVC common test condition [10], namely *Akko&Kayo, ballroom*, *exit*, *race1*, *rena* and *breakdancers*, were tested.

Table 1. Comparison of SLD to AP and simulcast

Sequence	SLD vs AP		SLD vs SI	
	Bit-rate saving	△PSNR	Bit-rate saving	△PSNR
Akko&Kayo	9.83%	0.477	37.39%	1.318
Ballroom	2.66%	0.105	26.05%	0.929
Exit	1.99%	0.056	17.29%	0.442
Race1	7.78%	0.318	30.62%	1.041
Rena	18.54%	0.788	37.03%	0.890
Breakdancers	4.00%	0.093	6.99%	0.165
Average	7.47%	0.306	25.90%	0.798

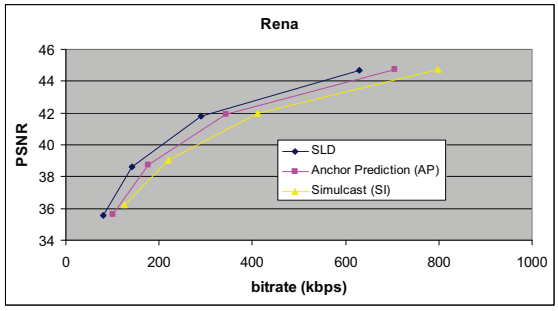


Fig. 4. RD curves for SLD, anchor prediction only and simulcast.

As shown in Table 1, compared to AP, there is up to 18.54% bit-rate saving (for *rena*) and the average among all the tested sequences is 7.47%. SLD shows especially better efficiency for low bit-rates, e.g., there is around 17% bit-rate saving for *Akko&Kayo* when quantization parameter (QP) is 37. Note that, in the tables, a bit-rate saving or Δ PSNR value greater than zero indicates that the left method is better than the right one. Results are generated using the Bjontegaard measurement [11] based on the bit-rates and average luma peak signal-to-noise (PSNR) values of the four test points corresponding to QP values 22, 27, 32, and 37.

One feasible way to avoid full decoding for the dependent views in the original MVC design is to disable the inter-view prediction at the non-anchor pictures and keep the inter-view sample prediction only at the anchor pictures. This AP scenario actually is a little less complex than SLD, because even parsing and motion prediction for the motion fields are not required for the non-anchor pictures in the dependent views. Furthermore, it does not require the storage of motion vectors. The efficiency of the motion prediction in SLD can be known by the comparison with AP scenario since there is no inter-view prediction in the non-anchor pictures in the AP scenario.

Compared to simulcast, the proposed SLD scheme has an average bit-rate gain around 25%. Rate-distortion (RD) curves for *rena* are shown in Fig. 4. The above results

show that SLD brings gain for the pictures compared with the cases when those pictures are coded with predictions only within a view.

In Table 2, the performances of SLD and MLD with or without motion skip (denoted as MLD and JD, respectively) are compared. Note that for SLD we enabled motion skip for non-anchor pictures in all the views, while for MLD and JD, inter-view prediction is enabled only for the B-views. Even with the additional prediction dependency, SLD is with comparable complexity and memory consumption in the P views and with much less complexity and memory computation in the B-views. So the total computational complexity and buffer required for the motion skip in SLD are still lower than JD or MLD.

Table 2. Comparison of SLD to JD and MLD

Sequence	SLD vs JD		SLD vs MLD	
	Bit-rate saving	△PSNR	Bit-rate saving	△PSNR
Akko&Kayo	2.62%	0.136	-0.83%	-0.037
Ballroom	-3.01%	-0.112	-3.57%	-0.134
Exit	-0.19%	-0.001	-0.94%	-0.022
Race1	3.80%	0.164	1.89%	0.082
Rena	9.30%	0.402	3.60%	0.154
Breakdancers	-3.70%	-0.083	-5.01%	-0.112
Average	1.47%	0.084	-0.81%	-0.0115

To show the efficiency of motion skip in SLD, the percentage of the MBs using inter prediction and inter-view motion prediction (motion skip) in the non-anchor pictures are shown in Fig. 5. As can be seen, a large percent of the MBs benefited from motion skip and selected the motion skip mode since it led to the best rate-distortion optimization for those MBs. Although the percentage of MBs coded with motion skip decreases when the bit-rate decreases, motion skip contributes more bit-rate saving for low bit-rates, because the share of bits for representing motion vectors is higher for low bit-rates than for high bit-rates.

5. CONCLUSIONS

In this paper, a single-loop decoding (SLD) scheme was proposed for multiview video coding. In the scheme, anchor pictures are coded with inter-view sample prediction and intra prediction while the non-anchor pictures are coded with motion skip, inter prediction, and intra prediction in the joint multiview video model (JMVM). The SLD scheme was compared with constrained MVC coding that exploits inter-view sample prediction only for the anchor pictures, hence achieving similar complexity and memory usage compared to the proposed SLD scheme. Experimental results showed that the proposed SLD scheme achieves a significant compression gain compared to the constrained MVC coding. Furthermore, the simulation results indicated that the proposed SLD scheme entails a minor compression efficiency loss compared to multiple-loop decoding (MLD) schemes but reduces decoder complexity and memory usage remarkably. The proposed SLD scheme has been adopted into the JMVM.

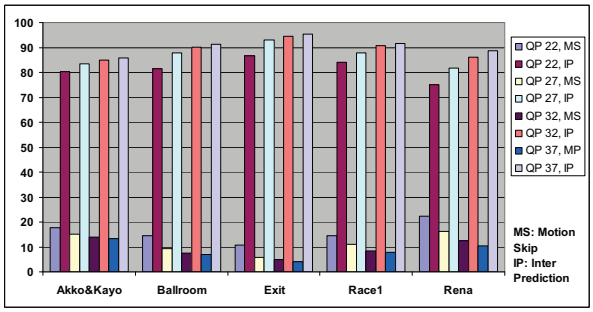


Fig. 5. Percentages of the MBs coded in motion skip and normal inter prediction modes.

6. REFERENCES

[1] "Joint Draft 5.0 on Multiview Video Coding," *JVT-Y209*, Shenzhen, China, Oct. 2007.

[2] ISO/IEC JTC1/SC29/WG11, "Requirements on Multi-view Video Coding v.5," *N7539*, Nice, France, Oct. 2005.

[3] A. Smolic, and P. Kauff, "Interactive 3D Video Representation and Coding Technologies", *Proc. IEEE, Special Issue on Advances in Video Coding and Delivery,* Jan. 2005.

[4] A. Vetro, W. Matusik, H. Pfister, J. Xin, "Coding Approaches for End-to-End 3D TV Systems," *Picture Coding Symposium*, 2004.

[5] "Joint Multiview Video Model (JMVM) 6.0," *JVT-Y207*, Shenzhen, China, Oct. 2007.

[6] H. Schwarz, T. Hinz, D. Marpe, T. Wiegand, "Constrained inter-layer Prediction for Single-Loop Decoding in Spatial Scalability," *IEEE Int. Conf. on Image Processing (ICIP),* 2005.

[7] "Joint draft 11 of SVC amendment," *JVT-X201*, Geneva, Switzerland, Jun.-Jul. 2007.

[8] Y. Chen, Y. –K. Wang, M. M. Hannuksela, "Single-loop decoding and motion skip study in JMVM," *JVT-Y053*, Shenzhen, China, Oct., 2006.

[9] P. Pandit, P. Yin, T. Dong, C. Gomila, H. Koo, Y. Jeon, B. Jeon, "MVC single-loop decoding," *JVT-Y042*, Shenzhen, China, Oct. 2006.

[10] "Common Test Conditions for Multiview Video Coding," *JVT-T207*, Klagenfurt, Austria, Jul. 2006.

[11] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," *VCEG-M33*, Mar. 2001.

[P8] Y. Chen, S. Liu, Y.-K. Wang, M. M. Hannuksela, H. Li, and M. Gabbouj, "Low-complexity Asymmetric Multiview Video Coding," *IEEE International Conference on Multimedia and Expo, ICME*'08, Hannover, Germany, June 23-26, 2008, pp. 773–776.

LOW-COMPLEXITY ASYMMETRIC MULTIVIEW VIDEO CODING

Ying Chen[1], Shujie Liu[2], Ye-Kui Wang[3], Miska M. Hannuksela[3], Houqiang Li[2], Moncef Gabbouj[1]

[1]Department of Signal Processing, Tampere University of Technology
[2]University of Science and Technology of China
[3]Nokia Research Center

ABSTRACT

Multiview video coding (MVC) is currently under development by the Joint Video Team (JVT) as an extension to Advanced Video Coding (H264/AVC). Based on the suppression theory in binocular vision, the fidelity of one of the two views of a stereoscopic display can be reduced without noticeable degradation of subjective quality. Thus, in MVC, a subset of views can be coded with lower spatial resolution at negligible cost to subjective quality. Due to different resolutions, a downsampling process is required in an MVC decoder in order to enable motion compensation (MC) between views. In this paper, a low-complexity MC algorithm is proposed for MVC to enable inter-view prediction between pictures with different resolutions. It requires lower memory consumption and lower computational complexity compared with the conventional downsampled inter-view prediction, while providing comparable efficiency, as shown by the simulation results.

Index Terms— Multiview video coding, asymmetric coding, motion compensation, decoded picture buffer

1. INTRODUCTION

Multiview video technologies have gained increasing interest recently. In multiview applications, the original video content is a group of video sequences captured by multiple cameras at the same time from the same scene. As the views have high correlation between each other, the JVT has worked to reduce the inter-view redundancy to obtain improved coding efficiency for the MVC standard [1], which will become an extension to H.264/AVC. Many of the display arrangements for multiview video are based on rendering of a different image to viewer's left and right eye. Hence, only two views are observed at a time in many typical MVC applications, such as 3D TV [2]. Based on the concept of asymmetric coding, one view in a stereoscopic pair can be coded with lower quality, while the perceptual quality degradation for the stereoscopic display is not noticeable by human eyes in comparison to the case when both of them are coded with equally high quality [3]. Therefore, a subset of all the views can be coded in lower resolution, e.g., quarter resolution, compared to the others and upsampled when being displayed. This scenario, referred to as asymmetric multiview video coding, requires less transmission bandwidth as well as lower complexity at the decoder for those low-resolution views.

For stereoscopic video, an approach has been proposed to code one view with high resolution while the other with half or quarter resolution [4]. In addition, there have been contributions to the JVT about asymmetric coding [5][6]. In the schemes proposed in [4] and [5], a part of views (e.g., every other view) is coded in the original resolution, while the remaining views can be coded with quarter resolution. To enable inter-view prediction between pictures with different resolutions, [4] and [5] took the following approach. The high-resolution pictures are downsampled before used for motion compensation (MC) by low-resolution pictures. Due to the fact that a high-resolution decoded picture is used as an inter prediction reference picture when the pictures in the same view (hence with the same resolution) are coded, the original version (with high resolution) is also kept in the decoded picture buffer (DPB). Since both the downsampled decoded picture and the full-resolution decoded picture coexist for one coded picture, the DPB size is inevitably increased. Instead of storing both downsampled and full-resolution decoded pictures, on-the-fly downsampling could be applied, but it would increase the computational demands for real-time processing. Therefore, the conventional solutions require either more memory or higher computational complexity.

In this paper, we propose a scheme with low complexity MC for the asymmetric MVC. This algorithm reduces the complexity for the decoding of the low-resolution views without increasing DPB size. Simulation results show that the proposed scheme provides comparable efficiency with the conventional solution.

This paper is organized as follows. In Section 2 and 3, asymmetric MVC and MC in H.264/AVC are reviewed. The proposed method is introduced in section 4. In Section 5, simulation results for coding efficiency of the proposed method are given, and then complexity reduction of the method is analyzed. Section 6 concludes the paper.

2. ASYMMETRIC MULTIVIEW VIDEO CODING

2.1. Inter-view prediction in MVC

A typical prediction structure of MVC is shown in Fig. 1, wherein T stands for the time axis and S stands for the view axis. Pictures in each view form a hierarchical bi-predictive (B) temporal prediction structure. The base view (view 0) is independently coded. For a picture in other views, inter-view prediction can be applied. Inter-view prediction is realized by placing the inter-view reference picture (inter-view picture for simplicity) into the reference picture lists and then the inter-view picture is utilized similarly to an inter prediction reference picture, and the inter-view prediction process is similar to the normal inter prediction process in the temporal direction. As shown in Fig. 1, a picture can use pictures in other views within the same time instance for inter-view prediction. In the joint draft of MVC [1], the inter-view prediction relationship is indicated as view dependency in the sequence parameter set MVC extension, wherein dependent views are signaled separately for the views that may be used as reference pictures in the two reference picture lists, namely list 0 and list 1. The dependent views corresponding to list 0/list 1 are also called forward/backward dependent views. A view that has both forward and backward dependent views is called a "B-view". For example, views 1, 3, 5 in Fig. 1 are "B-views".

In MVC, an anchor picture is a picture in a view that can be correctly decoded without the decoding of any earlier access unit in decoding order (i.e. bitstream order). Other pictures are non-anchor pictures.

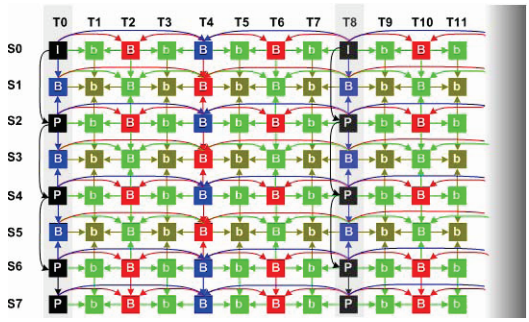


Fig. 1: Typical MVC prediction structure.

2.2. Asymmetric multiview video coding

In asymmetric MVC, for the stereoscopic scenario, two views are coded, one is in a higher resolution (e.g. VGA) and the other is in a lower resolution (e.g. QVGA). Downsampling is required to support the inter-view prediction between these two views. After that, MC of H.264/AVC can be applied. This asymmetric MVC approach is motivated by suppression theory of binocular vision [3], which indicates that the perceived sharpness and depth effect of a mixed-resolution stereoscopic pair is dominated by the higher-quality component, which can correspond to the right-eye, for example [4]. A 2D video communication system then can be enhanced to a 3D video communication system based on stereoscopic display, with acceptable transmission bandwidth increase, e.g., in a DVB-H (Digital Video Broadcasting –Handheld) system [7], and with reasonable decoder complexity increase, e.g. around 25%, compared with existing H.264/AVC decoders.

A picture in a DPB can play two different roles: an inter prediction reference picture for the following pictures in the same view, and an inter-view picture for the pictures in the same time instance. In asymmetric MVC, an inter-view picture, when referenced by a picture with a low resolution, needs to be downsampled to apply the conventional MC. Therefore, either the downsampled picture must be stored in the decoded picture buffer (DPB) or there must be an on-the-fly downsampling process. If a downsampled picture is added into the DPB, the required DPB size increases. If on-the-fly downsampling is used, the complexity increases, especially when the picture is used frequently as inter-view picture by lower-resolution pictures. These problems get worse when typical downsampling filter such as the MPEG-4 downsampling filter is applied. The MPEG-4 downsampling filter is [2, 0, −4, −3, 5, 19, 26, 19, 5, -3, -4, 0, 2]/64, which is a 13-tap filter and has 11 non-zero taps.

3. MOTION COMPENSATION IN H.264/AVC

In H.264/AVC, the accuracy of MC is in units of one quarter of the distance between luma samples. In case the motion vector points to an integer-sample position, the prediction signal consists of the corresponding samples of the reference picture; otherwise the prediction signal is obtained using interpolation to generate sample values at non-integer sample positions. The prediction values at half-sample positions are obtained by applying a one-dimensional 6-tap filter first horizontally and then vertically. The 6-tap filter utilized in H.264/AVC half-sample interpolation is [1, -5, 20, 20, -5, 1]/32 (named AVC filter for simplicity). Prediction values at quarter-sample positions are generated by averaging samples at integer-sample and half-sample positions. In the chroma component, a motion vector can point to integer sample, half-sample, quarter-samples or 1/8-sample positions. When the motion vector points to a non-integer sample position in a chroma component, the prediction values for chroma are obtained by utilizing the bilinear filter.

4. PROPOSED LOW-COMPLEXITY ASYMMETRIC MVC

In H.264/AVC, when a motion vector points to non-integer-sample positions, an area of the reference picture needs to

be interpolated. However, in the asymmetric MVC, since the inter-view picture is already with double width and height as the picture being coded, MC can be done without interpolation for the half-sample values. The downsampling of the inter-view picture to the same resolution, as well as the interpolation of values at half-sample positions can be avoided. Based on this, we provide a low-complexity MC approach for the asymmetric MVC. The proposed MC process consists of the following sub-processes.

4.1. Motion vector scaling

The motion vector to be used in the proposed MC process is first scaled as follows: *mv'=2mv*, wherein *mv* is the original motion vector and *mv'* is the scaled motion vector.

4.2. Luma motion compensation

In our proposed method, a scaled motion vector points to even-sample positions (when the original motion vector points to integer-sample positions), odd-sample positions (when the original motion vector points to half-sample positions), or half-sample positions (when the original motion vector points to quarter-sample positions) in the inter-view reference picture.

In Fig. 2, upper-case letters indicate samples on the full-sample grid, while lower case letters indicate samples in between at full-samples. When the scaled motion vector points to an odd sample position or an even sample positions, no interpolation is required and the sample values in those positions can be directly used. Similar to the method used in h.264/AVC for generating the luma values for quarter-sample positions, when the scaled motion vector points to a half-sample position, we average the integer samples.

4.3. Chroma motion compensation

Chroma MC is carried out in the same way as in H.264/AVC by applying the scaled motion vector to the chroma components of the inter-view reference picture. The scaled motion vector points to integer-sample positions, half-sample positions, or quarter-sample positions, but never points to 1/8 sample positions. As in H.264/AVC, the bilinear filter as follows and also shown in Fig. 3 is still used for interpolation of non-integer positions.

$$v = ((s-d_x)(s-d_y)A + d_x(s-d_y)B + (s-d_x)d_yC + d_xd_yD + s^2/2)/s^2$$

wherein *A, B, C, D* are the values of the integer samples and *v* is the value of the interpolated non-integer sample, s is 4 and d_x and d_y can be only 1, 2 or 3, while in H.264/AVC, s is 8 and d_x and d_y can be a value from 1 to 7.

5. SIMULATION AND ANALYSIS

5.1. Performance of the proposed approach

The proposed MC algorithm was implemented into the MVC reference software, JMVM (Joint Multiview Video Model) version 5. The low-resolution input views were generated by the MPEG-4 downsampling filter. The decoded low-resolution video was upsampled by the AVC filter for PSNR (luma peak signal-to-noise) calculation.

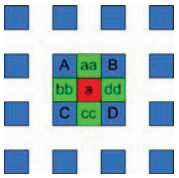


Fig. 2 Interpolation for half-sample luma values in the high-resolution picture.

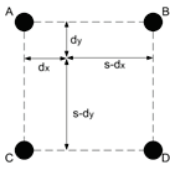


Fig. 3: Bilinear interpolation for chroma values in the non-integer positions.

Rate distortion (RD) performance was compared in two scenarios. The first scenario followed the JVT common test conditions [8] and the second one was the stereoscopic case, wherein two views were coded: view 0 was the base view and view 1 was of lower resolution and the pictures of view 1 are all depend on pictures (in the same time instance) in view 0. In both scenarios, three methods were compared: the proposed asymmetric MVC (PRO); the conventional asymmetric MVC (CON), wherein MPEG-4 downsampling was used [5]; and the simulcast MVC (SIM). When under common test conditions, SIM coded the base view and the "P-views" in one bit-stream with unchanged view dependencies and original spatial resolution. However, the "B-views" were coded into another MVC bitstream with quarter of the original resolution and with linear prediction structure. That is, e.g., if views 1, 3, 5 are the "B-views", view 1 is the forward dependent view of view 3, and view 3 is the forward dependent view of view 5. In stereoscopic case, there was only one high-resolution view (base view: view 0) and one low-resolution view (view 1) and thus the simulcast MVC was actually simulcast H.264/AVC.

The tested sequences were: *Exit, Ballroom, Rena, Race1, Akko&Kayo*, *Breakdancers* and *Uli*.

The performance comparisons for PRO and CON as well as simulcast are listed in Table 1. The bit-rate and PSNR values are generated for all the views. The asymmetric MVC approaches outperform simulcast MVC, and PRO has almost the same efficiency as that of the CON. Note that, in the tables, a bit-rate saving or ΔPSNR greater than zero indicates that the left method is better than the right one. Results are generated using the Bjontegaard measurement [9]. The curves (representing the bit-rate and PSNR values among the views) for *Akko&Kayo* are shown in Fig. 4, to save space, the RD curves for the "B-views" (the three left curves) and the curves for all the views (the three right curves) are shown in the same figure.

With the comparison results (only for view 1) shown in Table 2, similar conclusion can be reached for stereoscopic

case. Table 1 and 2 also show that asymmetric MVC is an efficient tool compared to simulcast.

Table 1. Comparison of PRO to conventional CON and simulcast (SIM) under common test conditions (all views)

Sequence	PRO vs CON		PRO vs SIM	
	Bit-rate saving	△ PSNR	Bit-rate saving	△ PSNR
Akko&Kayo	1.61%	0.073	22.43%	0.923
Ballroom	1.65%	0.058	16.29%	0.510
Exit	1.44%	0.039	12.59%	0.304
Race1	1.66%	0.056	12.58%	0.419
Rena	-0.42%	-0.020	18.57%	0.746
Breakdancers	0.52%	0.011	14.40%	0.292
Uli	0.03%	0.001	0.97%	0.034
Average	0.93%	0.031	13.98%	0.451

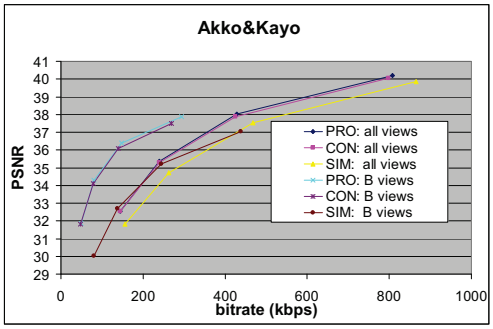


Fig. 4: RD curves for *"Akko&Kayo"*.

Table 2. Comparison of PRO to CON method and SIM for stereoscopic video (view 1 only)

Sequence	PRO vs CON		PRO vs SIM	
	Bit-rate saving	△ PSNR	Bit-rate saving	△ PSNR
Akko&Kayo	2.49%	0.082	144.14%	3.339
Ballroom	-0.07%	0.003	75.06%	1.743
Exit	2.79%	0.067	59.48%	1.121
Race1	1.66%	0.056	12.58%	0.056
Rena	-4.05%	-0.134	126.29%	0.419
Breakdancers	-0.92%	-0.022	73.93%	2.821
Uli	-0.19%	-0.005	2.81%	0.079
Average	0.24%	0.007	70.61%	1.368

5.2. Complexity analysis

In H.264/AVC, interpolation of the half-sample values for luma is the major part of the entire interpolation process at the decoder, and takes around 40% of the decoder execution time [10]. The results in [10] are for the Baseline profile. For the Main profile or a High profile, interpolation will takes a higher share of the decoding complexity because bi-predicted (B) pictures are present, especially when most of the pictures are B pictures, which is the case in hierarchical B prediction structure as shown in Fig. 1. These interpolation computations are saved for anchor pictures in our approach. For non-anchor pictures, similarly, in average approximately 40% complexity reduction can be expected for macroblocks coded using inter-view prediction.

If on-the-fly downsampling is utilized, it requires even higher complexity than interpolation because the downsampling filter, e.g., the MPEG downsampling filter, normally has many non-zero taps.

6. CONCLUSION

Asymmetric multiview video coding is a scenario for reducing transmission bandwidth and computational complexity. However, to support inter-view prediction, conventional solutions require downsampling of high-resolution pictures. Since memory usage and computational complexity are two vitally important aspects for a decoder design, conventional solutions are unfavorable. This paper presented a low-complexity motion compensation method for asymmetric multiview video coding, wherein the downsampling is not needed and the complexity of motion compensation is decreased compared to the conventional motion compensation. Moreover, there is no additional decoded picture buffer needed to store downsampled reference pictures in the proposed method. Simulation results showed that the coding efficiency of the proposed method is essentially identical compared to the coding efficiency of the conventional methods.

7. REFERENCES

[1] "Joint Draft 5.0 on Multiview Video Coding," *JVT-Y209*, Shenzhen, China, Oct. 2007.

[2] A. Vetro, W. Matusik, H. Pfister, J. Xin, "Coding Approaches for End-to-End 3D TV Systems," *Picture Coding Symposium,* 2004.

[3] Julesz B., *Foundations of Cyclopean Perception*, University of Chicago Press, Chicago, IL, USA, 1971.

[4] C. Fehn, P. Kauff, S. Cho, H. Kwon, N. Hur, J. Kim, "Asymmetric coding of stereoscopic video for transmission over T-DMB," *Proc. 3DTV-CON 2007*, Kos Island, Greece, May 2007.

[5] H. Kimata, S. Shimizu, K. Kamikura, Y. Yashima, "Inter-view prediction with downsampled reference pictures," *JVT-W079*, San Jose, CA, USA, Apr. 2007.

[6] Y. Chen, S. Liu, Y.-K. Wang, M. M. Hannuksela, H. Li, "Low complexity asymmetric multiview video coding," *JVT-Y054*, Shenzhen, China, Oct. 2007.

[7] G. Faria, J. A. Henriksson, E. Stare, and P. Talmola, "DVB-H: Digital Broadcast Services to Handheld Devices," Proceedings of the IEEE, vol. 94, no. 1, pp. 194-209, Jan. 2006.

[8] "Common Test Conditions for Multiview Video Coding*," JVT-T207,* Klagenfurt, Austria, Jul. 2006.

[9] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves*," VCEG-M33*, Mar. 2001.

[10] V. Lappalainen, A. Hallapuro, T.D. Hamalainen, "Complexity of optimized H.26L video decoder implementation," *IEEE Trans. on CSVT*, vol. 13, iss. 7, pp. 717–725, 2003.

[P9] Y. Chen, Y.-K. Wang, M. M. Hannuksela, and M. Gabbouj, “Picture-level Adaptive Filter for Asymmetric Stereoscopic Video,” *IEEE International Conference on Image Processing, ICIP*’08, San Diego, CA, USA, October 12-25, 2008, pp. 1944–1947.

PICTURE-LEVEL ADAPTIVE FILTER FOR ASYMMETRIC STEREOSCOPIC VIDEO

Ying Chen[1], Ye-Kui Wang[2], Miska M. Hannuksela[2], and Moncef Gabbouj[1]

[1]Institute of Signal Processing, Tampere University of Technology
[2]Nokia Research Center

ABSTRACT

In asymmetric stereoscopic video coding, one view is coded in a quarter of the resolution of the other and the low-resolution view is predicted from the high-resolution view. This way, stereoscopic video effect could be achieved with only moderately increased bandwidth and complexity. Inter-view prediction tools for generating the predictor of a maroblock (MB) or MB partition in the low-resolution view from the high-resolution view play a vital role for coding efficiency in asymmetric video coding. In this paper, we propose a method that applies an adaptive filter to generate picture-level adaptive inter-view predictors for MBs or MB partitions. At the encoder, a low complexity preprocessing module is built to find out the filters. Simulation results show that the proposed method provides a bit-rate saving of 26% at maximum and 5% on average.

Index Terms—Multiview video coding, stereoscopic video, asymmetric coding, inter-view prediction, motion compensation, adaptive filter

1. INTRODUCTION

The amount of interest in multiview video technologies has been increasing recently. As views are correlated, view redundancy has been reduced in the multiview video coding (MVC) standard [1], which will become an extension to H.264/AVC. Many display arrangements for multiview video are based on rendering of different images to viewer's left and right eyes. For example, when data glasses or autostereoscopic displays are used, only two views are observed by a viewer at a time in typical multiview applications, such as 3D TV [2], although the scene can often be viewed from different positions or angles. Based on the concept of asymmetric coding, one view in a stereoscopic pair can be coded with lower fidelity, while the perceptual quality degradation for the stereoscopic display can be negligible by human eyes [3]. Stereoscopic video applications therefore seem feasible with moderately increase of complexity and bandwidth on top of mono-view applications, even in the mobile application domain [4].

Asymmetric MVC or asymmetric stereoscopic video (ASV) codec can be realized by an H.264/AVC-compliant base view (view 0) and a lower resolution second view (view 1) compressed using inter prediction as specified in H.264/AVC as well as inter-view prediction. Approaches have been proposed to downsample base view pictures for inter-view prediction [4][5].

It is favorable to design the coding of low-resolution view in a manner with high efficiency as well as low complexity. A low-complexity motion compensation (MC) scheme has been proposed for asymmetric MVC to reduce the complexity of asymmetric MVC without compression efficiency loss [6]. In [6], direct MC from high-resolution inter-view reference picture to the low-resolution picture was proposed.

2D non-separable adaptive filter has been proposed for interpolating non-integer sample values for inter prediction in H.264/AVC [7]. The difference between inter-view prediction in ASV and inter prediction in H.264/AVC is that, in the former the reference pictures are of larger resolution than the picture to be predicted, and thus contain more information that can be potentially beneficial for inter-view prediction. To efficiently exploit this, we propose an algorithm to motion compensate the low-resolution picture from the high-resolution picture based on picture-level adaptive filters in this paper. A preprocessing module with a minor complexity increase to the encoder is designed to generate the optimal filters. Simulation results show that compared to the direct MC proposed in [6], about 5% bit-rate saving on average and up to 26% bit-rate saving can be achieved for ASV.

2. ASYMMETRIC STEREOSCOPIC VIDEO

A typical prediction structure of stereoscopic video is shown in Fig. 1. Pictures in a view form a hierarchical B temporal prediction structure. Each picture is associated with a temporal level. The base view (view 0, denoted as S0) is independently coded and the other view (view 1, denoted as S1) is dependent on view 0. Note that the MVC Joint Draft (JD) can deal with more views predicted from each other in the view dimension in a hierarchical manner [1].

Typically, in ASV, view 0 is in the original resolution (e.g., VGA) and the view 1 is in a lower (quarter) resolution (e.g., QVGA). The ASV approaches are motivated by the suppression theory of binocular vision [3], which indicates that the perceived sharpness and depth effect of a

stereoscopic pair with different fidelities is dominated by the higher-quality view, which can correspond to the right-eye, for example [4]. A 2D mobile system based on H.264/AVC then can be enhanced to stereoscopic mobile system with acceptable transmission bandwidth and decoder complexity increase, which is around 25%.

To support inter-view prediction between two views with different spatial resolutions, two solutions have been proposed. The first one downsamples the high-resolution views when they are used for inter-view prediction [4, 5] and the second applies a direct MC from the high-resolution pictures [6].

In [6], if the motion vector points to integer or half-sample positions in the virtually downsampled picture as in [4, 5], it will point to even or odd integer sample positions in the high-resolution (view 0) picture. If it points to quarter-sample positions, it will point to half-sample positions in the view 0 picture. As shown in Fig. 2, the samples in a 4x4 block can be predicted from an 8x8 block in the view 0 picture consisting of integer samples. The integer samples with the same parity then form a 4x4 block (each sample is predicted from the sample marked with the same number in the figure). When the motion vector points to half-sample positions in the view 0 picture, two neighboring integer sample values are averaged to get a predicted sample value.

In [6], the process of downsampling of the inter-view picture to the same resolution, as well as the interpolation of values at half-sample positions can be avoided for MC, and there is no extra buffer required to store the downsampled pictures. Storage of the high-resolution picture also preserves more information that can be utilized by potentially advanced inter-view prediction algorithms, e.g. the algorithm proposed in this paper.

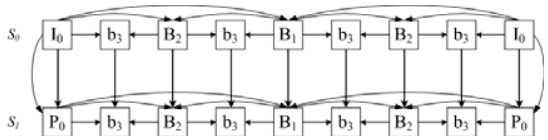


Fig. 1: Typical prediction structure for stereoscopic video.

3. ADAPTIVE FILTER FOR ASYMMETRIC STEREOSCOPIC VIDEO

For multiview content, different camera parameters may lead to differences that can not easily be compensated with conventional MC. Moreover, higher resolution in view 0 can potentially benefit the coding of view 1. Therefore, adaptive filters are designed to get a better predicted signal for the integer samples. A simple average from the filtered integer sample values is still used in the proposed algorithm for non-integer samples.

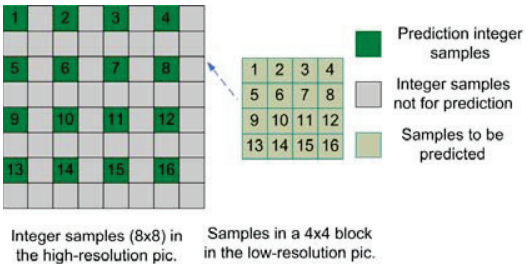


Fig. 2: MC when a motion vector points to integer samples in the inter-view picture (with high resolution).

3.1. Adaptive filter generation

Assume $\mathbf{S} = \{s_p \mid p = 1 \cdots M\}$ is the set of the samples in a picture of view 1 and the intensity (luma value) of s_p is b_p. For each sample s_p, there is a group of pixels that are located by motion estimation, e.g. using a block matching algorithm, at picture in view 0 in the same time instance. Let the pixel group corresponding to s_p be $R_p = [r_{p1}, r_{p2} \cdots r_{pn}]$ and their intensity values be $U_p = [u_{p1} \quad u_{p2} \quad \cdots \quad u_{pM}]$. Let the filter applied in inter-view prediction be $H = [h_1, h_2 \cdots h_N]^T$, wherein N is the length of the filter and also the number of the pixels in a corresponding pixel group. To get the best prediction of those samples in **S**, the following optimization problem is to be solved.

$$H^* = \arg\min_H (e^2) = \arg\min_H (\sum_{s_p \in S} (b_p - U_p \cdot H)^2)$$

The problem can be solved by Least Mean Square (LMS) algorithm as follows: $H^* = \mathbf{U}^+ \cdot$, wherein $\mathbf{U}^+$ is the pseudoinverse of the matrix **U**, which is of NxM and has all the $U_p, p = 1 \cdots M$ as row vectors and $= [b_1 \quad b_2 \quad \cdots \quad b_M]$.

There are two remaining issues: how to define the corresponding pixel group and how to construct set **S** that includes the pixels considered for optimization.

3.2. Corresponding pixel group locali ation

As there are disparities between views, the corresponding block in the picture in view 0 is found by block matching. Running the entire encoding process including motion estimation is an accurate way to get the disparity motion vectors. However, it requires multi-pass encoding and thus has shortcomings, e.g., high complexity, since the motion estimation and mode decision invoked in the H.264/AVC as well as MVC encoding includes searches for each reference pictures (several inter prediction reference pictures and one inter-view picture) and different modes of macroblock (MB) partitions and sub-macroblock partitions. So, in this paper, we utilize simple 16x16 block matching algorithm which greatly reduces the complexity. The block matching is

applied for each MB of the low-resolution picture. Only the integer samples in view 0 are searched.

Given a disparity motion vector, the best match sample in the inter-view picture (named center sample in this paper) for a pixel can be located by adding the disparity to the sample position. Two types of non-separable filters are proposed. Type-I filter requires a corresponding pixel group that has only the center sample and the nearest samples with the same parity (odd or even) as the center sample, as shown in Fig. 3 with upper case letters. Type-II filter requires a corresponding pixel group that includes also the samples with different parity, i.e., the center sample as well as all the nearby integer samples, with both upper case letters and the lower case letters in Fig. 3.

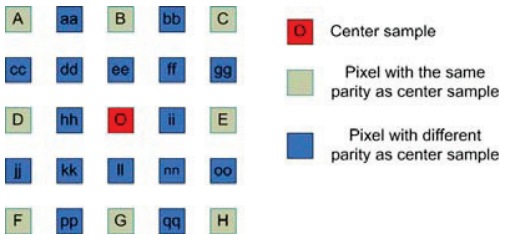


Fig. 3: Corresponding pixel groups for and 25 samples.

3.3. Relevant regions selection for adaptive filter

The optimization problem targets on the least square error for a specific sample set, which tends to use inter-view prediction. As a matter of fact, view 1 is coded in a hybrid way, which enables not only inter-view prediction (that utilizes adaptive filters) but also conventional H.264/AVC (intra-view) modes: inter prediction and intra prediction. The MBs or MB partitions for which intra-view modes are selected cannot benefit from the adaptive filter and thus allowing them be considered for adaptive filter generation can lead to a less optimal filter, which is less sensitive to the prediction errors of those samples finally predicted by inter-view prediction. So, relevant regions need to be well defined before the optimization equation is built. To guarantee that the adaptive filter is generated only by the prediction errors of the MBs for which inter-view prediction mode is preferred, the following function is proposed to select the MBs for the filter optimization.

$$f(MB_t)=\begin{cases}1 & Distortion(MB_t)\le T\\ 0 & else\end{cases}$$

wherein $f(\cdot)$ equal to 1 indicates that the t-th MB is selected for generation the adaptive filter and *Distortion*(.) returns the distortion of an MB. So we have

$$\mathbf{S}=\{s_t^i \mid s_t^i \in MB_t, f(MB_t)=1\}$$

For simplicity, MB_t denotes the t-th MB in the picture in view 1. The threshed T is content dependent and can vary picture by picture.

In this paper, T is set as follows to satisfy $Rate=|D|/NumMB, D=\{MB_t \mid f(MB_t)=1\}$, wherein $|\ |$ returns the number of elements in a set. When the *Rate* (percentage) of MBs that are used for inter-view prediction is decided, the threshold T can be decided by ordering the distortion values of all the MBs in a picture. Details of how to set different *Rate* values will be discussed in Section 4.

4. IMPLEMENTATION AND SIMULATION

The adaptive filter generation is applied by the following steps: 1. disparity estimation, 2. relevant MB selection, 3. construction and solving of the LMS equation. As step 1 and step 3 can follow canonical processes, in this section, we only describe more on the step 2.

As can be observed, inter-view prediction benefits pictures in different extents. Many factors influence this, but one of the most obvious factors is the temporal level (TL) of the pictures. Pictures with lower temporal levels tend to have more MBs predicted from the other view, because the temporal distance to intra-view reference for lower temporal levels is larger than for higher temporal levels. Thus, a simple method to get the *rate* values and therefore the threshold T is proposed by adopting the following temporal level to *rate* table (Table 1) for each sequence.

Table 1. Temporal level to *ra e* table

TL	0	1	2	3	4 or higher
a e	0.9	0.8	0.6	0.4	0.2

The proposed algorithm was implemented into the MVC reference software, JMVM (Joint Multiview Video Model) version 5 [8]. The tested sequences were *Exit*, *Ballroom*, *Rena*, *Race1*, *Akyo&Kayo*, *Breakdancers* and *Flamenco2*. For each video set, the first two views are selected to be coded as view 0 and view 1 in our simulation. Other parameters, e.g., the temporal prediction structure, the search range and the number of reference frames follow the common test condition of MVC [9].

The low-resolution input views were generated by utilizing the MPEG-4 downsampling filter, which is [2, 0, −4, −3, 5, 19, 26, 19, 5, -3, -4, 0, 2]/64. After decoding, the low-resolution video was upsampled by the H.264/AVC interpolation filter ([1, -5, 20, 20, -5, 1]/32) for peak signal-to-noise (PSNR) calculation.

The rate distortion (RD) performance comparison for the type-I filter (ASV-I), type-II filter (ASV-II) and the original ASV (ASV-O) [6] are compared and the results are listed in Table 2. Note that, in the table, a bit-rate saving or ΔPSNR value greater than zero indicates that the algorithm of the left is better than the one on the right. Results were generated using the Bjontegaard measurement [10] based on the bit-rate and average PSNR values of the four test points corresponding to different QP values.

For the *ballroom* sequence, the RD curves of view 1 are shown in Fig. 4. The performance increases resulting from the use of type-I and type-II filters are noticeable.

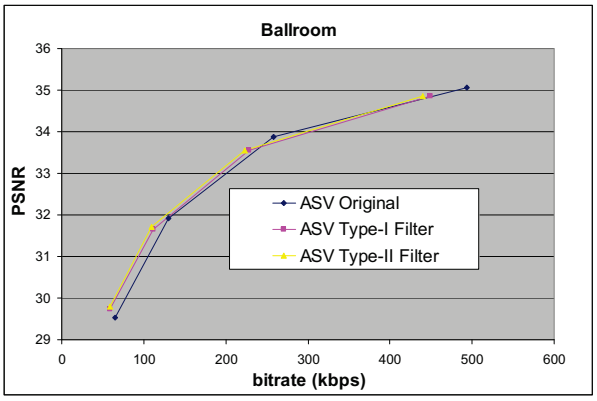


Fig. 4: RD curves for "*a room*".

Table 2. Comparison of ASV-II filter to ASV-II and ASV-O (view 1 only)

Sequence	ASV-I vs ASV-O		ASV-II vs ASV-O	
	Bit-rate saving	△PSNR	Bit-rate saving	△PSNR
Akyo &Kayo	-4.85%	-0.172	-3.32%	-0.120
Ballroom	5.82%	0.134	9.11%	0.221
Exit	-1.80%	-0.041	0.28%	0.003
Race1	-10.39%	-0.216	-6.07%	0.118
Rena	21.76%	0.622	26.67%	0.755
Breakdancers	6.42%	0.144	5.94%	0.137
Flamenco2	2.16%	0.063	1.07%	0.042
Average	2.73%	0.076	4.81%	0.165

Although the filters have many taps, they are applied only once to get the prediction value for each sample. So, if the related samples are in a small region (e.g., 5x5), the complexity of MC can be comparable to the H.264/AVC half-sample positions MC, which requires interpolation that may be related to a region as large as 6x6 for a sample.

5. DISCUSSION AND FUTURE WOR

As shown in Table 2, the proposed method can give an average bit-rate saving of 4.81%. However, it also introduces loss to some sequences. The proposed method relies on the distribution of the disparity motion vectors. If the picture contains different depth-levels, e.g., as shown in Fig. 5 for *Race1*, the adaptive filter for the global picture is not optimal and will even blur the signal and provide worse performance. Another typical distribution of disparity motion vectors for *Rena* indicates a highly converged single depth level and high efficiency is observed for this sequence. To further improve coding efficiency for those sequences with multiple depth levels, multiple adaptive filters can be utilized for different regions.

6. CONCLUSION

Picture-level adaptive filter was proposed to generate a better predicted signal in inter-view prediction for asymmetric stereoscopic video coding, wherein the second view is coded in a quarter resolution compared to that of the H.264/AVC-compliant base view. In the method, relevant marcoblocks for filter generation are selected based on their distortions as well as the associated temporal levels. Two types of filters were proposed, the first one only uses samples with the same parity (odd or even) with 2.7% bit-rate saving and the second one fully utilizes odd and even integer samples, with 2.1% additional bit-rate saving on average on top of the first type.

7. REFERENCES

[1] "Joint Draft 5.0 on Multiview Video Coding," *JVT-Y209*, Shenzhen, China, Oct. 2007

[2] A. Vetro, W. Matusik, H. Pfister, J. Xin, "Coding Approaches for End-to-End 3D TV Systems," *Picture Coding Symposium,* 2004

[3] Julesz B., *Foundations of Cyclopean Perception*, University of Chicago Press, Chicago, IL, USA, 1971

[4] C. Fehn, P. Kauff, S. Cho, H. Kwon, N. Hur, J. Kim, "Asymmetric coding of stereoscopic video for transmission over T-DMB," *Proc. 3DTV-CON 2007*, Kos Island, Greece, May 2007

[5] H. Kimata, S. Shimizu, K. Kamikura, Y. Yashima, "Inter-view prediction with downsampled reference pictures," *JVT-W079*, San Jose, CA, USA, Apr. 2007

[6] Y. Chen, S. Liu, Y.-K. Wang, M. M. Hannuksela, H. Li and M. Gabbouj, "Low complexity asymmetric multiview video coding," *IEEE Proc. ICME*, 2008.

[7] Y Vatis, B. Edler, D. T. Nguyen, J. Ostermann, "Motionand Aliasing-Compensated Prediction Using a Two-dimensional Non-separable Adaptive Wiener Interpolation Filter," ICIP 2005

[8] "JMVM 5 software," *JVT-X208*, Geneva, Switzerland, Jun.-Jul. 2007

[9] "Common Test Conditions for Multiview Video Coding*," JVT-T207,* Klagenfurt, Austria, Jul. 2006

[10] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves*," VCEG-M33*, Mar., 2001

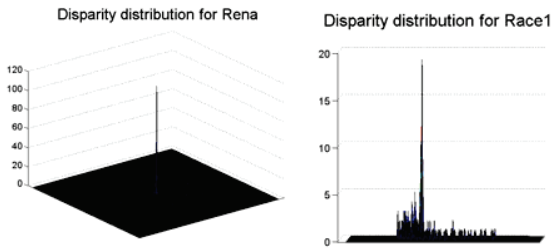


Fig. 5: Histograms of the disparities for "*ena*" and *a e* .

[P10] Y. Chen, Y.-K. Wang, M. M. Hannuksela, and M. Gabbouj, "Regionally Adaptive Filtering for Asymmetric Stereoscopic Video Coding," *IEEE International Symposium on Circuits and Systems, ISCAS*'09, Taipei, May 24-27, 2009, pp. 2585–2588.

Regionally Adaptive Filtering for Asymmetric Stereoscopic Video Coding

Ying Chen[1], Ye-Kui Wang[2], Moncef Gabbouj[1], and Miska M. Hannuksela[2]

[1]Department of Signal Processing
Tampere University of Technology, Tampere, Finland
ying.chen, moncef.gabbouj tut.fi

[2]Nokia Research Center
Tampere, Finland
ye-kui.wang, miska.hannuksela nokia.com

***Abstract*—In asymmetric stereoscopic video coding, one view can be coded in a lower resolution of the other. In this scenario, stereoscopic video can be compressed with only moderately increased bandwidth and complexity compared to 2D mono-view video coding. The subjective quality degradation of this scenario can be negligible compared to coding two views with original resolution. The low-resolution view can be predicted from the high-resolution view to achieve higher coding efficiency. In this paper, a regionally adaptive filtering algorithm is proposed to generate a predictor of a macroblock (MB) or MB partition of the low-resolution view from the high-resolution view. Different filters are applied for different picture regions. Disparity motion matching and clustering are applied in the encoder for generation of regionally adaptive filters. Simulation results show that the proposed algorithm results in up to 27% bit-rate saving compared with methods without adaptive filtering.**

I. INTRODUCTION

Multiview video technologies have gained significant interest recently. As views are highly correlated, efforts have been undertaken by the Joint video team (JVT) to reduce redundancy between coded views in the Multiview Video Coding (MVC) standard [1], which is an extension to the Advanced Video Coding Standard (H.264/AVC). Many display arrangements for multiview video are based on rendering a different image to the viewer's left and right eyes. For example, when data glasses or autostereoscopic displays are used, only two views are observed at a time in typical multiview applications, such as 3D TV [2], although the scene can often be viewed from different positions or angles. Based on the concept of asymmetric coding, one view in a stereoscopic pair can be coded with a lower fidelity, while the perceptual quality degradation can be negligible [3]. Thus, stereoscopic video applications may be feasible with moderately increased complexity and bandwidth requirement compared to mono-view applications, even in mobile applications domain [4].

It is desirable to have an Asymmetric Stereoscopic Video (ASV) coding system based on MVC, where one view is compliant to the existing mono-view standard, i.e., H.264/AVC, and the other view can be coded with techniques that provide high efficiency by exploiting redundancies between views. Approaches have been proposed in the literature to enable inter-view prediction in an ASV codec. In [4], a downsampling process is invoked before inter-view prediction [4]. In [5], direct motion compensation (MC) scheme has been proposed to substantially reduce the complexity without compression efficiency loss.

In the context of mono-view video coding, 2D non-separable adaptive filters have been proposed for the interpolation of values of non-integer sample positions a motion vector points to [6]. In this paper, regionally adaptive filtering algorithms focusing on the integer sample positions are applied to reduce the correlation between the high-resolution picture and the low-resolution picture further and address the potential focus mismatch problem. The proposed techniques target stereoscopic video applications with minor bandwidth increase compared to mono-view video communications but with subjective quality comparable to coding two views with roughly double the bandwidth. The proposed techniques provide about 8% bit-rate saving on average, and 27% bit-rate saving at most, which is equivalent to more than 0.7 dB luma peak signal-to-noise (PSNR) gain for the low-resolution view.

The rest of this paper is organized as follows. In Section II, ASV and the disparity motion compensation methods are described. In Section III, the proposed regionally adaptive filtering algorithm is presented. Implementation details and simulation results are provided in Section IV. Discussions are given in Section V and Section VI concludes the paper.

II. ASYMMETRIC STEREOSCOPIC VIDEO

A typical prediction structure of stereoscopic video is shown in Fig. 1. Pictures in each view form a hierarchical bi-predictive (B) temporal prediction structure. The base view (view 0, denoted as S0) is independently coded and the other view (view 1, denoted as S1) is dependent on view 0. Note that the MVC standard support more views predicted from each other in the view dimension in a hierarchical manner [1]. In ASV, view 0 is in the original resolution (e.g. VGA) and view 1 is coded in a quarter resolution (e.g. QVGA) of view 0. ASV approaches are motivated by the suppression theory of binocular vision [3], which indicates that the perceived sharpness and depth effect of a mixed-resolution stereoscopic

978-1-4244-3828-0/09/$25.00 ©2009 IEEE

pair is dominated by the higher-quality component. It is foreseen that a 2D mono-view mobile system can be enhanced to a stereoscopic mobile system with about 25% transmission bandwidth and decoder complexity increase.

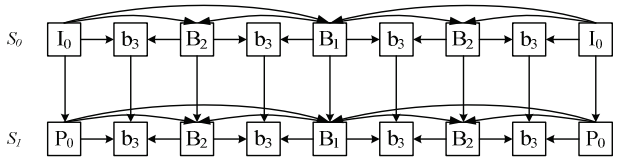


Figure 1. Typical prediction structure for stereoscopic video.

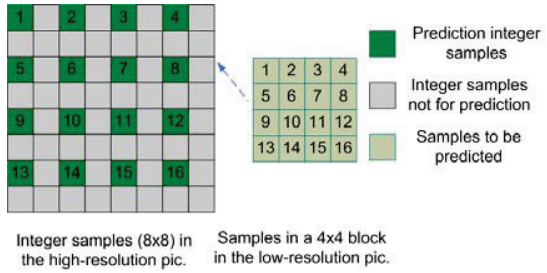


Figure 2. DMC when a motion vector points to integer samples in the a picture in view 0.

To decrease the bandwidth of view 1 further, inter-view prediction from view 0 to view 1 can be applied. One solution to enable inter-view prediction has been proposed in [4]. Since inter-view prediction, as the case in MVC, utilizes the motion compensation (MC) in H.264/AVC to realize so-called disparity compensation, view 0 must be downsampled for inter-view prediction [4].

To eliminate the potential extra buffer requirement or complexity increase caused by downsampling, another solution, based on a Direct MC (DMC) from the high-resolution (view 0) pictures, was proposed in [5]. In DMC, disparity compensation is done directly in the high-resolution picture of view 0. So, if a disparity motion vector of a low-resolution (view 0) picture points to integer or half sample positions in the virtually downsampled view 0 picture, it points to even or odd sample positions in the high-resolution (view 0) picture and a MC prediction block is formed from those integer samples. This is illustrated in Fig. 2, where the samples in a 4x4 block can be predicted from 16 pixels in a view 0 picture consisting of either all odd or all even integer samples (each sample is directly predicted from the sample marked with the same number in the figure). If the motion vector virtually points to quarter-sample positions, it points to half-sample positions in the high-resolution picture in view 0 and the DMC averages two neighbouring integer samples [5]. As reported in [5], the performance of the DMC is similar to downsampling based solutions proposed in [4]. However, since the inter-view picture is of high resolution, more information can be potentially utilized for inter-view prediction algorithms, based on DMC.

III. Regionally Adaptive Filtering

For multiview content, phenomena, such as imperfect calibration, different camera parameters, and focus changes across views, may lead to less efficiency in the inter-view prediction based on H.264/AVC MC or DMC. During DMC, only of the integer pixels in the high-resolution picture are considered for compensation and of the other pixels in the prediction area are not used. Those pixels can be potentially beneficial in coding of the low-resolution view. Moreover, multiview sequences can have regions with depth level difference. Regions of different depth levels are affected, e.g., blurred at different extents, because of focus mismatch. This makes the picture-level global adaptive filter [7] less efficient, since the resulting filter may not be optimal for some regions. To exploit the information in the high-resolution view more elegantly, a regionally adaptive filtering algorithm is proposed. Multiple optimal filters are applied to different regions of an inter-view picture. The proposed algorithm is realized as follows.

A. Regionally Adaptive Filter Generation

Assume that we have K depth levels in a picture, the following optimization problem is to be solved.

$$\mathrm{H} = \arg\min_{\mathbf{H}}(e^2) = \arg\min_{\mathrm{H}}\Big(\sum_{i=1\cdots K}\sum_{s_p^i \in S^i}(b_p^i - U_p^i \cdot H^i)^2\Big) \quad (1)$$

wherein $\mathrm{H} = H^1, \cdots H^K$ are the K filters and $\boldsymbol{S}^i$ is the set of the pixels belonging to the i-th depth level. b_p^i are the sample values of pixels in set $\boldsymbol{S}^i$ and U_p^i are the vectors containing sample values of the group of pixels corresponding to b_p^i. The filter applied in inter-view prediction targeting on the i-th depth level is $H^i = \left[h_1^i \quad h_2^i \quad \cdots \quad h_N^i\right]^T$, wherein N is the length of the filter and also the number of pixels in a group of corresponding pixels, which are used to obtain one prediction value. To solve this problem, the first step is to identify different sample sets $\boldsymbol{S}^i$ and the next step is to solve each optimization problem for each depth level. Each problem can be solved by Least Mean Square (LMS) algorithm, the same algorithm used in [7].

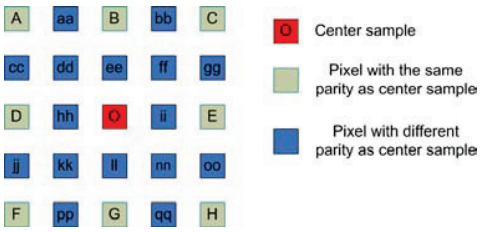


Figure 3. Corresponding group of pixel with 25 samples.

B. Locating the Corresponding Group of Pixels

The best matching sample in the inter-view picture (named center sample in this paper) for a pixel can be located by adding the disparity motion vector to the sample position. A corresponding group of pixels that contain the center sample and the nearest samples (with the same or different parity as the center sample) are shown in Fig. 3. In this paper, all those N (25 even and odd) integer samples are used for filtering a predicted value for one sample in view 1.

C. Disparity Clustering for Depth Level Segmentation

In this sub-section, a method is proposed to segment a picture into different depth regions by the disparity motion vectors (DMVs). To get the DMVs, the corresponding block in the picture of view 0 is found by block matching. We utilize a simple 16x16 block matching algorithm, which reduces the complexity. It is applied for each MB of the view 1 picture and only the integer positions in view 0 are searched.

Given the number of desired depth levels *K* for a picture, a *K*-means algorithm is utilized to cluster the DMVs into *K* classes, by minimizing the following squared error function:

$$E = \sum_{i=1}^{K} \sum_{v_j \in V_i} \left\| v_j - \mu_i \right\| \qquad (2)$$

where the *K* clusters of DMVs are $V_i, i = 1 \cdots K$ and $\mu_i, i = 1 \cdots K$ is the centroid (or the mean) of all the DMVs $v_j \in V_i$. $\|.\|$ is the Euclidean norm.

The *K*-means problem is solved by Lloyd s algorithm [8], in which the centroids are initialized and then updated based on the following repeatedly executed steps:

1. Classify all DMVs into different clusters based on the current centroids. A DMV is classified to the cluster with the nearest centroid to this DMV.
2. Recalculate the centroids: $\hat{\mu}_i, i = 1 \cdots K$.
3. If the centroids are not changed, that is $\|\mu_i - \hat{\mu}_i\| < \varepsilon, \forall i = 1 \cdots K$, terminate the iteration; else, set $\mu_i = \hat{\mu}_i, i = 1 \cdots K$ and return to Step 1.

In this paper, ε is set to 1.

To address multiple depth levels, there are *K* sets of samples $\mathbf{S}^i, i = 1 \cdots K$ in a picture. After the clustering of the DMVs, segmentation is done to divide the picture into different depth regions, each of which contains a DMV being classified into the corresponding DMV cluster.

After the filters are obtained, they are applied to the inter-view picture to get multiple filtered reference pictures.

D. Relevant Regions Selection for the Adaptive Filters

The optimization problem, described in equation (1), seeks the least squared error solution for a specific sample set in a picture. As a matter of fact, view 1 is coded in a hybrid way, which enables not only inter-view prediction (that utilizes adaptive filters) but also conventional H.264/AVC (intra-view) modes: inter prediction and intra prediction. The MBs or MB partitions for which intra-view modes are selected cannot benefit from the adaptive filter. So allowing those MBs to be considered for the adaptive filter generation can lead to less optimal filters, which are less sensitive to the prediction errors of those samples finally predicted by inter-view prediction. The regions that incline to use the generated filters are the relevant regions for the adaptive filtering generation.

To get the relevant regions, consisting of chosen MBs, the following function is proposed to select the MBs.

$$f(MB_t) = \begin{cases} 1 & Distortion(MB_t) \le T \\ 0 & otherwise \end{cases} \qquad (3)$$

wherein $f(\cdot)$ equals to 1 indicates that the t-th MB is selected as relevant MB and *Distortion*(.) returns the MSE (Mean Square Error) distortion between the original signal and the predicted signal of an MB. So we have:

$$\mathbf{S}^i = \{ s_t^j \mid s_t^j \in MB_t, f(MB_t) = 1, v_t \in V_i \}, i = 1 \cdots K \qquad (4)$$

For simplicity, MB_t denotes the t-th MB in the picture in view 1. The threshold *T* is content dependent and it is decided as follows:

$$Rate = |D| / NumMB, D = \{ MB_t \mid f(MB_t) = 1 \} \qquad (5)$$

wherein $|\cdot|$ stands for the cardinality of a set and *NumMB* is the number of MBs in a picture of view 1. When the *Rate*, the percentage of MBs that are chosen for inter-view prediction, is set, the threshold *T* can be fixed by ordering the distortion values of all the MBs in a picture. Next section will give more detail on the *Rate* values.

IV. Implementation and Simulation

The proposed algorithm was implemented into the MVC reference software, JMVM (Joint Multiview Video Model) version 5 [9]. The tested sequences were *Exit*, *Ballroom (BR)*, *Rena*, *Race1*, *Akko&Kayo (AK)*, *Breakdancers (BD)* and *Flamenco2 (FL)*. For each test sequence, the first two views were selected to be coded as view 0 and view 1, respectively, in our simulation. The low-resolution input views were generated by utilizing the MPEG-4 downsampling filter, which is [2, 0, −4, −3, 5, 19, 26, 19, 5, -3, -4, 0, 2]/64. After decoding, the low-resolution pictures were upsampled by the H.264/AVC interpolation filter ([1, -5, 20, 20, -5, 1]/32) for PSNR calculation. Other parameters, e.g., the temporal prediction structure and the motion estimation search range followed the MVC common test conditions specified in [10].

In this paper, a fixed *Rate* value was used for all pictures in a sequence. The *Rate* values used are shown in Table I.

The rate distortion (RD) performances for the proposed regionally adaptive filtering (RAF) method, DMC [6] as well as the picture-level adaptive filter (PAF) proposed in [7] were compared and the results are listed in Table II. Note that, in the table, a positive bit-rate saving or a positive ΔPSNR value indicates that the algorithm on the left is better than the right one. The results were generated using the Bjontegaard measurements [11] based on the bit-rates and average PSNR values of the four test points corresponding to different QP values.

TABLE I. RATES FOR DIFFERENT TEST SEQUENCES

Sequence	Exit	BR	Rena	Race1	AK	BD	FL
***Rate* (%)**	75	55	85	60	70	90	80

TABLE II. COMPARISON BETWEEN RAF, DMC AND SIMULCAST (FOR VIEW 1)

Sequence	RAF vs DMC		RAF vs GAF	
	Bit-rate saving	*⊿PSNR*	*Bit-rate saving*	*⊿ PSNR*
Akko&Kayo	-2.56%	-0.092	3.61%	0.111
Ballroom	13.41%	0.307	4.66%	0.113
Exit	2.32%	0.042	2.02%	0.039
Race1	-0.47%	-0.008	6.02%	0.114
Rena	27.43%	0.776	0.68%	0.019
Breakdancers	14.08%	0.312	7.86%	0.177
Flamenco2	2.16%	0.081	1.07%	0.039
Average	8.05%	0.203	3.70%	0.087

The proposed method gives more than 10% bit-rate saving for three of the test sequences. The average bit-rate saving is 8% for the low-resolution view. As shown in Table II, on average RAF approximately doubled the bit-rate saving of PAF. RAF improved the compression efficiency for the sequences for which PAF performed poorly, i.e., *Race1*, *Akko&Kayo* and *Exit*. Meanwhile, it further increases the gain for the other sequences significantly, such as *Ballroom* and *Breakdancers*.

The RD curves of the four methods are shown in Fig. 3 for the *Breakdancers* sequence. As shown in Fig. 3, RAF is much better than the simulcast coding, this observation confirms that the proposed RAF based on DMC can achieve substantial bandwidth decrease compared to simulcast coding and make the whole bandwidth much closer to that of 2D mono-view H.264/AVC video. On top of the DMC, adaptive filtering algorithms provided extra gains. The bit-rate saving of RAF is more significant than that of PAF. Note that for PAF, we selected the best configuration in [7]. DMC outperforms simulcast with more than 1.3 dB [5], and the proposed RAF algorithm further increases the coding efficiency. So RAF on top of DMC can greatly reduce the bandwidth of the ASV.

V. DISCUSSION

RAF is suitable for those sequences that have multiple depth levels and provides better coding efficiency than PAF because it better adapts the distribution of the disparity motion vectors. However, it requires more reference pictures, which actually increases the decoder memory requirement. There is always a tradeoff between the resource consumption in the decoder and the bandwidth efficiency; this is especially the case here, as more depth levels require a larger memory size. This is the reason we select 2 or 3 depth levels.

VI. CONCLUSIONS

Asymmetric stereoscopic video coding has the second view coded in quarter resolution compared to the resolution of the H.264/AVC compliant base view, with unnoticeable subjective quality decreases. To further decrease the bandwidth of the low-resolution view, regionally adaptive filtering method was proposed to generate better predicted signal in inter-view prediction. The main motivation is based on the fact that the objects in a scene can have different depth levels thus different optimizations should be applied for different regions. By searching and classifying the disparity motion vectors, different regions can be segmented and get different filters. Compared with the direct motion compensation method without filtering, on average 8% and up to 27% bit-rate savings can be achieved. With the proposed method, stereoscopic video applications can be realized with minor bandwidth increase to the H.264/AVC based mono-view applications.

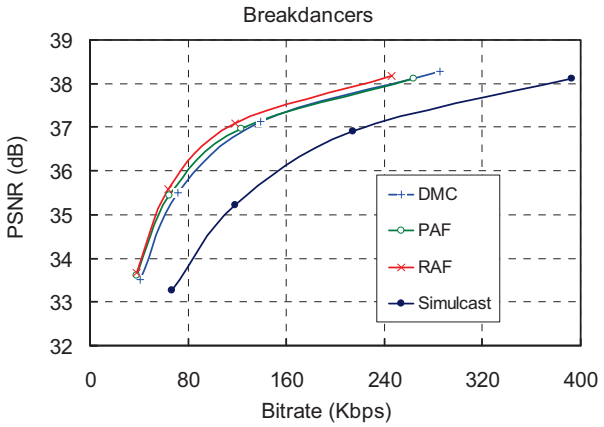


Figure 4. Rate distortion curves for the *Breakdancers* Sequence.

REFERENCES

[1] "Text of ISO/IEC 14496-10:200X/FDAM 1 Multiview Video Coding," ISO/IEC JTC1/SC29/WG11, Doc. W9978, Hannover, Germany, 2008.

[2] A. Vetro, W. Matusik, H. Pfister, J. Xin, "Coding Approaches for End-to-End 3D TV Systems," Picture Coding Symposium, 2004.

[3] Julesz B., Foundations of Cyclopean Perception, University of Chicago Press, Chicago, IL, USA, 1971.

[4] C. Fehn, P. Kauff, S. Cho, H. Kwon, N. Hur, J. Kim, "Asymmetric coding of stereoscopic video for transmission over T-DMB," Proc. 3DTV-CON 2007, Kos Island, Greece, May 2007.

[5] Y. Chen, S. Liu, Y.-K. Wang, M. M. Hannuksela, H. Li and M. Gabbouj, "Low complexity asymmetric multiview video coding," IEEE International Conference on Multimedia Expo, 2008.

[6] Y Vatis, B. Edler, D. T. Nguyen, J. Ostermann, "Motionand Aliasing-Compensated Prediction Using a Two-dimensional Non-separable Adaptive Wiener Interpolation Filter," IEEE International Conference on Image Processing, 2005.

[7] Y. Chen, Y.-K. Wang, M. M. Hannuksela, and M. Gabbouj, "Picture-level Adaptive Filter for Asymmetric Stereoscopic Video," IEEE International Confrence on Image Processing, 2008.

[8] S. P. Lloyd, "Least Squares Quantization in PCM," IEEE Transactions on Information Theory, vol. 28, no. 2, pp. 129-137, 1982.

[9] P. Pandit, A. Vetro, Y. Chen, "JMVM 5 software," JVT-X208, Geneva, Switzerland, Jun.-Jul. 2007.

[10] Y.Su, A. Vetro, A. Smolic, "Common Test Conditions for Multiview Video Coding," JVT-T207, Klagenfurt, Austria, Jul. 2006.

[11] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," VCEG-M33, Mar. 2001.

[P11] S. Tao, Y. Chen, M. M. Hannuksela, Y.-K. Wang, M. Gabbouj and H. Li, "Joint Texture and Depth Map Video Coding Based on the Scalable Extension of H.264/AVC," *IEEE International Symposium on Circuits and Systems, ISCAS'09,* Taipei, May 24-27, 2009, pp. 2353–2356.

Joint Texture and Depth Map Video Coding Based on the Scalable Extension of H.264/AVC

Siping Tao[1], Ying Chen[2], Miska M. Hannuksela , Ye-Kui Wang , Moncef Gabbouj[2], and Hou iang Li[1]

[1]University of Science and Technology of China
Hefei, China
anhuitsp mail.ustc.edu.cn
lihq ustc.edu.cn

[2]Department of Signal Processing, Tampere University of Technology
Tampere, Finland
ying.chen tut.fi
moncef.gabbouj tut.fi

[3]Nokia Research Center
Tampere, Finland
miska.hannuksela nokia.com
yekuiwang huawei.com

***Abstract*—Depth-Image-Based Rendering (DIBR) is widely used for view synthesis in 3D video applications. Compared with traditional 2D video applications, both the texture video and its associated depth map are required for transmission in a communication system that supports DIBR. To efficiently utilize limited bandwidth, coding algorithms, e.g. the Advanced Video Coding (H.264/AVC) standard, can be adopted to compress the depth map using the 4:0:0 chroma sampling format. However, when the correlation between texture video and depth map is exploited, the compression efficiency may be improved compared with encoding them independently using H.264/AVC. A new encoder algorithm which employs Scalable Video Coding (SVC), the scalable extension of H.264/AVC, to compress the texture video and its associated depth map is proposed in this paper. Experimental results show that the proposed algorithm can provide up to 0.97 dB gain for the coded depth maps, compared with the simulcast scheme, wherein texture video and depth map are coded independently by H.264/AVC.**

I. INTRODUCTION

Due to recent advances in acquisition and display technologies, 3D video has become a reality in consumer domain with different application opportunities, such as 3D TV [1]. When transmitting 3D content based on multiple representations of 2D videos, the bandwidth constraint is an important issue, thus a compressor is required to code 3D content even with only a reasonably small number of views. However, the rendering equipment may require simultaneous providing of many views, which e.g., is the case for auto-stereoscopic displays. In order to provide an immersive user experience, it is desirable to enable the decoder to render as many and continuous views as possible. View synthesis can satisfy these requirements, by transmitting a reasonable number of views while interpolating other views at the renderer. The Depth-Image-Based Rendering (DIBR) technique [2] is a typical view synthesis algorithm in a communication system. As the high computational complexity required for depth estimation [3] is typically not acceptable at the decoder, a depth map should be transmitted [4]. In so-called video plus depth applications [5], only one view (also named as texture video) with its associated depth map is required for view synthesis.

To utilize limited transmission bandwidth efficiently, texture video and depth map should be compressed before transmission. One standard-compliant way for compression of the texture video and the depth map is to compress them separately, by utilizing H.264/AVC. However, it is worth exploiting the correlation between the texture video and its associated depth map, to get higher coding efficiency. In [6], Grewatsh et al. proposed one MPEG-2 based method for compressing the depth map by reusing the motion vectors (MVs) of its corresponding texture video. Instead of reusing the texture video motion information without any modifications, a so-called Candidate Mode Generation process is used in [7], which can generate more accurate motion information for the depth map. Both of these cases do not need to transmit MVs for depth map. However, the previous methods have two drawbacks. First, although they are originated from MPEG-2 or H.264/AVC, they are not compliant with any existing standard. Second, they do not necessarily result in optimal compression efficiency. For example, sharing the texture video MVs with the depth map may increase the residual data when coding the depth map because the texture video MV may not be sufficiently accurate for its associated depth map. In this paper, to solve the above problems, we propose to utilize the inter-layer motion prediction tool in SVC, the scalable extension of H.264/AVC [8], to compress the texture video and its associated depth map jointly. Experimental results demonstrate that, compared with the simulcast scheme, the proposed method achieves on average 0.56 dB and up to 0.97 dB gain for the coded depth maps, which is equivalent to 22% of bit-rate reduction.

The rest of the paper is organized as follows. Section II introduces the inter-layer prediction tools in SVC. Section III analyzes the correlation between the texture video and its associated depth map. The details of the proposed method are presented in section IV. Section V describes the test scenarios and presents the simulation results, and Section VI concludes the paper.

The work of Houqiang Li and Siping Tao is partially supported by NSFC General Program under contract No. 60572067, NSFC General Program under contract No. 60672161, and NSFC Key Program under contract No. 60736043. The work of Ying Chen is partly supported by the Nokia Foundation Award granted by Nokia Research Center (NRC). The work of Ye-Kui Wang was carried in NRC and he is currently with Huawei Technologies, Bridgewater, NJ, USA.

978-1-4244-3828-0/09/$25.00 ©2009 IEEE

II. INTER-LAYER PREDICTION IN SVC

The Scalable Video Coding (SVC) extension of the H.264/AVC standard has been developed by the Joint Video Team (JVT). The layered coding approach is employed in SVC, wherein more than one dependency layer can be presented and each layer is identified by a dependency identifier. The layer with the lowest dependency identifier value is called the base layer and the other layers are called enhancement layers in this paper. To remove the redundancy between different spatial or quality layers, three inter-layer prediction tools are used, namely motion prediction, intra texture prediction, and residual prediction. The inter-layer motion prediction tool is described in the next paragraph. For an introduction to the other inter-layer prediction tools, which are not employed in the proposed method, please refer to [8].

Co-located macroblock (MB) in the base layer can be used to derive predictors for the MVs in an MB of an enhancement layer. It is called inter-layer motion predictor. For dyadic spatial scalability case, the base layer MVs need to be simply scaled. While for Coarse Granularity Scalability (CGS), an MV in the base layer can be directly used as the predictor. Note that an MV difference can be signaled for each MB or MB partition to further refine the inter-layer motion predictor to a better one in terms of e.g., rate-distortion performance.

III. MOTION INFORMATION CORRELATION BETWEEN TEXTURE VIDEO AND DEPTH MAP

A texture video consists of three components, namely one luma component Y, and two chroma components U and V, whereas the depth map only has one component representing the distance between the object pixel and the camera. Generally, a texture video is represented in YUV 4:2:0 format and a depth map is regarded as luma-only video in YUV 4:0:0 format. Fig. 1 is an example of a texture video frame and its associated depth map. Obviously, the color information of a pixel and its distance from the camera are less relevant. However, from Fig. 1, it can be observed that both the texture video and its associated depth map have similar object silhouette, so they should have similar object boundary and movement. To confirm this observation, the following experiment was performed.

Figure 1. A texture video frame and its associated depth map.

The texture video and its associated depth map were coded with H.264/AVC, and their motion fields in the unit of the 4x4 block were extracted. Let $\vec{t}$ and $\vec{d}$ be the motion field of a specific frame of the texture video and its associated depth map, respectively. The correlation coefficient between the two motion fields is calculated as

$$\rho_i = \frac{Cov(\vec{t},\vec{d})}{\sqrt{Var(\vec{t})Var(\vec{d})}}, \quad (1)$$

wherein

$$Cov(\vec{t},\vec{d}) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1}[(\vec{t}_{m,n}-\vec{t}_{avg})\cdot(\vec{d}_{m,n}-\vec{d}_{avg})], \quad (2)$$

$$Var(\vec{t}) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\left\|\vec{t}_{m,n}-\vec{t}_{avg}\right\|^2, \quad (3)$$

$\vec{t}_{avg}$ is the average MV of $\vec{t}$, $\vec{d}_{avg}$ is that of $\vec{d}$, M/N is the picture height/width divided by 4, " " denotes the inner product, and denotes the norm operator. The correlation coefficients were calculated for 100 frames in the *Ballet* test sequence, and the curve of correlation coefficient per frame is plotted in Fig. 2. It can be observed that, the texture video motion field and its associated depth map motion field are well correlated. Therefore, coding efficiency would be improved if one can efficiently use this correlation. In the next section, a new method of compressing texture video and depth map jointly using SVC is proposed.

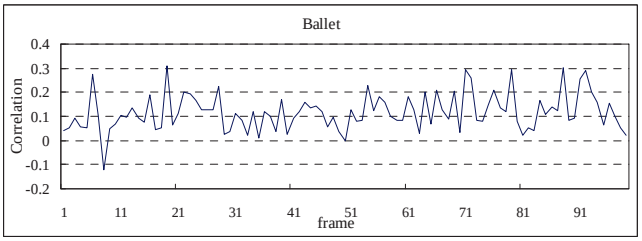


Figure 2. Correlation coefficient of motion fields per frame.

IV. TEXTURE VIDEO AND DEPTH MAP COMPRESSION USING SVC

In the proposed method, two layers are coded: the texture video is coded as the base layer, and the depth map is coded as the CGS enhancement layer. The texture video is coded using the same mechanism as H.264/AVC single layer coding, while the depth map is coded using SVC inter-layer motion prediction in addition to single layer coding techniques. It should be noted that the other two inter-layer prediction tools are disabled in the proposed method. In the depth map motion estimation process, the conventional spatial MV predictor or the inter-layer MV predictor can be chosen for each MB in the enhancement layer. Furthermore, when the co-located MB in the base layer is inter coded, the MB in the enhancement layer can adaptively choose the base mode in addition to the conventional H.264/AVC modes in the mode decision process. After motion estimation and mode decision, transform and quantization are applied to the enhancement layer as in SVC.

The detailed enhancement layer mode decision process for the inter frame coding is illustrated in Fig. 3. When the co-located MB in the base layer is intra coded, inter and intra modes without inter-layer prediction will be checked. Otherwise, the base mode without residual prediction will be checked first, then the inter modes using inter-layer motion

predictors (without residual prediction) are checked next, as well as inter and intra modes without inter-layer prediction.

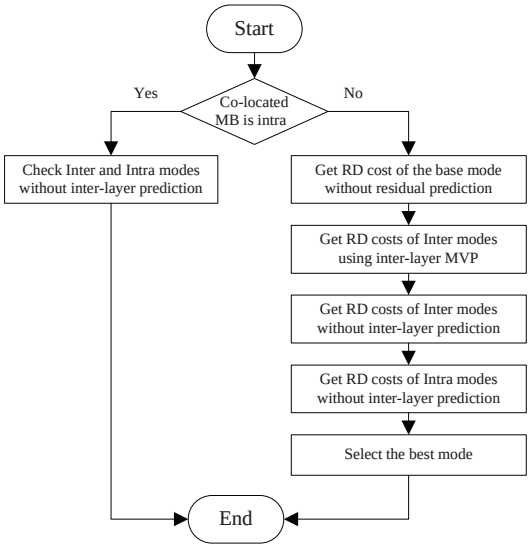


Figure 3. Enhancement layer mode decision process.

V. SIMULATIONS

The proposed algorithm was implemented in SVC reference software JSVM 9 13 [9], and the simulation conditions were as follows:

- IPPP coding structure and hierarchical B picture coding structure with GOP size equal to 16 were tested separately.
- 100 frames were encoded.
- Spatial / temporal resolution: 1024x768 15Hz.
- The CABAC entropy coding method was used.
- Intra picture refresh was turned off.
- 8x8 transform was turned on.
- The base layer QP (QP0) varies in 24, 28, 32, 36 . The enhancement layer QP (QP1) was selected to result in depth map bit-rate approximately in the range of 10% to 20% compared with the bit-rate of the texture video, which has been considered to provide sufficient quality for DIBR [10].

Two sequences, namely *Ballet* and *Breakdancers* [11], were tested. These two sequences were selected due to the fact that accurate depth maps were available for them.

The following four methods were tested in the simulations:

- The proposed method as described in section IV.
- Simulcast: texture video and depth map were independently compressed.
- All inter-layer prediction (AP): the proposed method as described in section IV but additionally all the inter-layer prediction tools were used adaptively.
- Forced motion prediction (FP): inter-layer motion prediction (without motion refinement) was always used for the depth maps, whereas the other inter-layer prediction tools and spatial/temporal MV prediction were disabled for the depth maps.

The FP method is similar to the method in [6]. Because the texture video MV may not be accurate for its associated depth map, both of the methods in [6] and [7] have worse efficiency than Simulcast in medium and high bit-rates, and the overall efficiency improvement is limited. In contrast, the simulation results showed that the efficiency gain of the proposed method is consistent over all the bit-rates.

In Table I, the comparison of the proposed method and Simulcast is presented. It can be observed that the proposed method offers up to 0.97 dB gain in terms of depth map PSNR compared with Simulcast, and 0.56 dB gain on average. Fig. 4 shows the RD curves of the above four methods. It can be found that the proposed method has similar performance as AP; this is because inter-layer texture prediction and residual prediction have little contribution to improve the coding efficiency. However, comparing with AP, the proposed method does not need to check the modes with residual prediction, as well as the modes with texture prediction. Note that those are about half of the modes tested in JSVM and requires a substantial large part of the encoding computations in the reference JSVM encoder. Therefore, the proposed method has much lower encoding complexity than AP. From Fig. 4, we can see that the FP method has the worst performance among the tested methods, about 3.6 dB loss on average in terms of depth map PSNR compared with Simulcast. The cause for the inferior performance of the FP method is illustrated in Fig. 5, which presents the percentage of MBs using inter-layer motion prediction for the proposed method. The percentage of MBs using inter-layer motion prediction is about 10%. Thus, nearly 90% of the MB modes of the depth maps are not optimal in the FP method.

TABLE I. COMPARISON OF THE PROPOSED METHOD AND SIMULCAST.

Sequence (GOP)	Proposed vs. Simulcast Gain	
	PSNR B	***Bitrate***
Ballet (16)	0.81	-20.81
Breakdancers (16)	0.36	-7.72
Ballet (1)	0.97	-21.87
Breakdancers (1)	0.08	-1.59

VI. CONCLUSIONS

In this paper, a new coding algorithm was proposed for joint texture and depth map video coding. The method encodes texture video as the base layer of a scalable video bitstream, and depth map as the enhancement layer. Inter-layer motion vector prediction is adaptively used to improve the compression efficiency compared with coding texture video and depth map as two independent bitstreams. The proposed method complies with the Scalable Video Coding (SVC) standard, hence facilitating the reuse of SVC implementations for Depth-Image-Based Rendering. Moreover, the fact that the base layer remains H.264/AVC compliant enables phased introduction of depth maps into existing H.264/AVC-based services, because devices that do not have 3D functionality

can decode the base layer to provide a conventional 2D video sequence. Simulation results showed that compared with simulcast coding of texture video and depth map, the proposed method can bring significant coding gain for the associated depth map.

REFERENCES

[1] L. Onural, T. Sikora, and A. Smolic, "An overview of a new European consortium: integrated three-dimensional television–capture, transmission and display (3D TV)," Proc. European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology (EWIMT), London, UK, Nov. 2004.

[2] W. R. Mark, "Post-rendering 3-D image warping: visibility, reconstruction, and performance for depth-image warping," PhD thesis, University of North Carolina at Chapel Hill, NC, USA, 1999.

[3] N. Fukushima, T. Yendo, T. Fujii and M. Tanimoto, "Free viewpoint image generation using multi-pass dynamic programming," Proc. SPIE Stereoscopic Displays and Virtual Reality Systems XIV, vol 6490, pp. 460-470, Feb. 2007.

[4] "Description of exploration experiments in 3D video coding," ISO/IEC JTC1/SC29/WG11, Doc. W9991, Hannover, Germany, 2008.

[5] C. Fehn, P. Kauff, M. Op de Beeck, F. Ernst, W. IJsselsteijn, M. Pollefeys, L. Van Gool, E. Ofek, and I. Sexton, "An evolutionary and optimized approach on 3D-TV," Proc. IBC 2002, pp. 357-65, Amsterdam,. Netherlands, Sept. 2002.

[6] S. Grewatsh and E. Muller, "Sharing of motion vectors in 3D video voding," Proc. IEEE International Conference on Image Processing, vol 5, pp. 3271-3274, Oct. 2004.

[7] H. Oh and Y. S. Ho, "H.264-based depth map sequence coding using motion information of corresponding texture video," Spinger Berlin/Heidelberg, Advances in Image and Video Technology, vol. 4319, pp. 898-907, 2006.

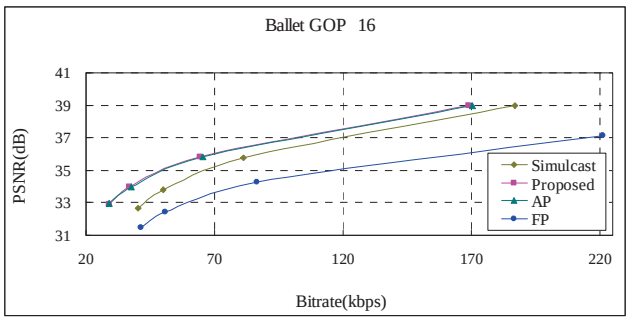
[8] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," IEEE Trans on Circuits and Systems for Video Technology, vol. 17, No. 9, pp. 1103-1120, Sept. 2007.

[9] J. Reichel, H. Schwarz, and M. Wien, Joint Scalable Video Model 11 (JSVM 11), Joint Video Team, Doc. JVT-X202, Jul. 2007.

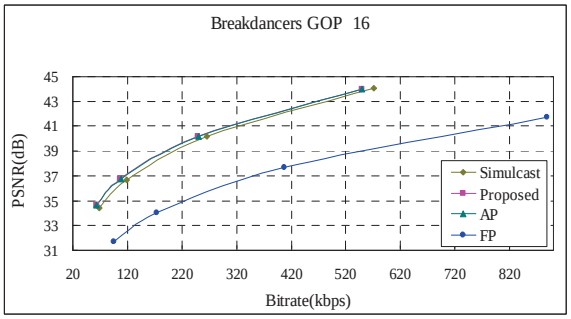
[10] A. Smolic, K. Mueller, N. Stefanoski, J. Ostermann, A. Gotchev, G. B. Akar, G. Triantafyllidis, and A. Koz, "Coding algorithms for 3DTV-a survey," IEEE Trans. On Circuits and Systems for Video Technology, vol. 17, no. 11, pp. 1606-1621, Nov. 2007.

[11] C.L. itnick, S.B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, High-quality video view interpolation using a layered representation, ACM SIGGRAPH and ACM Trans. on Graphics, Los Angeles, CA, pp. 600-608, Aug. 2004.

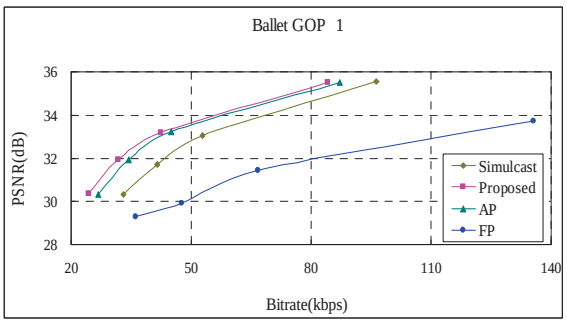
[12] G. Bjontegaard, Calculation of average PSNR differences between RD-curves, VCEG-M33, Mar. 2001.



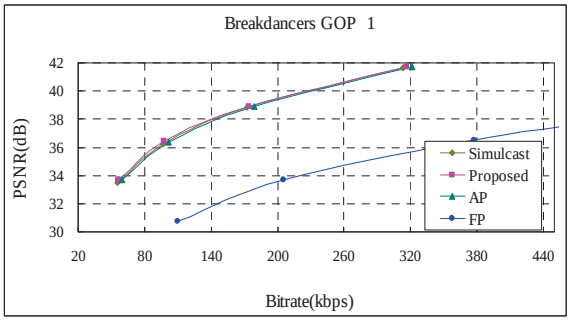
(a) *Ballet* with GOP 16



(b) *Breakdancers* with GOP 1



(c) *Ballet* with GOP 1



(d) *Breakdancers* with GOP 1

Figure 4. RD performance of the proposed method, Simulcast,AP and FP.

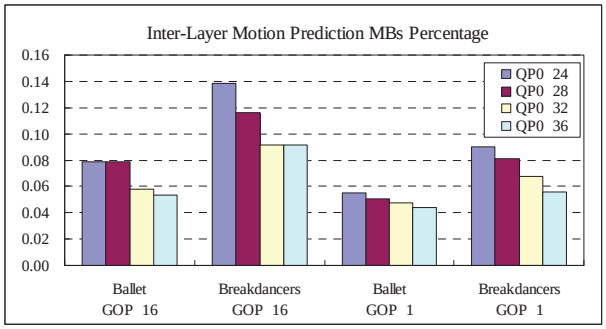


Figure 5. Percentages of MBs using inter-layer motion prediction.