www.zte.com.cn/magazine/English

ISSN 1673–5188
CODEN ZCTOAK

ZTE COMMUNICATIONS

June 2012, Vol.10 No.2

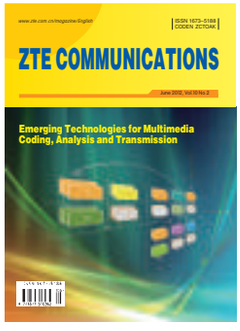
Emerging Technologies for Multimedia Coding, Analysis and Transmission

ISSN 1673-5188
06>
9 771673 518062

ZTE中兴 ZTE COMMUNICATIONS
VOLUME 10 NUMBER 2 JUNE 2012

CONTENTS

Http://www.zte.com.cn/magazine/English
Email: magazine@zte.com.cn



EDITORIAL BOARD

Cheng–Zhong Xu
Wayne State University (USA)

Houlin Zhao
International Telecommunication Union (ITU)

Ke–Li Wu
The Chinese University of Hong Kong (China)

Shiduan Cheng
Beijing University of Posts and Telecommunications (China)

Wen Gao
Peking University (China)

Zhengkun Mi
Nanjing University of Posts and Telecommunications (China)

Special Topic

Emerging Technologies for Multimedia Coding, Analysis and Transmission

Department

ZTE COMMUNICATIONS
Vol. 10 No.2 (Issue 34)
Quarterly
First Issue Published in 2003

Supervised by:
Anhui Science and Technology Department

Sponsored by:
ZTE Corporation and Anhui Science and Technology Information Research Institute

Staff Members:
Editor–in–chief: Xie Daxiong
Associate Editor–in–chief: Zhao Jinming
Executive Associate Editor–in–chief: Huang Xinming
Editor in Charge: Zhu Li
Editors: Paul Sleswick, Xu Ye, Yang Qinyi, Lu Dan
Producer: Yu Gang
Circulation Executive: Wang Pingping
Assistant: Wang Kun

Editorial Correspondence:
Add: 12/F Kaixuan Building,
329 Jinzhai Road,
HeFei 230061, P. R. China
Tel: +86–551–5533356
Fax: +86–551–5850139
Email: magazine@zte.com.cn

Published and Circulated (Home and Abroad) by:
Editorial Office of ZTE COMMUNICATIONS

Printed by:
Hefei Zhongjian Color Printing Company

Publication Date:
June 25, 2012

Publication Licenses:
ISSN 1673–5188
CN 34–1294/TN

Advertising License:
皖合工商广字0058号

Annual Subscription Rate:
USD$50

Responsibility for content rests on authors of signed articles and not on the editorial board of ZTE COMMUNICATIONS or its sponsors.
All rights reserved.

Guest Editorial of Emerging Technologies for Multimedia Coding, Analysis and Transmission

Huifang Sun and Dong Wang

Emerging Technologies for Multimedia Coding, Analysis and Transmission

Huifang Sun

Dong Wang

Over the past two decades, significant progress has been made in the coding, analysis, and transmission of digital audio, video, and images. Digital multimedia signal processing technologies have arisen out of practical necessity and have significantly affected the multimedia industry. This special issue contains nine papers written by experts from academia and industry. The papers detail the most recent achievements in audio and video coding analysis and transmission and give an overview of emerging technologies.

The paper by Ping Wu et al. introduces high–efficiency video coding (HEVC). This new video coding standard greatly improves on the coding efficiency of H.264/AVC and is a milestone in digital video coding standards. In this paper, the technical features of HEVC (up to HEVC CD stage) are discussed.

The paper by Yichen Zhang and Lu Yu discusses the ongoing standardization of MPEG's 3D video (3DV) coding. In this paper, coding tools proposed in response to MPEG's Call for Proposals on 3DV coding are summarized.

The paper by Siwei Ma et al. gives an overview of the 3DV coding standard developed by the China Audio Video Coding Standard (AVS) Working Group.

The paper by Euee S. Jang describes a configurable codec framework. Video codec devices are becoming increasingly complex because of a wider range of applications. MPEG suggests tackling the problem with a reconfigurable video coding (RVC) framework and by standardizing modular definitions of tools and their connections. This paper gives a comprehensive overview of the RVC standard.

The paper by Minjie Xie describes lattice vector quantization (LVQ) algorithm for speech and audio coding. Various LVQ schemes have been developed for speech and audio coding, and some of these have been successfully used in ITU–T G.718 and G.719, 3GPPAMR–WB+, and MPEG USAC.

In the paper by Stephane Ragot et al., noise feedback coding (NFC) NFC is reviewed, and a novel coding technique is proposed. The technique involves using noise shaping in embedded pulse code modulation (PCM) and adaptive differential PCM (ADPCM).

The paper by Gerard Fernando describes the newest multimedia transport (MMT) standard, developed by MPEG. This paper describes the architecture and functions of MMT up to the MMT CD stage, and it lists the advantages and shortfalls of existing technologies.

The paper by Zhijie Zhao et al. describes low–complexity error resilience and error concealment for the scalable extension of H.264/AVC (SVC). This paper describes multiple description coding (MDC) for the encoder. It also describes an error concealment method in network abstraction layer (NAL) for decoder when medium grain scalability (MGS) is used. This method requires minimal computation and is suitable for real–time video streaming. In this paper, experimental results are also given.

The paper by Ling–Yu Duan et al. introduces the MPEG standard on compact descriptors for visual search (CDVS) for mobile applications. MPEG is developing a competitive and collaborative platform for evaluating existing visual search technologies. They are also developing a standard for visual descriptors.

Biographies

Huifang Sun (hsun@merl.com) received his BSc degree from Harbin Military Engineering Institute, China, and his PhD degree from the University of Ottawa, Canada. In 1990, he was an associate professor at Fairleigh Dickinson University. Also in 1990, he joined Sarnoff Corporation and was later promoted to technology leader. In 1995, he joined Mitsubishi Electric Research Laboratories and was promoted to vice president, deputy director, and fellow (2003). He has co–authored two books and published more than 140 journal and conference papers. He holds more than 60 US patents. In 1994, Huifang Sun received a Technical Achievement Award for optimization and specification of the Grand Alliance HDTV video compression algorithm. In 1992, he won the Best Paper award from *IEEE Transaction on Consumer Electronics*. In 1996, he won the Best Paper award at ICCE, and in 2003, he won the Best Paper award from IEEE Transactions on CSVT. He has been associate editor of *IEEE Transaction on Circuits and Systems for Video Technology* and was the chair of the Visual Processing Technical Committee of IEEE's Circuits and System Society. He is an IEEE Fellow.

Dong Wang (wang.dong@zte.com.cn) received his MSc degree from Southeast University in 2000. He has worked for ZTE Corporation since 2001 and is currently the multimedia standards director. He is responsible for pre–research and standardization activities on multimedia applications, system, coding, and transport. He is the vice chairman of the Study Group for Broadband Cable TV (ITU–T SG9). His research interests include the multimedia applications, systems and the next generation audiovisual coding and the next generation multimedia transport technologies, which are related to the standardization activities of ITU–T SG9&SG16, ISO/IEC JTC1 SC29 WG11 (MPEG) and 3GPP SA4.

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P4

Introduction to the High–Efficiency Video Coding Standard

Ping Wu and Ming Li
(R&D Center, ZTE Corporation, Nanjing 210012, China)

Abstract

The high–efficiency video coding (HEVC) standard is the newest video coding standard currently under joint development by ITU–T Video Coding Experts Group (VCEG) and ISO/IEC Moving Picture Experts Group (MPEG). HEVC is the next–generation video coding standard after H.264/AVC. The goals of the HEVC standardization effort are to double the video coding efficiency of existing H.264/AVC while supporting all the recognized potential applications, such as, video telephony, storage, broadcast, streaming, especially for large picture size video (4k × 2k). The HEVC standard will be completed as an ISO/IEC and ITU–T standard in January 2013. In February 2012, the HEVC standardization process reached its committee draft (CD) stage. The ever–improving HEVC standard has demonstrated a significant gain in coding efficiency in rate–distortion efficiency relative to the existing H.264/AVC. This paper provides an overview of the technical features of HEVC close to HEVC CD stage, covering high–level structure, coding units, prediction units, transform units, spatial signal transformation and PCM representation, intra–picture prediction, inter–picture prediction, entropy coding and in–loop filtering. The HEVC coding efficiency performances comparing with H.264/AVC are also provided.

Keywords

HEVC; JCTVC; AVC; H.264; MPEG–2; MPEG–4; standards; video

1 Introduction

The high–efficiency video coding (HEVC) standard is the newest video coding standard and is undergoing the development in the Joint Collaborative Team on Video Coding (JCT–VC) of ITU–T SG16 WP3 Video Coding Expert Group (VCEG) and ISO/IEC JTC1/SC29/WG11 (also known as Moving Picture Expert Group (MPEG)) [1]. HEVC aims to be the next generation video coding standard after the existing H.264/AVC (Advanced Video Coding–ISO/IEC 14496–10) [2]. It is well known that H.264/AVC is superior to MPEG–2 in video coding efficiency by 50%, i.e. similar quality but half the bitrate. The target of HEVC is to double the video coding efficiency of H.264/AVC [3]. The quality is mainly on subjective quality, while some peak signal–to–noise ratio (PSNR) based objective quality assessment methods are also employed during the standardisation process. The HEVC standardisation process reached Committee Draft (CD) stage in February 2012, and will become International Standard (ISO/IEC) by July 2013. However, usually when a CD stage is reached, the general form of a video coding standard will be stable.

Video coding standardisation for telecommunications has evolved through ITU–T H.261 [4], H.262 (MPEG–2) [5], H.263 (with its later enhancements of H.263+ and H.263++) [6] and H.264/AVC [2]. The video coding standards, e.g. MPEG–2 and H.264/AVC, are now widely used for the transmission of standard–definition (SD) and high–definition (HD) TV signals over satellite, cable, and terrestrial emission and for the storage of high–quality video signals on ordinary mediums, such as DVDs, blue–ray disks, hard disks and so on. The ever–growing demands of high–definition 4K × 2K video with better resolution and wider colour gamut as well as flexible streaming in various application scenarios continually raise the requirements of higher video coding efficiency with a flexible coding structure arrangement, which have been well covered in HEVC standardisation process.

The Call for Proposals on HEVC [1] was finally issued in January 2010. The test model under consideration (TMuC) was established three months later [7]. In October 2010, the first HEVC test model (HM) was created [8]. After 5 versions of working drafts (WD) [9]–[13] and HM reference software, the HEVC codec has been improved continuously, which ensures a matured design for HEVC CD stage.

In this paper, the main structure and the video coding tools in HEVC will be described in details which will be accurate at least close to HEVC CD stage. The remainder of this paper is organized as follows. In section 2, some applications and HEVC high–level structures will be briefly presented. In

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P5

section 3, the major features in HEVC design are described. Performance comparisons between the draft HEVC standard and H.264/AVC High Profile is reported in section 4. Finally, section 5 concludes this paper.

2 Applications and High-Level Structures

As with previously successful video coding standards, the HEVC standard is designed to provide technical solutions for at least the following application areas:

- cable TV (CATV) over optical and copper networks
- direct broadcast satellite (DBS) video services
- digital subscriber line (DSL) video services
- digital terrestrial television broadcasting (DTTB)
- interactive storage media (ISM), for example, optical disks
- multimedia mailing (MMM)
- multimedia services over packet networks (MSPN)
- real-time conversational (RTC) services, for example, video conferencing and video phone
- remote video surveillance (RVS)
- serial storage media (SSM), for example, digital VTR.

All these applications may be deployed in existing and future networks, which raises the question of how to handle a variety of applications and networks. To address this requirement for flexibility the HEVC design covers a video coding layer (VCL) as well as a network abstraction layer (NAL), which is the same as the layer structure in H.264/AVC. In the high level structure of an HEVC encoder, NAL is located below VCL to provide "network friendliness" to support simple and effective customization of the use of the VCL for a broad variety of systems, where the similar concepts to the counterparts in H.264/AVC, such as NAL unit, access unit, etc., are used. In VCL, the same concepts and applications of sequence parameter set (SPS) and picture parameter set (PPS) are adopted by JCT-VC into HEVC to convey the information which rarely changes and is referred to in decoding a large number of VCL NAL units.

To achieve high coding efficiency, HEVC introduces several picture-level coding tools, including scaling list [14], sample adaptive offset (SAO) [15], and adaptive loop filter (ALF) [16], whose parameters may stay the same when coding the slices in a picture but change among pictures. To share such information among slices effectively to support processing slices in parallel while facilitating the updating and referring to the tool parameters, adaptation parameter set (APS) [17] is designed and introduced to HEVC. As a new parameter set used for picture adaptive data (especially ALF data), APS forms a major feature in HEVC parameter set structure.

In an HEVC codec, each coded picture is represented in block-shaped units of associated luma and chroma samples called coding units (CUs) [18], [19]. The sizes of the largest CU (LCU) and the smallest CU (SCU) can be flexibly set in SPS, which is different from the concept of macroblock (MB) of a fixed number of 16 × 16 square pixels in the prior standards. A quad-tree based recursive splitting approach [18], [19] is used to partition the LCU until the partitions reach SCU size. The basic source coding algorithm in HEVC is a hybrid intra- and inter-picture prediction exploiting spatial and temporal statistical dependencies and transform coding of the prediction residual removing spatial statistical redundancies left in residuals after prediction. A unified entropy coding method, i.e. context-based adaptive binary arithmetic coding (CABAC) similar to that in H.264/AVC [20], is employed to generate the coded bit-streams.

To deal with the constraints of the maximum transmission unit (MTU) of the network, slices are introduced into HEVC. In certain configurations, a slice can be used as an independent decoding unit to provide error resilience and parallel processing with the help of parameter sets of information sharing. In HEVC, a slice usually contains a sequence of LCUs in raster scanning order, but it can also be non-LCU-aligned. A leaf-CU-aligned slice application with leaf-CU granularity of no less than 16 × 16 can be configured in SPS, which makes the encoder much more flexible in terms of implementing slices for transmission and error resilience [21].

Besides slices, tiles have also been developed so that HEVC can support MTU matching, error resilience, and parallel processing [22]. Intersecting column and row boundaries divide a picture into rectangular regions called tiles each containing an integer number of LCUs. The raster scanning order is applied in coding tiles within a picture and then LCUs within a tile. Similar to slice boundaries, tile boundaries also have an ability to break prediction mechanisms (e.g., intra prediction and motion vector (MV) prediction) pending tiles configurations unless indicated otherwise. Therefore, tiles are able to support sub-picture based coding when processing high resolution video (e.g. ultra high definition (UHD) video).

Although slices and tiles with proper configurations can provide parallel processing to the codec, they always lead to performance loss because of the boundary restrictions on the prediction mechanisms. To keep coding efficiency while obtaining somewhat parallel processing features, entropy slices [23] and wavefront parallel processing (WPP) [24] are integrated in HEVC. These two methods both aim at entropy coding in parallel, while utilizing the available information from adjacent blocks for high-efficiency prediction. The boundaries of an entropy slice only force flushing and re-initializing the entropy engine, which is also one of the functionalities of an ordinary slice. WPP is designed for multi-core architectures, which is believed to become more and more widely used on a variety of devices including mobile terminals. WPP initializes the CABAC probabilities of the first LCU of each line with the probabilities updated after the second LCU of the upper line has been processed, which carries out parallel processing of LCU lines but with two LCUs' delay among neighbouring LCU lines.

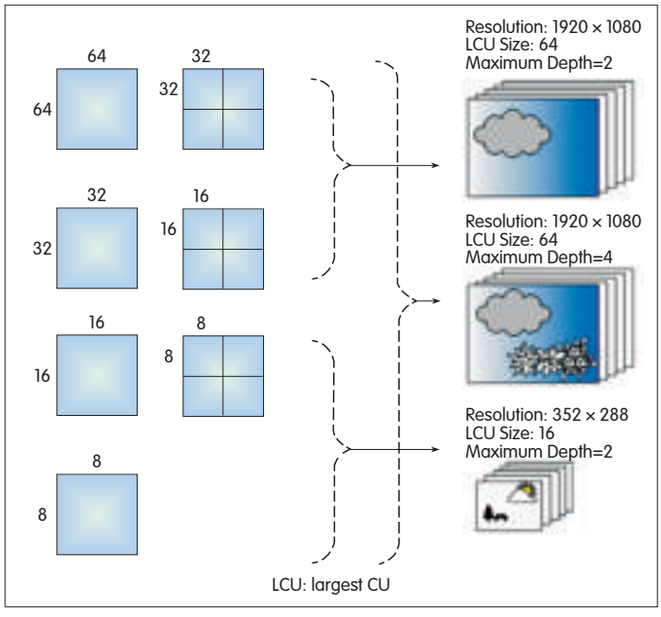
3 Major Coding Features

The HEVC standard adopts the well-known block-based hybrid coding scheme that relies on motion-compensated prediction and transform coding with high-performance

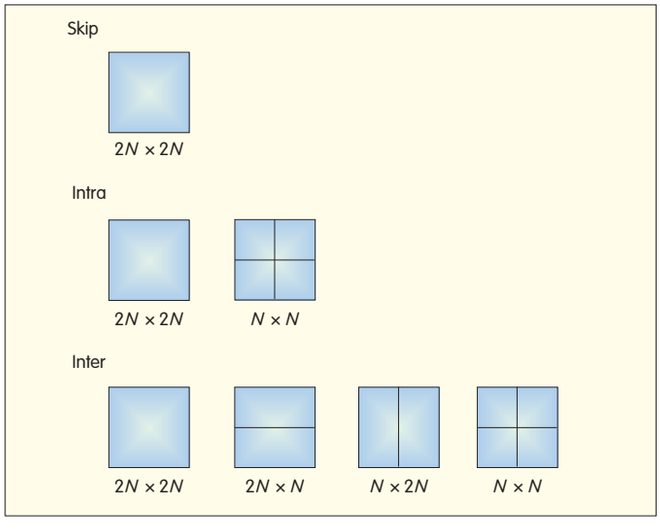
D:\EMAG\2011-02-29/VOL9\CONTETN.VFT——1PPS/P6

Introduction to the High–Efficiency Video Coding Standard

Ping Wu and Ming Li



▲Figure 1. CU splitting.



▲Figure 2. Symmetric PU partitions.

entropy coder. However, a series of newly developed techniques and modules are applied to this traditional structure in HEVC to make an accumulation of the highly achieved improvement in coding efficiency.

In this section, the major features of HEVC are briefly described from a perspective of encoding. And most of the decoding algorithms are already the reverse processes of the encoding counterparts, which are elaborated in the latest working draft document for HEVC [13].

3.1 Coding Units

CU is the basic unit of region splitting used for inter/intra coding. It is always square and may take a size from 8 x 8 luma samples up to the size of the LCU whose size can be set to 64 × 64, 32 × 32 or 16 × 16. A quad–tree based recursive splitting approach is designed to partition an LCU into four equally sized blocks with the limitation of an SCU with its minimum allowable size of 8 × 8. A parameter of maximum splitting depth refers to the quad–tree depth from LCU to SCU. By using this mechanism, a picture can be flexibly partitioned to meet the characteristics of the input video, as illustrated in Fig. 1.

3.2 Prediction Units

Prediction unit (PU) is the basic unit used to carry information related to prediction processes. Each CU may contain one or more PU. In general, PU is not restricted to being square in shape, in order to facilitate partitioning which matches the boundaries of real objects in the picture. Generally, all kinds of PU partitions can be employed in inter–picture prediction, while only square partitions are used in intra prediction. Besides the symmetric PU partitions dividing a CU into two or four blocks of equal size, four asymmetric PU partitions are also adopted by JCT–VC into HEVC to further improve the inter–picture prediction performance. The PU partitions in HEVC are presented in Figs. 2 and 3. Note that $N \times N$ PU partition is only applied to SCU.

3.3 Transform Units

Transform unit (TU) is the basic unit used for the transform and quantization processes. TU shape depends on PU partitioning mode. When PU is square, TU is also square, and TU may range in sizes from 4 × 4 to 32 × 32 luma samples. When the PU is not square, TU may be non–square with its sizes of 32 × 8, 8 × 32, 16 × 4, or 4 × 16 luma samples. Each CU may contain one or more TUs, and multiple TUs may be arranged in a quad–tree structure. The recursive quad–tree transform (RQT) [16] and non–square quad–tree transform (NSQT) [25] can be used in TU splitting as depicted in Figs. 4 and 5.

3.4 Spatial Signal Transformation and Quantization

Similar to the 4 × 4 and 8 × 8 transforms in H.264/AVC, the core transforms in HEVC [26] are also derived from discrete cosine transform (DCT) with integer precision and operations from 4 × 4 to 32 × 32. The core transform designs in HEVC

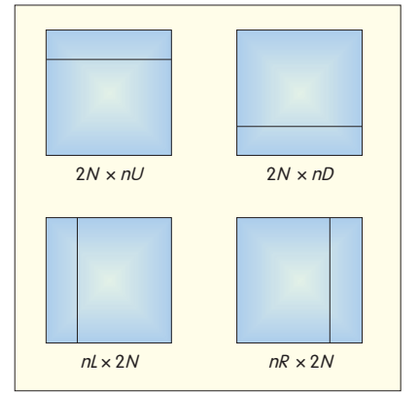
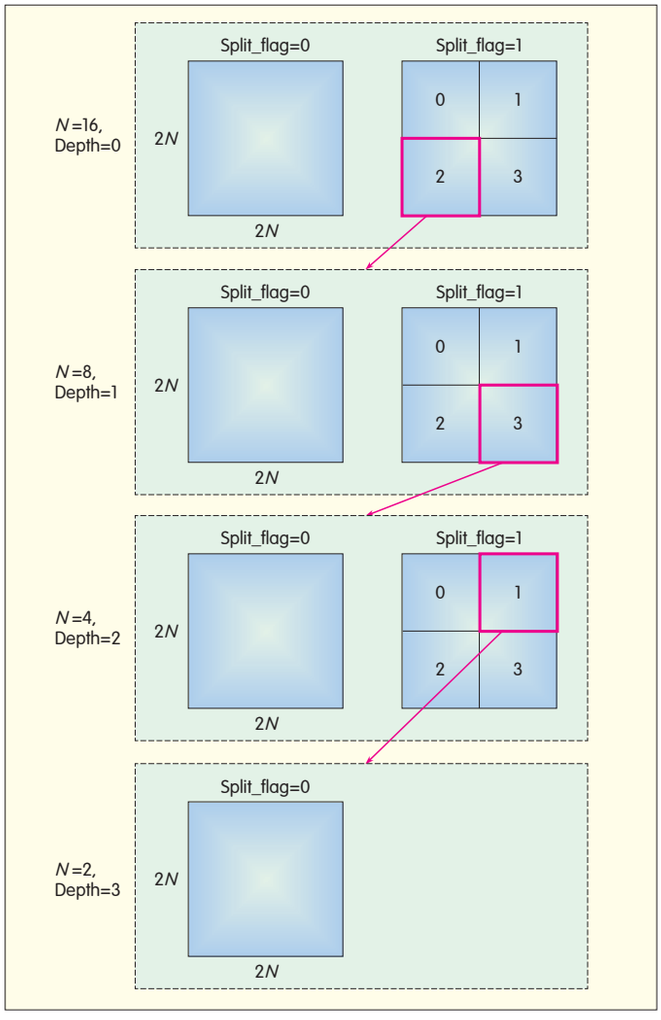


Figure 3. ▶ Asymmetric PU partitions.

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P7



▲Figure 4. Recursive quad–tree transform.

have many advanced properties for software and hardware implementation. For instance, it has 16 bit data representation (independent of the internal bit depth) before and after each transform stage. It also has 16 bit multipliers for all internal multiplications. There is no need for correction of different norms of basis vectors during quantization/dequantization. It allows reusing arithmetic operations for smaller transform sizes. It also allows implementations to use either pure matrix multiplication or a combination of matrix multiplication and feasible butterfly structures.

As in H.264/AVC, a scalar quantization method is also employed in HEVC after transform and can be implemented together with integer transform in an integrated module.

3.5 PCM Representation

The PCM representation in HEVC transmits sample values of its associated CU without prediction, transform coding and entropy coding. Thus, the PCM representation allows an encoder to adjust the number of bits of a CU or less without complicated computation, which is similar to the PCM mode in H.264/AVC. However, considering the features of CU structure, a more sophisticated method of PCM–mode coding is designed and implemented in the HEVC codec restricting the signalling of PCM mode flag in the bit–stream based on the CU splitting information [27].

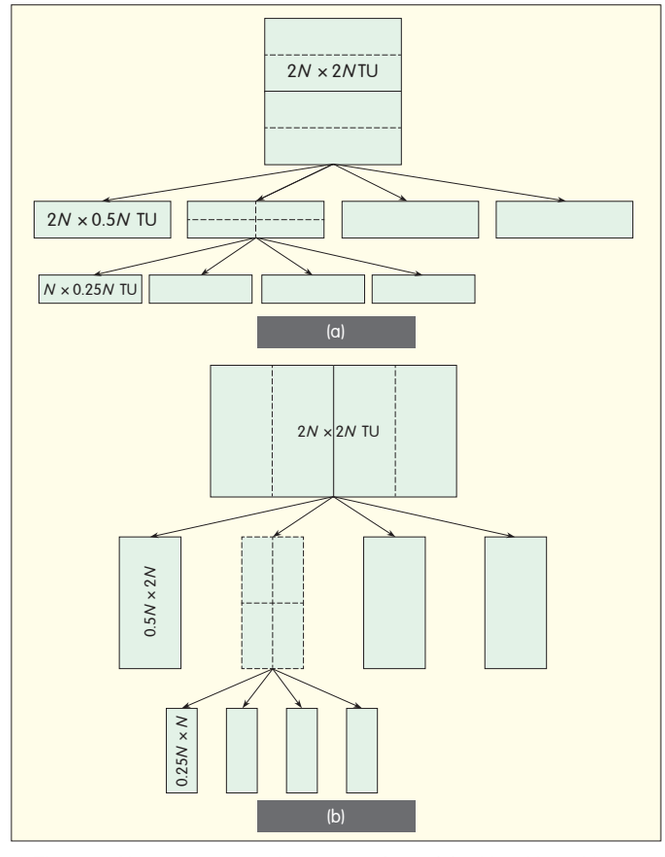
3.6 Intra–Picture Prediction

The unified intra–prediction coding tool provides up to 35 directional prediction modes, including DC and planar modes for the luma component of each PU [28]. The 33 possible intra prediction directions are illustrated in Fig. 6. The derivation of the prediction when using planar mode [29] is given in Fig. 7.

To further improve the intra prediction efficiency of the chroma components, besides the increasing prediction directions, HEVC introduces a new intra chroma prediction mode which utilizes the correlation between chroma and luma samples [30]. Chroma samples are predicted from the reconstructed luma samples around the prediction block by modelling chroma samples as a linear function of luma samples. The model parameters are determined by linear regression using the neighbouring reconstructed pixels of the current coding luma and chroma blocks.

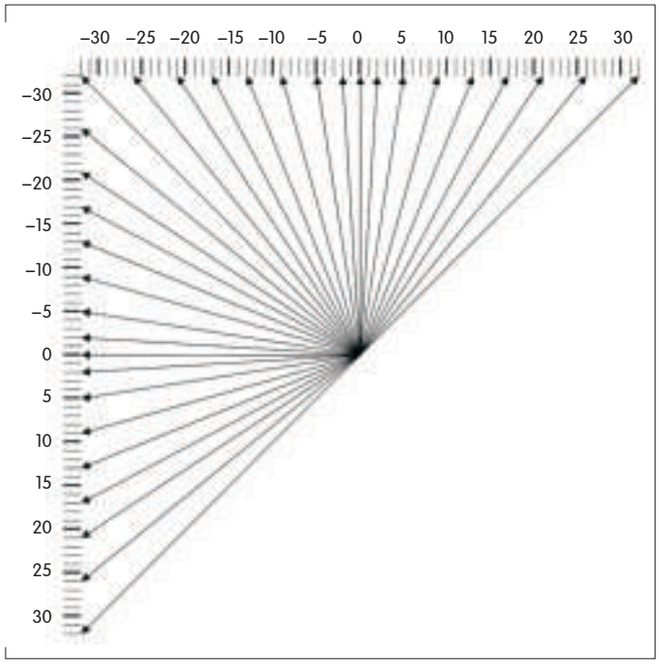
3.7 Inter–Picture Prediction

Combining the quad–tree CU splitting with the designed



▲Figure 5. Non–square quad–tree transform: (a) $2N \times N$ partition, (b) $N \times N$ partition.

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P8



▲Figure 6. Intraprediction directions in HEVC.

PU partitioning methods equips HEVC with an ability of coping with various local features of video texture. Meanwhile, the spatial statistical redundancy of the prediction residuals is reduced by employing RQT with varied transforms. Besides the above mentioned features that bring much improvement to the performance of inter–picture prediction in HEVC, some well–developed coding tools also make a large contribution for the finally achieved high coding efficiency from different aspects, such as motion information representation, de–aliasing filtering, fractional pixel motion prediction and compensation, and so on.

To effectively code the information of MVs, advanced MV prediction (AMVP) [13] is proposed to conduct adaptive MV prediction by exploiting spatial–temporal correlation of MVs from neighbouring PUs. AMVP is used in deriving the predictor for the current MV. AMVP scans first the MVs from spatial PUs and then temporal neighbouring PU positions in some specified locations and orders to construct MV predictor candidate list. Then, encoder selects the best predictor from the candidate list for the current coding MV and codes corresponding index indicating chosen candidate, as well as the MV difference, in the bit–stream.

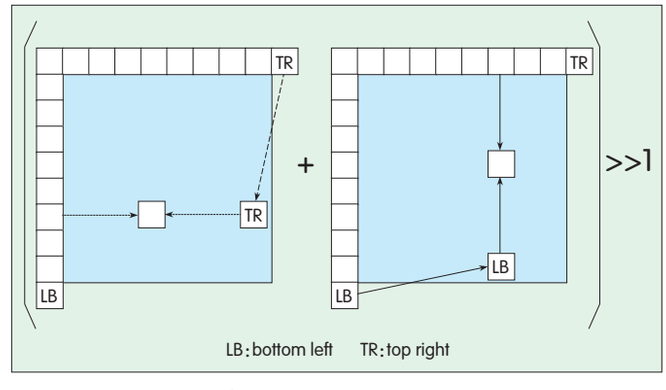
Besides AMVP, merge and skip modes [13] are also adopted into HEVC for implicitly signalling motion information (including inter–picture prediction direction, MV and reference index) of a PU. When using merge mode, encoder picks out the motion information from both spatial and temporal PUs neighbouring to the current PU in a pre–defined pattern to construct a motion information candidate list. Then encoder selects the best candidate and directly employs that motion information in the current PU's motion compensated prediction (MCP) process and codes the candidate index instead of the motion information into the bit–stream. The derivation of the motion information for skip mode is similar to that of merge mode, with the only difference that the prediction residual is not encoded but assumed to be 0.

To suppress the aliasing artefacts in the reference pictures, interpolation filters (IFs) are always employed to generate fractional samples for high precision inter–picture prediction. In HEVC, separable filters using fixed coefficients are derived based on fractional point DCT transforms, namely DCT–IF [31]. The prediction precision can be 1/4–pel for luma samples and 1/8–pel for chroma samples. Different from the cascading interpolation filters in H.264/AVC, where 1/4–pel luma reference pixels are obtained firstly performing 1/2–pel interpolation and then getting 1/4–pel samples using bi–linear filters, HEVC directly calculates both 1/2 and 1/4–pel reference samples by the designed set of filters, which helps to achieve superior performance.

In order to overcome the shortcomings of the relative manner (through memory management control operations (MMCO) and sliding window) for decoded picture buffer (DPB) management method in H.264/AVC, such as vulnerability to losses of pictures that contains MMCO commands, restrictions on the encoder in its selection of coding structure and reference picture usage when temporal scalability is used, etc., reference picture set (RPS) [32] is developed and integrated in HEVC. RPS describes the reference pictures in the DPB in an absolute manner in each slice header of a picture. It contains a list of delta picture order count (deltaPOC) information of all reference pictures that the decoder shall keep. The deltaPOC is used to calculate the picture order count (POC) value of a reference picture as $POC_{\text{reference}} = POC_{\text{reference}} + deltaPOC$. Therefore, POC is not only used by the decoder to deliver the pictures in the correct order for display but also for identification of reference pictures during reference picture list construction and decoded reference picture marking.

3.8 Entropy Coding

CABAC is determined as the single entropy coding method for HEVC. CABAC combines an adaptive binary arithmetic coding engine with context modelling and achieves a high degree of adaptation and redundancy reduction. Generally,



▲Figure 7. Derivation of the prediction using planar mode.

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P9

there are four stages for CABAC to code a value of a syntax element. First, CABAC uses a binarization algorithm selected according to the input syntax to convert its value into bins suitable for entropy coding. Then, CABAC chooses statistical model, namely context model, for the current coding one or several bins by referencing the available information from the adjacent coded blocks and bins. And arithmetic entropy coding using the selected context model is performed to generate coding bits. Finally, the context model is updated according to the information collected during the actual coding process.

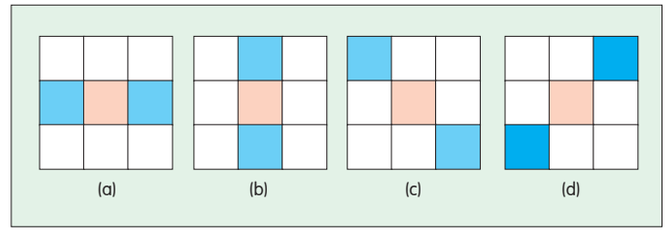
3.9 In–Loop Filtering

In–loop filtering module is introduced in the HEVC codec to suppress noise and artifacts brought about by lossy compression. The filtered pictures can be used as references in the MCP loop for encoding and decoding successive pictures. Deblocking filter, SAO, and ALF are the in–loop filters cascaded for HEVC. Every filter has various candidate working modes to cope with different kinds of compression noises in pictures with diversified features, which gives flexibility to an encoder to achieve desired coding qualities under the considerations of the requirements of applications.

Blocking artifacts are often observed at the boundaries of TUs, which is caused by the discontinuity of the quantization errors between adjacent TUs. The deblocking filter in HEVC [13] is designed based on that of H.264/AVC with some modifications and improvements to fit the HEVC coding features.

SAO and ALF are two filters aiming at removing the quantization noise. SAO is applied to the reconstruction signal after the deblocking filter. SAO classifies reconstructed pixels into categories and reduces the distortion by adding a corresponding offset to pixels of each category in different picture regions. There are two kinds of offsets in SAO, i.e. band offset (BO) and edge offset (EO). BO classifies all pixels of a region into multiple bands where each band contains pixels in the same intensity interval. The intensity range is equally divided into a pre–defined number of intervals from zero to the maximum intensity value (e.g. 255 for 8–bit pixels), and each interval has an offset. Then the bands are divided into two groups. One group consists of the central half of the total bands, while the other group consists of the remaining half. Only offsets in one group are transmitted. EO uses four 1–D 3–pixel patterns representing the typical edge directional features for pixel classification (Fig. 8). The encoder selects one pattern for each region of a picture to classify pixels into multiple categories by comparing each pixel with its two neighbouring pixels, and sends. The selection in bit–stream as side information.

ALF is a Wiener filter based algorithm which is applied to the reconstructed signal after the SAO and/or the deblocking filter. The filtering process uses 2D filters for luma and chroma samples. The encoder calculates the filter coefficients using Wiener based adaptive filtering algorithms, makes a decision on whether or not ALF is applied via a rate–distortion optimization process, and finally codes both filter coefficients and control flags into bit–streams. At the receiver, the decoder first obtains the filter coefficients from parsing APS while the decoder obtains the CU control information from the slice header, and then applies ALF filtering process to the reconstructed pictures.



▲Figure 8. Four 1D three–pixel patterns for the pixel classification in EO: (a) 0 degrees, (b) 90 degrees, (c) 135 degrees, (d) 45 degrees.

4 Performance Comparisons with H.264/AVC

Performance comparisons are reported in [33] between the HEVC draft standard and an anchor reference of the H.264/AVC High Profile. The conditions used for the comparison tests were reportedly designed to reflect relevant application scenarios, while enabling a fair comparison to the maximum extent feasible, i.e. using comparable quantization settings, reference frame buffering, etc. Several of the encoder optimizations currently found in the HEVC software were tested and reportedly shown to be helpful to improve the H.264/AVC anchor performance. Therefore, the testing was indicated to be generally configured in favour of using a relatively strong H.264/AVC anchor reference. When compared to the improved anchor encoder configurations, the HEVC draft standard design reportedly provides a bit rate savings of about 39% for equal coding quality measured in PSNR for random access applications, 44% for low–delay applications, and 25% for all–intra use cases.

5 Conclusions

The coming HEVC video coding standard is being jointly developed by ITU–T VCEG and ISO/IEC MPEG organizations. HEVC represents a number of advances in standard video coding technology, in terms of both coding efficiency enhancement and flexibility for effective applications. Its VCL design is based on conventional block–based motion–compensated hybrid video coding concepts, but with some important differences relative to prior standards as summarized below:

- APS containing common information of picture adaptive tools
- quad–tree based CU splitting
- asymmetric PU partitioning
- quad–tree based TU partitioning
- unified intraprediction, including planar mode and chroma linear prediction using luma samples
- AMVP, merge and skip modes for inter–picture prediction

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P10

• DCT–IF for both 1/2–pel and 1/4–pel fractional sample calculation
• RPS–based DPB management and reference picture marking
• unified entropy coding approach
• advanced in–loop filtering, including SAO and ALF beside deblocking filters
• leaf–CU–aligned slices, tiles, entropy slices, and WPP.

When used well together, the features of the new design provide approximately a 40% bit rate savings for equivalent coding quality in PSNR terms relative to the performance of prior standard H.264/AVC, which is very appealing for the desired applications.

Acknowledgments

The authors thank the experts of ITU–T VCEG, ISO/IEC, MPEG, and the ITU–T/ISO/IEC Joint Collaborative Team on Video Coding (JCT–VC) for their contributions.

References

[1] ITU–T Q6/16 Visual Coding and ISO/IEC JTC1/SC29/WG11 Coding of Moving Pictures and Audio, *Joint Call for Proposals on Video Compression Technology*, MPEG Document, N11113, Kyoto, Japan, January 2010.
[2] ITU–T. *Advanced Video Coding for Generic Audiovisual Services, ITU–T Recommendation H.264 and ISO/IEC 14496–10 AVC Standard*, Geneva, Switzerland, June, 2011.
[3] Video and Requirements Subgroups, *Vision, Applications and Requirements for High–Performance Video Coding (HVC)*, MPEG Document, N11096, Kyoto, Japan, January 2010.
[4] ITU–T. *ITU–T Recommendation H.261 Version 2, Video Codec for Audiovisual Services at p × 64 kbit/s*, Geneva, Switzerland, 1990.
[5] ISO/IEC. *ISO/IEC 13818–2: 1994, Information Technology–Generic Coding of Moving Pictures and Associated Audio, Part 2: Visual*, 1994.
[6] ITU–T. *Video Coding for Low Bit Rate Communication, ITU–T Recommendation H.263*, Geneva, Switzerland, January 2005.
[7] JCT–VC. *Test Model under Consideration*, JCT–VC Document, JCTVC–A205, Dresden, Germany, April 2010.
[8] T. K. Tan, G. J. Sullivan, J.–R. Ohm, *Summary of HEVC working draft 1 and HEVC test model (HM)*, JCT–VC Document, JCTVC–C405, Guangzhou, China, October 2010.
[9] T. Wiegand, W.–J. Han, J.–R. Ohm, G. J. Sullivan, *High Efficiency Video Coding (HEVC) text specification Working Draft 1*, JCT–VC Document, JCTVC–C403, Guangzhou, China, October 2010.
[10] T. Wiegand, W.–J. Han, B. Bross, J.–R. Ohm, G. J. Sullivan, *WD2: Working Draft 2 of High–Efficiency Video Coding*, JCT–VC Document, JCTVC–D503, Daegu, Korea, January 2011.
[11] T. Wiegand, B. Bross, W.–J. Han, J.–R. Ohm, G. J. Sullivan, *WD3: Working Draft 3 of High–Efficiency Video Coding*, JCT–VC Document, JCTVC–E603, Geneva, Switzerland, March 2011.
[12] B. Bross, W.–J. Han, J.–R. Ohm, G. J. Sullivan, T. Wiegand, *WD4: Working Draft 4 of High–Efficiency Video Coding*, JCT–VC Document, JCTVC–F803, Torino, Italy, July 2011.
[13] B. Bross, W.–J. Han, J.–R. Ohm, G. J. Sullivan and T. Wiegand, *High efficiency video coding (HEVC) text specification draft 7*, JCT–VC Document JCTVC–I1003, Geneva, Switzerland, May 2012.
[14] Y. Morigami, J. Tanaka, T. Suzuki, *CE4 subtest 3: Quantization matrix for HEVC based on JCTVC–F362 and F475*, JCT–VC Document, JCTVC–G434, Geneva Switzerland, November 2011.
[15] C.–M. Fu, C.–Y. Chen, C.–Y. Tsai, Y.–W. Huang, S. Lei, *CE13: Sample Adaptive Offset with LCU–Independent Decoding*, JCT–VC Document, JCTVC–E049, Geneva, Switzerland, March 2011.
[16] K. McCann, B. Bross, S. Sekiguchi, W.–J. Han, *HM4: High Efficiency Video Coding (HEVC) Test Model 4 Encoder Description*, JCT–VC Document, JCTVC–F802, Torino, Italy, July 2011.
[17] S. Wenger, J. Boyce, Y.–W. Huang, C.–Y. Tsai, P. Wu and M. Li, *Adaptation Parameter Set (APS)*, JCT–VC Document JCTVC–F747, Torino, Italy, July 2011.
[18] K. McCann, W.–J. Han and I.–K. Kim, *Samsung's Response to the Call for Proposals on Video Compression Technology*, JCT–VC Document, JCTVC–A124, Dresden, Germany, April 2010.
[19] T. Davies, *BBC's Response to the Call for Proposals on Video Compression Technology*, JCT–VC Document, JCTVC–A125, Dresden, Germany, April 2010.
[20] D. Marpe, H. Schwarz and T. Wiegand, *Context–Based 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, July 2003.
[21] Y.–W. Huang, I.–K. Kim, *CE4: Summary report of core experiment on slice boundary processing and fine granularity*, JCT–VC Document, JCTVC–E024, Geneva, Switzerland, March 2011.
[22] A. Fuldseth, M. Horowitz, S. Xu, A. Segall, M. Zhou, *Tiles*, JCT–VC Document, JCTVC–F335, Torino, Italy, July 2011.
[23] K. Misra, J. Zhao and A. Segall, *Lightweight slicing for entropy coding*, JCT–VC Document, JCTVC–D070, Daegu, Korea, January 2011.
[24] C. Gordon, F. Henry, S. Pateux, *Wavefront Parallel Processing for HEVC Encoding and Decoding*, JCT–VC Document, JCTVC–F274, Torino, Italy, July 2011.
[25] L. Guo, M. Karczewicz, X. Wang, Y. Yuan, Y. He, X. Zheng, H. Yu, *CE2: Asymmetric Motion Partition, Non–Square Quadtree Transform and Overlapped Block Motion Compensation*, JCT–VC Document JCTVC–F582, Torino, Italy, July 2011.
[26] A. Fuldseth, G. Bjøntegaard, M. Budagavi, V. Sze, *CE10: Core transform design for HEVC, JCT–VC document*, JCTVC–G495, Geneva, Switzerland, November 2011.
[27] K. Chono, H. Aoki, Y. Senda, *Pulse code modulation mode for HEVC*, JCT–VC document, JCTVC–E057, Geneva, Switzerland, March 2011.
[28] J.–H. Min, S. Lee, I.–K. Kim, W.–J. Han, J. Lainema, K. Ugur, *Unification of the Directional Intra Prediction Methods in TMuC*, JCT–VC Document, JCTVC–B100, Geneva, Switzerland, July 2010.
[29] J. Chen, T. Lee, *Planar intra prediction improvement,* JCT–VC document, JCTVC–F483, Torino, Italy, July 2011.
[30] J. Chen, V. Seregin, J. Kim, B. Jeon, *CE6.a.4: Chroma intra prediction by reconstructed luma samples*, JCT–VC documents, JCTVC–E266, Geneva, Switzerland, March 2011.
[31] E. Alshina, A. Alshin, J.–H. Park, J. Lou, K. Minoo, *CE3: 7 taps interpolation filters for quarter pel position MC from Samsung and Motorola Mobility*, JCT–VC Document, JCTVC–G778, Geneva, Switzerland, November 2011.
[32] R. Sjöberg, D. Flynn, Y. Chen, Y.–K. Wang, TK Tan, W. K. Wan, *JCT–VC AHG report: Reference picture buffering and list construction (AHG21)*, JCT–VC Document, JCTVC–G021, Geneva, Switzerland, November, 2011.
[33] B. Li, G. J. Sullivan, J. Xu, *Comparison of Compression Performance of HEVC Working Draft 4 with AVC High Profile*, JCT–VC Documents, JCTVC–G399, Geneva, Switzerland, November 2011.

Manuscript received: January 17, 2012

Biographies

Ping Wu (ping.wu@zte.com.cn) received his BEng degree in electrical engineering from Tsinghua University, Beijing, in 1985 and his PhD degree in signal processing from Reading University, England, in 1993. From 1993 to 1997, he was a Research Fellow in the area of medical data processing in Plymouth University, England. From 1997 to 2008, he was a consultant engineer in News Digital Systems Ltd, Tandberg Television, and Ericsson. He participated in the development of ISO/IEC MPEG and ITU–T video coding standards. He also supervised the engineering team to build the high–definition H.264 encoder products for broadcasters. From 2008 to 2011, he joined Mitsubishi Electric Research Centre Europe and continued to participate in high efficiency video coding (HEVC) standard development with contributions in Call for Evidence and Call for Proposals. From 2011, he has been a senior specialist in video coding at ZTE. He has many technical proposals and contributions to the international standards on video coding over past 15 years.

Ming Li (li.ming42@zte.com.cn) received his BEng. degree in telecommunication engineering and PhD degree in communication and information systems in Xidian University, Xi'an, China, in 2005 and 2010. He has been a standardization engineer in video coding in ZTE since 2010. His current research interests include video coding and multimedia communication.

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT———1PPS/P11

Recent MPEG Standardization Activities on 3D Video Coding

Yichen Zhang and Lu Yu
(Department of Information Science & Electronic Engineering, Zhejiang University, Hangzhou 310007, China)

Abstract

The Moving Picture Experts Group (MPEG) has been developing a 3D video (3DV) coding standard for depth–based 3DV data representations, especially for multiview video plus depth (MVD) format. With MVD, depth–image–based rendering (DIBR) is used to synthesize virtual views that are based on a few transmitted pairs of texture and depth data. In this paper, we discuss ongoing 3DV standardization and summarize coding tools proposed in the responses to MPEG ' s call for proposals on 3DV coding.

Keywords

3DV coding; call for proposal; auto–stereoscopic; depth map

1 Introduction

Depth–based 3D video (3DV), including multiview video plus depth (MVD), have attracted interest from industry and academia [1]. 3DVs are textured videos with several possible views and associated depth maps. With depth data, virtual views can be synthesized from transmitted views using depth–image–based rendering (DIBR) [2].

Depth–based 3DV formats have some advantages over conventional multiview formats. In a multiview video shot using a camera rig, a stereo pair can be presented on a stereoscopic display, and the baseline from the stereo pair is fixed. In this case, the 3D experience may not sit comfortably with different users because they have different preferences for depth intensity (which is mainly dictated by the baseline). With view synthesis, the arbitrary viewpoints between two coded views can be easily interpolated, and new stereo pairs with desired baseline distances can be generated. This enables disparity–adjustable stereoscopic video. Autostereoscopic displays, which provide a glasses–free 3D experience, require five, nine, or even 28 views as input. Coding all views one by one (simulcasting) or using multiview video coding (MVC) is insufficient. With 3DV, most of the required views can be rendered with a few coded views, and a video in 3DV format consumes less bandwidth than one in a multiview format.

At the 96th MPEG meeting in Geneva, a call for proposals (CfP) was issued on 3DV coding technology [3]. The CfP represented the start of standardization of depth–based 3D formats, among which MVD was the first priority.

In section 2, we introduce the CfP, including all requirements and test conditions. In section 3, we summarize the responses to the CfP, give a brief overview of the proposed 3DV coding tools, and introduce three representative coding algorithms. In section 4, we discuss the standardization schedule of 3DV. Section 5 concludes the paper.

2 Requirements and Test Conditions

2.1 Test Materials

In the CfP, two classes of test sequences (MVD format) were used as test materials. One class included four sequence sets: Poznan_Hall2, Poznan_Street, Undo_Dancer, and GT_Fly. These sets had 1920 × 1088 resolution and were 25 frames per second (fps). The other class included four sequence sets: Kendo, Balloons, Lovebird1, and Newspaper. These sets had 1024 × 768 resolution and were 30 fps. The individual sequences in each set were eight or ten seconds long. For each sequence set, two and three specific views of the texture and depth data were the input of the two–view and three–view test scenarios, respectively.

2.2 Compatibility Requirement

3DV coding must be compatible with existing H.264/AVC (advanced video coding) or future high–efficiency video coding (HEVC). The compressed data format in 3DV must be compatible with that of H.264/AVC, which supports mono or stereo video. Existing AVC decoders must be able to

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P12

reconstruct samples from the mono (AVC–compatible) or stereo (MVC–compatible) views of the compressed bit streams. Similarly, the 3DV compressed data format must also be compatible with the HEVC standard, which was close to completion when the 3DV CfP was issued. HEVC decoders must be able to reconstruct at least one view from the compressed bit streams [4].

Two test categories were defined in the CfP: AVC–compatible, and HEVC–compatible and unconstrained. In the former, proposals are AVC compatible; in the latter, proposals are HEVC–compatible or have no compatibility constraints [3]. Fig. 1 shows examples of AVC–compatible and MVC–compatible coding schemes.

2.3 Anchor Generation

For the AVC–compatible test, anchors for the objective and subjective measurements were generated using an MVC encoder (JMVC version 8.3.1) to encode the test sequences. For the HEVC compatibility test, anchors for the objective and subjective measurements were generated using an HEVC encoder (HM version 2.0) to encode the test sequences. Both encoders had a high efficient, random–access configuration.

At the time the CfP was issued, the JMVC and HM encoders were state–of–the–art reference software for AVC and HEVC standards.

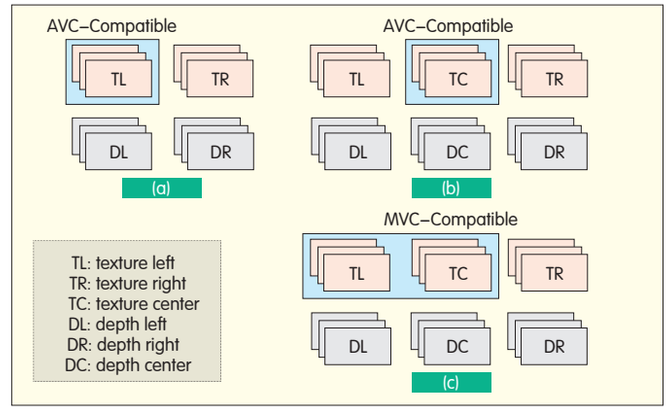
For the AVC compatibility test, MVC was applied separately to texture data and depth data. For the HEVC compatibility test, HEVC simulcasting was used for each view of texture data and depth data. To calculate the objective rate–distortion (RD) performance and provide appropriate materials for subjective evaluation, four rate points (R1, R2, R3, and R4) were determined for each test sequence. R1 was the lowest rate point, and R4 was the highest rate point. The rate points differed for each sequence according to the coding results of previous exploration experiments. These experiments were conducted to develop software and test coding configurations for 3DV standardization [5]. The ratio of R4 to R1 varied from 2 to 5. In all submissions for the CfP, bit rates were limited to below the corresponding target–rate points.

2.4 Evaluation Method

Submissions for the CfP (including decoded and synthesized videos) were subjectively evaluated against the anchors by using stereoscopic and autostereoscopic displays.

Stereoscopic evaluation was performed as follows: In the two–view test scenario, the stereo pair for stereoscopic viewing comprised one of the two decoded views and a synthesized view rendered from the two decoded views. In the three–view test scenario, two stereo pairs were selected. One was centered on the center view of the three decoded views; the other was randomly selected and had the same baseline distance between left view and right view.

For subjective tests with 28–view autostereoscopic displays, all views were formed from the three decoded views and 28 synthesized views. The synthesized views could be produced using VSRS software or another new synthesis method. The 28 synthesized views were distributed evenly among the three coded views. Thus, in the subjective tests, the quality of reconstructed and synthesized texture videos was fully taken into consideration.



▲Figure 1. (a) AVC–compatible category for two–view case, (b) AVC–compatible category for the three–view case, and (c) MVC–compatible category for the three–view case.

3 Responses to the Call for Proposals

Twenty–three proposals were submitted for the CfP. Proposals from Nokia, Sony, Nagoya University, Fraunhofer HHI, NICT, Qualcomm, Philips and Ghent University–IBBT, NTT, Sharp, Samsung, MERL, and Zhejiang University were in the AVC–compatible category. Proposals from RWTH Aachen University, Sony, Fraunhofer HHI (two proposals with different encoder and renderer configurations), Disney and HHI, LG Electronics, Ericsson, ETRI and Kwangwoon University, Samsung (two proposals with different coding tools and prediction structure), and Poznan University of Technology were in the HEVC and unconstrained category.

Prior to the 98th MPEG meeting, the submitted test materials were subjectively assessed in 13 test laboratories around the world [6]. The subjective evaluations showed that, for most test sequences, the subjective quality of R3 of the best–performing proposal was better than R1 of the anchor. This suggests a significant improvement in coding efficiency compared to the anchor. In terms of objective performance, more than 25% rate saving was reported by several proponents.

3.1 Proposed 3DV Coding Tools

To encode the depth–based 3DV format, existing video coding standards, for example, H.264/AVC or HEVC, can be used. However, these standards are optimized for single–view 2D video coding. MVC is an extension of H.264/AVC and is designed for coding a number of texture video sequences with interview prediction. It could be a good candidate for encoding depth–based 3DV; however, it does not take into account the unique functionality or statistical properties of depth data, and it does not exploit the coherence between texture and depth signals. New coding

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P13

tools were included in some of the coding platforms of CfP submissions, and these tools are designed to improve coding efficiency in MVC–based 3DV. These coding tools, which stand apart from those usually found in AVC, MVC or HEVC, can be classified into five categories:

1) Texture–coding dependent views that are independent of depth. This involves coding the texture images of the side view. A side view is any view other than the first view in the coding order. The first view (also called the base view) is expected to be fully compatible with AVC or HEVC; the side view only uses inter–view texture information. Tools in this category include motion parameter prediction and coding, and inter–view residual prediction.

2) Texture–coding dependent views that are dependent on depth. This is applicable to side–view texture, in which original or reconstructed depth information is used to further exploit the correlation between texture images and associated depth maps. Tools in this category include view synthesis prediction for texture and depth–assisted in–loop filtering of texture.

3) Depth coding that is independent of texture. Inter–view depth information or neighboring reconstructed depth values are used to compress the current macroblock in the depth map. Tools in this category include depth intra coding, synthesis–based inter–view prediction and intersample prediction, and in–loop filtering for depth.

4) Depth coding that depends on texture. Original or reconstructed texture information is used to further exploit the correlation between texture images and associated depth maps. Tools in this category include prediction parameter coding, intrasample prediction, and coding of quantization parameters.

5) Encoder optimization. Tools in this category include new rate–distortion optimization (RDO) optimization techniques for depth and texture encoding. They do not affect syntax or semantics.

Tools in the first four categories are used for both encoding and decoding. Tools in the last category are used for encoder optimization only [7].

3.2 Zhejiang University Proposal

The proposal from Zhejiang University focused on depth coding that is AVC–compatible and MVC–compatible. Different tools were proposed, and the average rate reduction for both two–view and three–view cases was 8% [8]. Three coding tools in this submission are introduced in the following subsections. Two of these—view synthesis prediction (VSP) and joint–rate distortion optimization (JRDO)—were also proposed in several other submissions.

3.2.1 Visual Synthesis Prediction for Depth

VSP for depth is a depth–coding tool that synthesizes inter–view depth reference pictures. A depth map of a side view is rendered based on the reconstructed depth picture of another view (often the base view). This depth map is then inserted into the reference picture list. The rendering involves 1) projecting depth pixels onto the target side view according to the depth values of the pixels and the corresponding camera parameters, and 2) filling the holes in the warped image.

Inter–view prediction with VSP is better than inter–view prediction with MVC, which is based on reconstructed pictures from other views. View synthesis compensates for disparities in MVC–based inter–view reference pictures. A depth map synthesized to the target view may be closer to the coded depth map from another view provided that the original depth maps have high inter–view coherence.

3.2.2 Joint Rate–Distortion–Optimization

JRDO is a new RDO for depth coding. Distortion is not measured as loss of depth fidelity due to coding, that is, as the mean–squared error between the original and reconstructed depth signals. Instead, JRDO measures distortion based on reconstructed texture and depth values in order to estimate the distortions in synthesized views. The quality of a synthesized view is more important than that of the depth data, which is not viewed.

Specifically, the distortion measurement in JRDO is a function of the depth distortion (between original and reconstructed values) and the corresponding texture gradient. This measurement is given by

$$D=\sum_{(x,y)\in B}\left|D(x,y)-d(x,y)\right|*\{(t(x+1,y)-t(x,y))^2+(t(x-1,y)-t(x,y))^2\}/c \quad (1)$$

where $D(x, y)$ and $d(x, y)$ denote the original and reconstructed depth values of pixel (x, y), respectively, and $t(x, y)$ denotes the reconstructed texture value. B is the current coding block, and c is a constant.

The distortion measurement is designed based on the fact that the same depth distortions generally cause higher synthesis errors in highly textured regions than in textureless regions.

3.2.3 BBDS in CABAC for Coding mb_type Element

The distribution of mb_type in depth coding is different to that in texture coding. Therefore, a new binarization based on the distribution of syntax (BBDS) element is proposed. We call this element mb_type. The context model selection (CS) also varies with the change of binarization.

According to Huffman code, high–frequency symbols are assigned short codes. This concept is applied to the binarization process. BBDS does not use Huffman code directly because the distribution of a certain syntax element is irregular, and the code is usually irregular. BBDS uses a code tree similar to configurable variable–length code (CVLC) [9] for binarization.

The binarization and CS process has four steps:

1) Remove the values of mb_type that correspond to chroma because the depth image is grayscale.

2) Use one–to–one mapping that translates mb_type into a new SE called mb_type_index. The value of mb_type with a higher probability is mapped to a smaller value of mb_type_index.

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P14

3) As in CVLC, a group of codes is used for the binarization. These codes reflect an approximated distribution of mb_type_index.

4) For each bin of the string, one or more context models are chosen as the model candidates.

4 Standardization Schedule

Standardization plans were established after the Geneva meeting.

4.1 MVC–Compatible Extension Including Depth (AVC–Based First Track)

The main goal of this work item is to enable 3D enhancements and maintain MVC stereo compatibility. Block–level changes to AVC or MVC syntax and decoding processes will not be considered in this item. However, high–level syntax that enables efficient depth–data coding will be supported.

4.2 AVC–Compatible Video–Plus–Depth Extension (AVC–Based Second Track)

A short–term goal is to significantly improve coding efficiency for 3D enhancements in systems that only require 2D AVC compatibility. The syntax and decoding process for non–base texture views and depth information may differ from AVC and MVC at the block level provided the process results in marked improvement in coding efficiency. Coding efficiency is expected to improve 30–40% on existing AVC and MVC technology.

4.3 HEVC 3D Extensions

A third goal is to extend emerging HEVC design to enable efficient stereoscopic/multiview video coding and to support depth coding. Coding efficiency is expected to improve 40–60% on the base specification of HEVC.

At the 98th MPEG meeting, a tentative timeline was established for the standardization of the three extensions (Table 1) [10]. The timeline may be slightly adjusted depending on the development of the standards in the future.

The MVC–compatible extension is due to be finalized soon. The AVC–compatible second track will proceed at a similar pace to HEVC 3D extensions.

5 Conclusion

This paper provides an overview of recent activities on 3DV standardization in MPEG. It summarizes various coding tools that were proposed in submissions for the CfP on 3D video coding. In particular, three depth–coding tools proposed by Zhejiang University are described in some detail. An 8% rate reduction is possible using only a few depth–coding tools. Reduced rate is possible using other tools for depth–assisted texture coding and using tools that exploit the correlation between texture and depth. The best–performing codec in the CfP reduced the rate by more than 25%, which is encouraging evidence supporting the feasibility of AVC and HEVC 3D extensions.

▼Table 1. Timeline for the standardization of 3DV (three categories)

Date	MVC–Compatible	AVC–Compatible	HEVC–Compatible
Dec. 2011	WD		
Feb. 2012	PDAM	WD1	
May 2012	DAM	WD2	WD1
July 2012		WD3	WD2
Oct. 2012	FDAM	PDAM	WD3
Jan. 2013		DAM	PDAM
Jul. 2013			DAM
Jan. 2014		FDAM	FDAM

AVC: advanced video coding
DAM: draft amendment
FDAM: final draft amendment
HEVC: high–efficiency video coding
MVC: multiview video coding
PDAM: proposed draft amendment
WD: working draft

References

[1] K. Müller, P. Merkle, T. Wiegand, "3–D video representation using depth maps," *Proceedings of the IEEE*, vol. 99, no. 4, pp. 643–656, Apr. 2011.
[2] C. Fehn, "A 3D–TV approach using depth–image–based rendering (DIBR)," in *Proc. Visualization, Imaging and Image Processing (VIIP)*, Benalmadena, Spain, 2003, pp. 482–487.
[3] Video and Requirements, "Call for Proposals on 3D Video Coding Technology," ISO/IEC JTC1/SC29/WG11 N12036, March 2011.
[4] Video and Requirements, "Applications and Requirements on 3D Video Coding," ISO/IEC JTC1/SC29/WG11 N12035, March 2011.
[5] Video, "Description of Exploration Experiments in 3D Video Coding," ISO/IEC JTC1/SC29/WG11 N10925, October 2009.
[6] Karsten Müller, Anthony Vetro, and Vittorio Baroncini, "Report of Subjective Test Results from the Call for Proposals on 3D Video Coding Technology," ISO/IEC JTC1/SC29/WG11 N12347, November 2011.
[7] Heiko Schwarz, Krzysztof Wegner, and Thomas Rusert, "Overview of 3DV coding tools proposed in the CfP," ISO/IEC JTC1/SC29/WG11 N12348, December 2011.
[8] Lu Yu, Deliang Fu, Yin Zhao, Xingguo Zhu, Yichen Zhang, and Peng Lv, "Description of 3D Video Coding Technology Proposal by Zhejiang University," ISO/IEC JTC1/SC29/WG11 M22674, November 2011.
[9] Ngai–Man Cheung, and Yuji Itoh, "Configurable variable length code for video coding," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Proc.*, Salt Lake City, Utah, 2001, pp. 1805–1808.
[10] Video Subgroup, "Standardization Tracks Considered in 3D Video Coding," ISO/IEC JTC1/SC29/WG11 N12434, December 2011.

Manuscript received: February 29, 2012

Biographies

Yichen Zhang (felixzyc@gmail.com) received his BEng degree in communication engineering from Hangzhou Dianzi University, China, in 2010. He is currently completing his PhD degree at the Institute for Information and Communication Engineering, Hangzhou Dianzi University. His research interests include 3D video processing and video coding.

Lu Yu (yul@zju.edu.cn) received her BEng degree in radio engineering and PhD degree in communication and electronic systems from Zhejiang University, China, in 1991 and 1996. She is currently a professor at the Institute for Communication Engineering, Zhejiang University. In 2002, she was a senior visiting scholar at the University of Hannover, Germany, supported by the China Scholarship Council and German Research Foundation. In 2004, she was a senior visiting scholar at the Chinese University of Hong Kong, supported by the United College Resident Fellow Scheme. She has published more than 100 technical papers and contributed more than 200 proposals to national and international standards. Her research interests include video coding, multimedia communication, and relative application–specific integrated circuit design. She is the chair of the Video Subgroup of the Audio Video Coding Standard (AVS) of China and was previously co–chair of the Implementation Subgroup of AVS.

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P15

AVS 3D Video Coding Technology and System

Siwei Ma, Shiqi Wang, and Wen Gao
(Institute of Digital Media, Peking University, Beijing 100871, China)

Abstract

Following the success of the audio video standard (AVS) for 2D video coding, in 2008, the China AVS workgroup started developing 3D video (3DV) coding techniques. In this paper, we discuss the background, technical features, and applications of AVS 3DV coding technology. We introduce two core techniques used in AVS 3DV coding: inter–view prediction and enhanced stereo packing coding. We elaborate on these techniques, which are used in the AVS real–time 3DV encoder. An application of the AVS 3DV coding system is presented to show the great practical value of this system. Simulation results show that the advanced techniques used in AVS 3DV coding provide remarkable coding gain compared with techniques used in a simulcast scheme.

Keywords

AVS; 3D video coding; inter–view prediction; stereo packing

1 Introduction

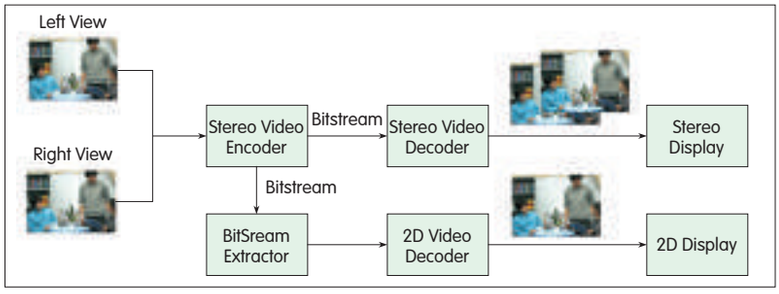
The China Audio Video Standard (AVS) video coding standard is developed by the AVS workgroup, whose role is to establish general technical standards for the compression, decoding, processing, and representation of digital audio and video [1]. After ten years, the AVS workgroup has developed a series of standards based on different applications, and these standards have attracted the attention of both industry and academia. In 2007, AVS was accepted by the ITU–T IPTV focus group as one of four video formats. With the fast development of display technologies and rapidly growing demands of 3D video (3DV) applications, high–efficiency 3DV compression is needed. The most straightforward 3DV coding scheme is simulcast, in which compression and transmission are performed separately for each view. However, simulcast ignores inter–view correlation, which produces double the amount of data compared with traditional video. Thus, simulcast is not the optimal solution for 3DV coding. In 2008, the AVS workgroup launched the 3DV coding project to satisfy demand for higher resolution and better quality that had arisen as a result of widespread 3DV usage [2].

Currently, the AVS workgroup is focused on stereoscopic video coding because of the rapidly growing 3DV market and number of applications. Two advanced stereoscopic video coding schemes have been adopted: inter–view prediction and enhanced stereo packing coding [3]. In inter–view prediction, the correlation between two channels is greatly reduced by allowing disparity compensation from the inter–view frame. Enhanced direct–mode prediction and enhanced motion–vector prediction further improve coding performance [4]. In enhanced stereo packing, the stereoscopic image of each view is down–sampled by half and merged into a single frame. View prediction is allowed in the frame in order to improve coding efficiency. This technique supports backward compatibility with existing 2DV coding infrastructure. To flexibly support these two 3DV coding schemes, AVS defines high–level syntax at both system layer and video layer.

Because of the fast development of microelectronic techniques, there is an urgent need to develop a dedicated AVS 3DV real–time encoder chip that is capable of huge throughput and mass computation in consumer applications. Although AVS is designed for optimized coding and low complexity, compressing high–definition (HD) stereoscopic video in real time is a very big challenge. Several key techniques have therefore been proposed. These techniques include parallel pipeline video coding, advanced rate control, and inter–view synchronization. In this paper, we review these key techniques used in encoder chip design and propose an AVS 3D system that incorporates a real–time encoder for TV broadcasting. Our proposal is the first end–to–end AVS 3DV system, and it has already been successfully used to broadcast the Guangzhou Asia Games 2010 on 3DTV.

In section 2, we introduce inter–view prediction and stereo packing schemes used in AVS. In section 3, we propose the

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P16



▲Figure 1. System structure of inter–view based AVS 3DV coding.

3D AVS coding system, including the core techniques for designing an AVS 3D real–time encoder chip and broadcasting system. In section 4, we give the results of experiments conducted with these technologies. Section 5 concludes the paper.

2 3D Video Coding in the Audio Video Standard

2.1 Inter–View Prediction

Fig. 1 shows the basic concept of the AVS inter–view stereoscopic coding system. The input signal comprises left and right views that are captured by a stereo camera. These views are coded using an AVS 3D encoder, and the resulting bitstreams are multiplexed to form the final bitstream packet. At the receiver, the bitstream packet is decoded with the AVS 3D decoder for stereo display. To ensure compatibility with AVS 2D, the sub–bitstream, which represents the independent view, can also be decoded using an AVS 2D decoder and displayed on a conventional 2D display system.

Inter–view prediction uses the already coded data in the other view to efficiently represent the current view [3]. One of the two views, referred to as the base view or independent view, is coded independently using an unmodified AVS P2 video coder. Fig. 2 shows the coding structure of inter–view prediction. To ensure compatibility with monoview AVS, the number of reference frames for both the base view and dependent view is restricted to two. The base view can be decoded independently for 2D display.

To exploit the inter–view correlation, in the dependent view, the first frame is inter–predicted from the reconstructed I frame in the base view. Other P frames in the dependent view can reference either the previous P frames in the same view or the corresponding simultaneously displayed P frame in the base view. Inter–view prediction for the B frame does not affect coding performance; therefore, references for the B frame can only be reconstructed frames from forward and backward directions in the same view.

Because the AVS inter–view coding structure changes the reference–frame mechanism of the dependent view, the related view prediction techniques should also be developed. Recently, two advanced techniques were adopted by the AVS workgroup: enhanced motion vector prediction and enhanced direct mode [4]. These techniques can be used to exploit the correlation between the base view and dependent view in order to improve coding performance.

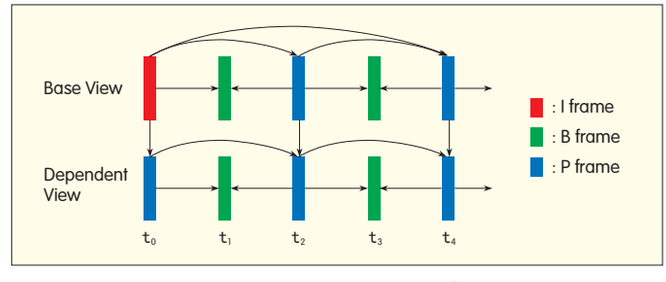
Conventional AVS motion vector prediction for monoview uses scaled motion vectors from four neighboring blocks. However, for the P frame of the dependent view, it is not desirable to use the motion vectors of neighboring blocks if they refer to different channels. To resolve this problem, enhanced motion–vector prediction is proposed. We can assume the current block is A. If A is temporally predicted, the inter–view predicted block in the neighboring blocks is unavailable. Similarly, if A is inter–view predicted, the temporally predicted block in the neighboring blocks is unavailable. This approach ensures that appropriate motion vectors are used for prediction.

In monoview AVS coding, the motion vectors of direct mode for the B frame are derived from the motion vector of the co–located block of the backward reference [5]. However, in inter–view prediction, the farthest reference frame of the backward reference is substituted by the inter–view frame. To obtain accurate motion vectors, when the backward reference is inter–view predicted, the motion vectors of the neighboring blocks are used instead of the disparity vectors.

2.2 Enhanced Stereo–Packing Coding

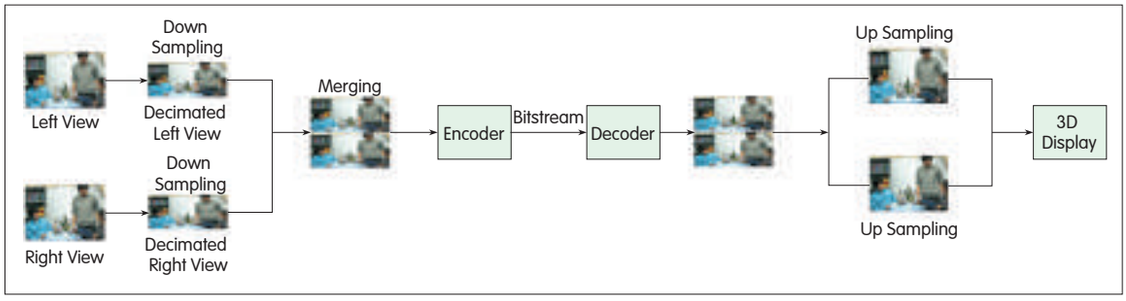
The stereo–packing mode is used for backward compatibility with 2DTV infrastructure and to improve coding performance. Fig. 3 shows how stereo–packing mode is used in AVS 3DV coding. At the encoder, each view is first decimated by half using down–sampling, then the two down–sampled frames are merged into one frame that is the input of a conventional AVS 2D encoder. At the decoder, the bitstream can be decoded using an AVS 2D decoder and can then be detached into multiple views. Each view is up–sampled to support 3D display. Two key techniques in stereo–packing mode involve down–sampling and up–sampling algorithms, and view–merging [6]. Because sampling algorithms are non–normative for the video coding standard, various algorithms can be supported depending on the application scenarios. For more details on the sampling algorithms refer to [6]. Currently, AVS supports two merging approaches: side–by–side and top–to–bottom (Fig. 4). These two approaches make the down–sampling and up–sampling algorithms more flexible.

Coding efficiency in the stereo–packing scheme can be further improved by exploiting inter–view redundancies.

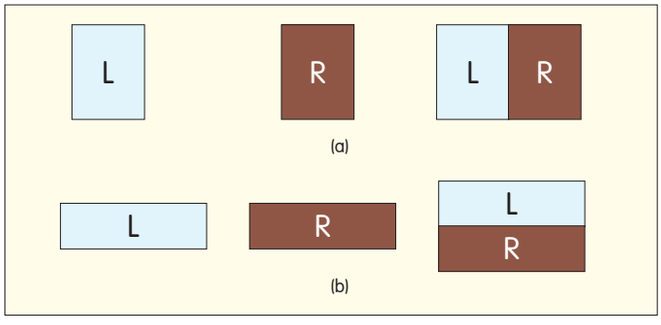


▲Figure 2. Inter–view prediction structure in AVS.

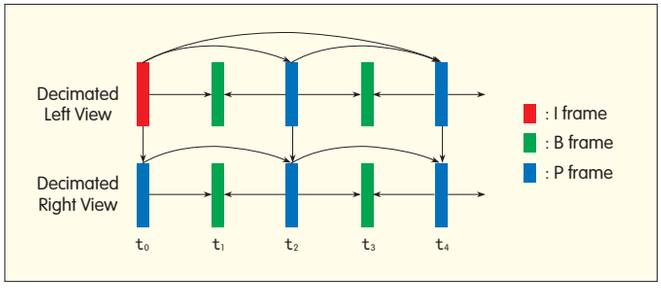
D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P17



◀Figure 3. Stereo packing scheme for AVS 3DV coding.



▲Figure 4. (a) Side–by–side and (b) top–bottom view merging employed in AVS 3DV coding.



▲Figure 5. Inter–view coding structure in stereo packing.

Similar to inter–view prediction, AVS allows interprediction between two decimated views (Fig. 5). This technique is limited in that the encoding process of the dependent view's decimated frame cannot begin until the base view has been encoded.

2.3 High–Level Syntax

To support the two kinds of AVS 3D coding, high–level syntax is designed at both the system layer and video layer. Three AVS 3D coding schemes are created by incorporating the descriptor: simulcast compression, inter–view prediction, and stereo packing. We define a syntax *view_orgnizing_type* for describing each coding system. If the syntax is zero, both simulcast compression and inter–view prediction are supported. If the syntax is one, only stereo packing is supported.

The syntax in the video layer indicates different merging approaches for stereo packing mode (Table 1). Stereo packing mode is fully compatible with monoview coding when the stereo packing mode is set at zero. Moreover, a reserved value is also defined for future extension.

3 AVS 3D Video Coding System

From production to broadcasting, 3DTV usually goes through the processes of acquisition, encoding, multiplexing, modulation, demodulation, demultiplexing, decoding, and display. Among these, the most important is real–time encoding of the HD stereoscopic video. In this section, we discuss AVS 3DV encoder chip design techniques. We also present a 3DTV broadcasting system and discuss potential applications of the AVS 3DV coding standard.

3.1 AVS 3D Real–Time Encoder

Fig. 6 shows the AVS 3D real–time encoding system for HD stereoscopic video. In the encoder, the left and right views are down–sampled and merged into a single frame. The down–sampling direction can be horizontal, to support the side–by–side merging approach, or vertical, to support the top–to–bottom merging approach. The syntax for these approaches is defined in section 2.3. The packing frame is fed into the AVS HD encoder, which generates the AVS bitstream. Finally, the AVS bitstream is packaged into transport stream format for storage or transmission.

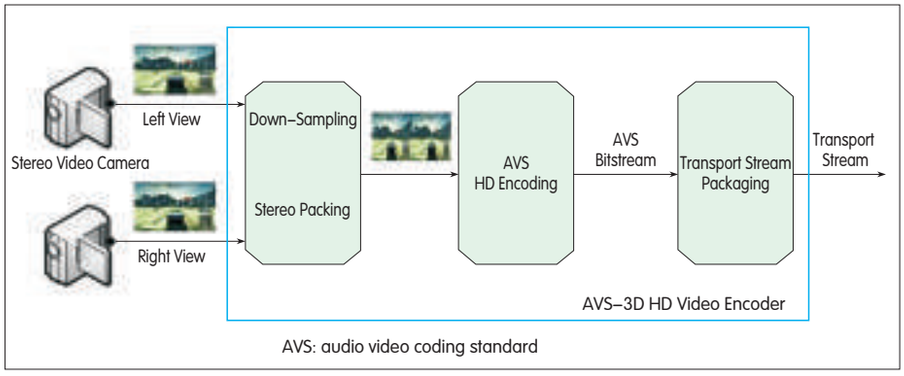
The computing power required by the SD/HD encoder is far beyond the capacity of a single central processing unit (CPU). Fortunately, multicore processors allow the possibility of achieving real–time encoding. To fully exploit multicore processors, parallel encoding algorithms are highly desired in the encoder design.

The motion estimation (ME) module generally takes more than 60% of the total encoding time, and this is a bottleneck for real–time compression. We therefore isolate the ME module for parallel processing. Because the ME module frequently needs to exchange data with other modules, it is not appropriate to use macroblock or finer–level parallel processing for ME. We propose a frame–level parallel ME algorithm that exchanges ME information with other modules

▼Table 1. Stereo packing mode in AVS 3DV coding

Stereo Packing Mode	Packing Method
00	Monoview Coding
01	Side–by–Side
10	Top–Bottom
11	Reserved

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P18



▲Figure 6. AVS 3DV encoder based on stereo packing.

until the ME of the whole frame is finished. Fig. 7 shows the architecture of the proposed dual–pipeline parallel scheme. The ME process is completely isolated and is the first–level encoding process. The output of the ME module is used by other modules in second–level encoding.

The main obstacle in the proposed dual–pipeline parallel video coding scheme is the generation of the reference frame. Conventional video coding uses the reconstructed frame as the reference in the ME process, which means the frame–level ME process cannot start until the reconstructed frame has been obtained. This is problematic for frame–level parallel ME because the ME of the next frame can only begin after the current frame has been encoded. Fortunately, the original frame can be used in the encoder for reference. Because the output of the ME module is only the motion vector information, no error–drifting is incurred in this approach. The reconstructed frames are still used in residual calculation. Although this approach does not ensure that motion vectors obtained in the ME are optimal, it is a practical approach to frame–level parallel ME and strikes a good balance between computational complexity and coding performance.

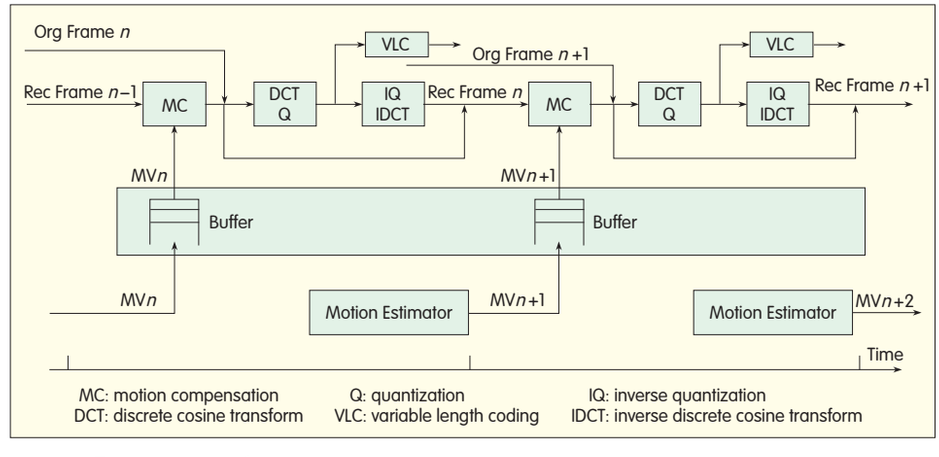
Rate control is important in a practical encoder design. Without rate control, there would be mismatch between the source bit rate and channel capacity, and this would cause underflow or overflow. To accurately control the rate of AVS 3DV coding, we propose a window–based scheme to tackle the problem of interference between rate control and rate distortion optimization (RDO). In this scheme, rate–Qstep (R–Q) and distortion–Qstep (D–Q) models are used to allocate the appropriate number of bits to each coding unit and to adjust the quantization parameter so that each unit is properly encoded with the allocated bits. With the proposed D–Q model, distortion can be estimated, and the optimized coding mode can be obtained by comparing the rate distortion cost of the coding modes. Scene switching is also considered in the window–based scheme because it may cause large bit–rate fluctuations.

Besides these encoder control techniques, synchronization between two views in the 3DV coding is also very important. To synchronize the two views, we use a clock control mechanism and design a scheme based on the AVS system layer specification. We define a transport–stream program map table (PMT) that creates a relationship between the program and its elements. Using this map, we attach a timestamp to each frame and synchronize the timestamps of the two views for synchronized 3DV display.

3.2 AVS 3D Live Broadcasting System

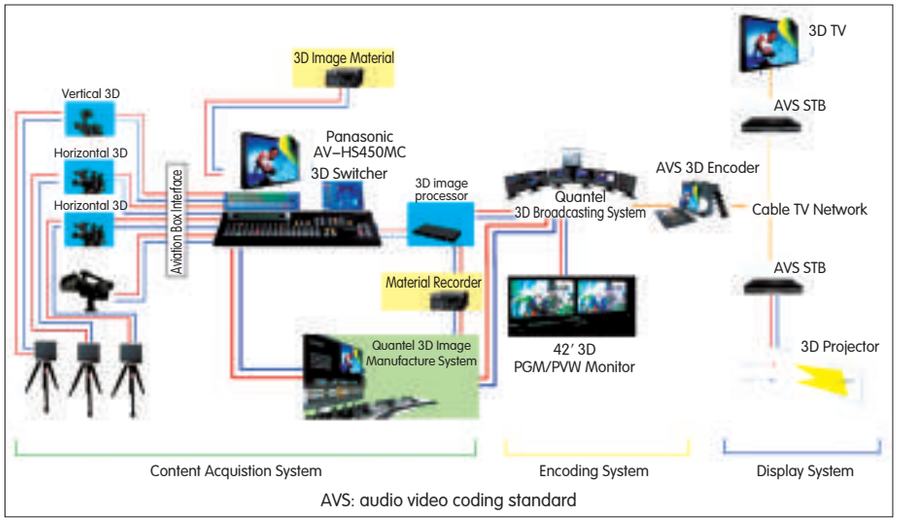
We have already incorporated the AVS 3D real–time encoder into a real 3D live broadcasting system. This system was the first end–to–end broadcasting system and was an example of the practical application of AVS 3DV coding. The system was used for broadcasting 3D TV programs from the Guangzhou Asian Games in 2010. The system successfully delivered an immersive entertainment experience.

Fig. 8 shows the architecture of the broadcasting system, including content acquisition, and encoding and display modules. Two–channel high–definition serial digital interface (HD–SDI) audio and video signals are the input. These signals are first transmitted to the switching station for editing. Other program content, such as captions, can be integrated into the process. Then, the uncompressed audio and video signals are fed into the 3DV processor. In the processor, left and right views are adjusted to ensure the two views match exactly. The signals are then transmitted to the Quantel 3D broadcasting system where the programs can be edited and reviewed. Finally, the signals are fed into the real–time AVS 3D encoder for compression. In the display system, the 3D program stream is input into the set–top box for decoding, and the decoded signal can be displayed by various 3D



▲Figure 7. Architecture of the dual–pipeline parallel video coding scheme.

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P19



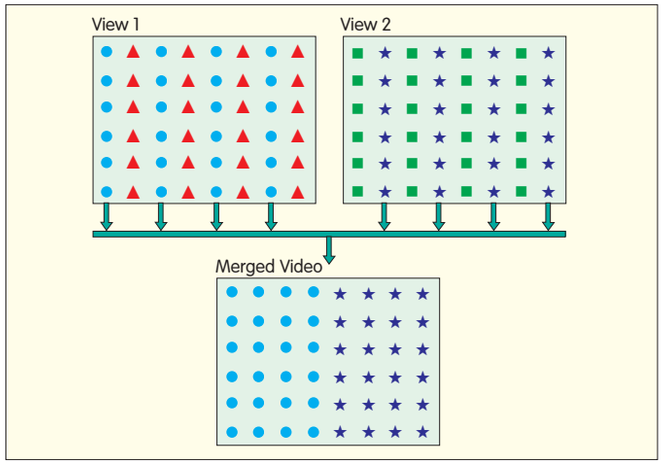
▲Figure 8. AVS 3D live broadcasting system.

▼Table 2. Performance of inter-view prediction and simulcast schemes

Sequence	ΔR	ΔPSNR
Akko_Kayo	−16.86%	0.941
Alt_Moabit	−30.34%	2.312
Book_Arrival	−26.74%	1.694
Car	−21.3%	0.985
Door_Flower	−26.17%	1.691
Flower3	−26.19%	1.709

display systems, such as a 3D TV or projector.

This system is an optimal, low-cost solution to smoothly transferring from a monoview to 3D TV broadcasting system. This system also highlights the great value of the AVS 3DV coding standard in practical applications such as 3D mobile phone TV, remote interview, video surveillance, and remote learning. The whole 3DV industry chain, from acquisition to display, will benefit from the development of AVS 3DV coding technology.



▲Figure 9. Horizontal downsampling for stereo packing.

4 Performance Comparisons

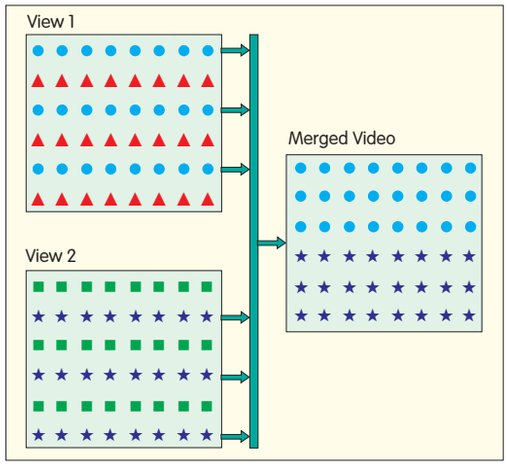
4.1 Inter-Frame Prediction Versus Simulcast

Inter-frame prediction, enhanced motion-vector prediction, and directed mode are integrated into the AVS reference software RM52k_r2. The coding parameters are set according to the general test conditions [7]. The RD performance comparison in [8] is shown in Table 2. From Table 2, inter-frame prediction can reduce the rate by up to 30% for the same peak signal-to-noise ratio (PSNR). The reason for the superior coding performance is that the correlations between two channels are exploited to reduce inter-view redundancy.

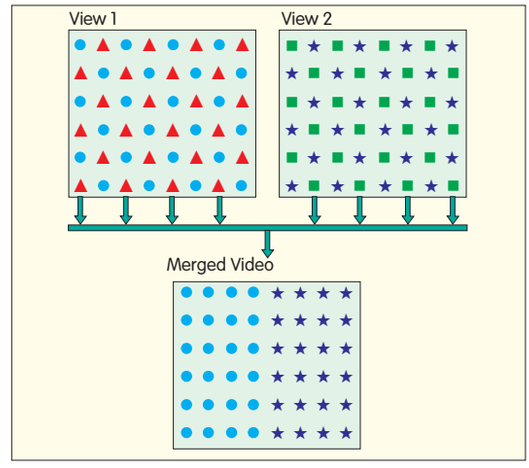
4.2 Stereo Packing Scheme

Side-by-side and top-to-bottom stereo packing make the coding process very flexible because various downsampling and upsampling algorithms can be used. The downsampling and upsampling methods greatly affect coding efficiency. Figs. 9 to 11 show the horizontal, vertical, and diamond downsampling algorithms, respectively.

For each of the down-sampling algorithms, several corresponding upsampling algorithms are used. These

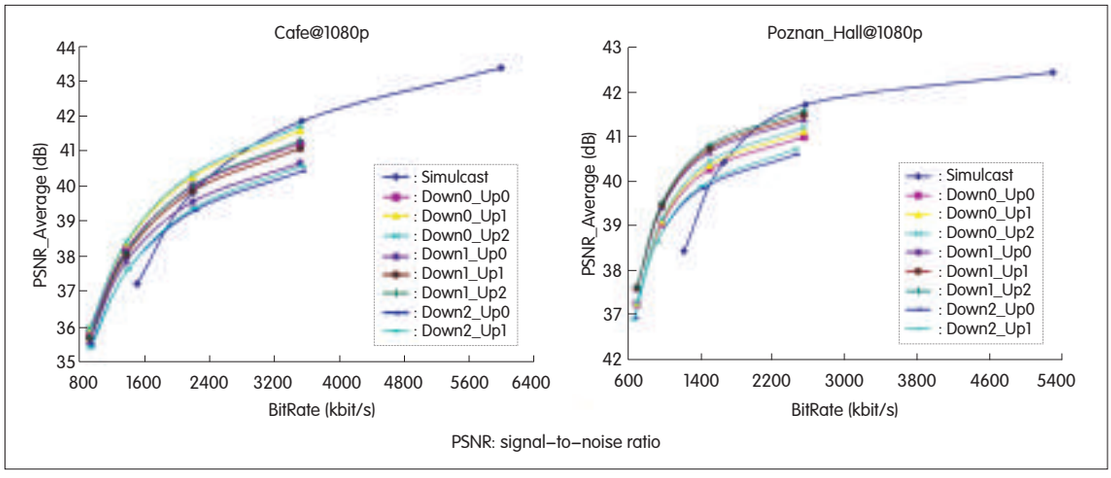


▲Figure 10. Vertical downsampling for stereo packing.



▲Figure 11. Diamond downsampling for stereo packing.

D:\EMAG\2011-02-29/VOL9\CONTETN.VFT——1PPS/P20



◀Figure 12. Performance of different sampling algorithms with the simulcast scheme.

include bilinear, cubic, and AVC–based interpolation algorithms. In Fig. 12, Down0, Down1, and Down2 denote horizontal, vertical, and diamond down–sampling, respectively, and Up0, Up1, and Up2 denote bilinear, cubic, and AVC–based upsampling algorithms, respectively. Combinations of these algorithms are then integrated into the AVS 3D stereo packing scheme, which is implemented in RM52k_r2. The RD performance is shown in Fig. 12. For the sequence Cafe, horizontal downsampling with AVC–based up–sampling performs the best. For sequence PoznanHall, vertical downsampling performs better than horizontal down–sampling. This suggests that the performance of the downsampling method is depends greatly on the properties of video sequence. In the case of low bit rate, the stereo packing scheme is capable of great coding gain compared with the simulcast scheme. At a low bit rate, the quantization of encoding causes most of the distortion, and the stereo packing scheme can provide the best RD trade–off.

5 Conclusion

In this paper, we have discussed the background, technical features, and applications of the AVS 3DV coding standard. The AVS 3DV coding greatly advances coding efficiency and backward compatibility in standard video coding technology. We also introduce the two main features in AVS 3DV coding: inter–view prediction and stereo packing. The AVS 3D TV live broadcasting system shows that the adopted schemes can provide great flexibility for effective use over broad application domains. In the future, more and more new applications will be developed over existing and future AVS 3DV coding technology.

References

[1] L. Yu, S. Chen, and J. Wang, "Overview of AVS–video coding standards," *Signal Process.: Image Commun.*, vol.24, Issue 4, pp. 247–262, 2009.
[2] AVS Requirement Group, Technical requirement of 3D video applications, AVS Doc. AVS N1566. 2008.
[3] X. Ji, Y. Zhang, L. Yu, G. Lee, "Stereoscopic video coding in AVS," *Visual Communications and Image Processing*, Nov. 2011, pp. 1–4.
[4] D. Li, Y. Zhang, Q. Liu, X. Ji, Q. Dai, "Enhanced block prediction in stereoscopic video coding," *3DTV Conference: The True Vision – Capture, Transmission and Display of 3D Video (3DTV–CON)*, 2011, May. 2011, pp. 1–4.
[5] X. Ji, D. Zhao, F. Wu, Y. Lu, and W. Gao, "B–picture coding in AVS video compression standard," *Signal Processing: Image Commun.*, vol.23, Issue 1, pp. 31–41, 2008.
[6] X. Zhao, X. Zhang, L. Zhang, S. Ma, W. Gao, "Low–Complexity and Sampling–Aided Multi–view Video Coding at Low Bitrate," in *Proceedings of IEEE Pacific–Rim Conference International Conference on Multimedia, PCM*, Shanghai, China, Sep. 2010.
[7] AVS Video Group, General test conditions of stereoscopic video coding, AVS Doc. AVS N1760. 2010.
[8] D. Li, X. Ji, Q. Liu, "AVS BM reference software maintenance report", AVS Doc. AVS M2712. 2010.

Manuscript received: April 16, 2012

Biographies

Siwei Ma (swma@pku.edu.cn) received his BSc degree from Shandong Normal University, Jinan, China, in 1999. He received his PhD degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, in 2005. From 2005 to 2007, he was a postdoctoral student at the University of Southern California. Then he joined the Institute of Digital Media, Peking University, where he is currently an associate professor. He has published more than 100 technical articles in refereed journals and proceedings in the areas of image and video coding, video processing, video streaming, and transmission.

Shiqi Wang (sqwang@jdl.ac.cn) received his B.S. degree from Harbin Institute of Technology, China, in 2008. He is currently pursuing his PhD degree in computer science At Peking University.

Wen Gao (wgao@pku.edu.cn) received his PhD degree in electronics engineering from the University of Tokyo in 1991. He is a professor of computer science at Peking University. From 1991 to 1995, he was a professor of computer science at Harbin Institute of Technology and a professor at the Institute of Computing Technology, Chinese Academy of Sciences. He has published five books and more than 600 technical articles in refereed journals and conference proceedings in the areas of image processing, video coding and communication, pattern recognition, multimedia information retrieval, multimodal interface, and bioinformatics. Dr. Gao has been on the editorial boards of several journals, including IEEE Transactions on Circuits and Systems for Video Technology, IEEE Transactions on Multimedia, IEEE Transactions on Autonomous Mental Development, EURASIP Journal of Image Communications, and Journal of Visual Communication and Image Representation. He chaired a number of prestigious international conferences on multimedia and video signal processing, including IEEE ICME and ACM Multimedia. He has been in the advisory and technical committees of numerous professional organizations.

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT———1PPS/P21

Configurable Media Codec Framework: A Stepping Stone for Fast and Stable Codec Development

Euee S. Jang
(Division of Computer Science and Engineering, College of Engineering, Hanyang University, 222 Wangsimni–ro, Seongdong–gu, Seoul, Republic of Korea)

Abstract

Recent advances in reconfigurable computing have led to new ways of implementing complex algorithms while maintaining reasonable throughput. Video codecs are becoming more complex in order to provide efficient compression for video with ever–increasing resolution. This problem is compounded by the fact that spectra of video decoding devices has become wider in the move from traditional TV to cable and satellite TV, IPTV, mobile TV, and Internet media. MPEG is tackling this problem with a reconfigurable video coding (RVC) framework and is standardizing a modular definition of tools and connections. MPEG's work started with video coding and has recently extended to graphics data coding. RVC will be supported by non–MPEG standards such as the Chinese audio–video standard (AVS). This article gives a brief background to the reconfigurable codec framework. The key to this framework is reconfigurability and reducing granularity to find commonality between different standards.

Keywords

MPEG; reconfigurable coding; RVC; RMC

1 Introduction

The Motion Picture Experts Group (MPEG) has created many audio–visual coding standards, including MP3, MPEG–2, and MPEG–4 AVC/H.264 [1]. MPEG's multimedia coding standards have been central in the shift from an analog to digital paradigm. Video coding standards have been developed for specific applications: MPEG–1 for video CD, MPEG–2 for digital TV and DVD, MPEG–4 Part 2 Visual for mobile video, and MPEG–4 AVC/H.264 for DMB and Internet. It has been 20 years since MPEG–1 was standardized. There are many video coding standards, some from MPEG and others from non–MPEG organizations. However, competition between standards makes it difficult to develop video devices because such devices must support an ever–increasing number of codecs.

It could be argued that there should only be one or two generic video codecs and that standards should be unique. This may seem idealistic considering a huge amount of content has already been created using various standards. However, if we consider how media coding is done, a generic video coding standard is not impossible. Most media coding standards have basic processes: prediction, quantization, transform, and entropy coding. If a decoder is componentized into modules, it may be possible for a video coding standard to reuse modules from another video coding standard. The size of the module determines the granularity of the module, and the hardware and software of a module may vary in size, performance, and cost. The reusability of a module greatly increases when the right granularity is found for a given architecture.

The granular design of a codec can be used to describe a media coding standard, from bitstream syntax parsing to reconstruction of pixels or audio samples. All the media coding standards cannot be merged into one, but they can be described in a generic coding framework. In 2003, MPEG began standardizing a reconfigurable video coding (RVC) framework. The RVC framework can be considered a configurable media codec (CMC) framework that encompasses not only video coding but also audio and graphics coding.

MPEG first took the CMC framework approach in the area of video coding. This is a fast–evolving area because more demanding video services require more efficient coding standards. MPEG and ITU–T's Video Coding Experts Group (VCEG) have joined together to standardize high–efficiency video coding (HEVC), which aims to provide the most efficient

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P22

compression. HEVC is expected to be more complex than MPEG–4 AVC/H.264. Very recently, Internet video coding (IVC) and web video coding (WVC) have also been proposed as royalty–free coding standards for Internet applications. Such diversity in video coding standards calls for CMC to be considered.

One of the main objectives of CMC is to narrow the gap between the design and implementation of algorithms. Generally speaking, designers of video coding algorithms do not take implementation into consideration when determining the merits of one algorithm over another. They instead design algorithms according to compression efficiency, the first requirement of video coding. Preferred algorithm designs are often complex and are difficult to implement. Designing algorithms according to implementation has been tried, but it is difficult because architecture such as hardware and software, single core and multicores, and floating–point and fixed–point arithmetic varies widely. Algorithm–architecture co–design has only recently been acknowledged as an important next step in research [2].

The idea of modularizing the codec with common tools came about by first considering how a module is constituted. A module is a functional unit (FU) comprising input, output, and internal processing. The FU can be described as a function call in a program, a logic unit in a chip, or a thread running in a parallel computing environment. An FU is designed to provide an abstract form of a function that can be implemented in different environments. MPEG's FU design is similar to the black–box approach, although this was not cleatly stated in the RVC standard. As long as input and output behaviors in an FU implementation conform to the standard, internal implementation of FU is left open.

A decoder can be viewed as an FU with one input (for example, a bitstream) and three outputs (for example, YUV). However, the granularity of a large FU does not conform to the goal of the RVC framework, that is, to define a toolbox containing FUs that can be reused in many coding standards. FU granularity is key in determining how efficient the RVC framework is. FUs that are standardized in a video tool library (ISO/IEC 23002–4) are not thoroughly verified in terms of whether they are efficiently segmented or divided with optimal granularity. The initial goal of RVC standardization was to design a proper framework for configuring FUs to form a decoder network.

FUs must be configured and connected in such a way as to form a decoder network that is interoperable with different implementations. The model of computation (MoC) of the formed decoder network is a dataflow model in which the input and output of the FUs are called tokens. The availability of input tokens determines FU execution of input tokens to produce output tokens. Therefore, connections between FUs are data–driven. The dataflow model is a significant departure from the traditional model of computation based on signal flow. Most signal–processing algorithms can be modeled as a signal–flow graph, and there is no room for functions or computations at the individual node. In a data–flow graph, additions, multiplications, and cosine functions can be hidden in a node. Input and output are described as input and output edges (or tokens).

A data–flow–based description of MoC is a simplified description of a decoder network (FU network). The remaining implementation details, such as buffer management, timing, and data precision, are unspecified so that implementation can be flexible. This is why there are two standard specifications, one for the framework (ISO/IEC 23001–4) and one for the toolbox (ISO/IEC 23002–4). The framework standard contains the decoder description language, used to describe the FU network, and bitstream syntax parsing. The toolbox standard contains video coding FUs and a simulation model with several decoder configurations for existing video coding standards.

The RVC framework is intended to cover not only video coding but also audio and graphics. The MPEG graphics community recently started work on a reconfigurable graphics coding (RGC) framework that is similar in principle to the RVC framework. The main goal of the RGC framework is to construct a toolbox for MPEG graphics coding tools [3]. Activities relating to the RGC framework include confirming the RVC approach for any–media coding. GPUs are heavily used in graphics applications, and the modular design of the RGC framework helps in the implementation of FUs, which are well–suited for such graphics applications.

The CMC approach looks promising, and modular design in parallel computing is attracting interest in areas where multicores and GPUs can accelerate computing. MPEG is not the only group interested in CMC. The Audio Video Standard (AVS) Group in China shares MPEG's vision of CMC and has been developing its own FUs to support AVS codecs.

This paper takes the MPEG RVC framework as a good example of the CMC approach. A few years have passed since CMC was first hatched by MPEG, but there is much room for improvement of technology and standards.

2 The CMC Framework

There are two main issues for the modular design in CMC: how to define a module and how to connect modules. In MPEG RVC, a module is called a functional unit (FU). Input and output behavior is normatively defined, and internal processing is left open and is implementation–specific.

2.1 Module Design Philosophy

When designing a module in CMC, implementation and granularity, testability, and interoperability should be taken into account.

2.1.1 Implementation and Granularity

A module should be implementable in platforms that have hardware or software, single core or multiple cores. Abstract modeling is often preferred to physical implementation because it increases flexibility when implementing modules in various platforms. The module in MPEG RVC is designed using an abstract definition of FU in the text specification and using an examplar implementation of FU in RVC–CAL

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P23

Texture decoding

Algo_IS_Zigzag_4x4 FU

FU Name	Algo_IS_Zigzag_4x4	
Description	This module inverts the one-dimensional array of coefficients ordered in zigzag scan to 2D raster order. It inputs a list of 16 integer coefficients (one per 4x4 block) and outputs the ordered list of integer values.	
Profiles@levels supported	MPEG-4 AVC Constrained BP	
Input		
Name	Token	
levarr	BLOCK token	
Output		
Name	Token	
Lev2d	BLOCK token	
Parameter		
Name	Description	Range
SAMPLE_SZ	Size in bits of the levarr and Lev2d ports	[0...16]

▲Figure 1. An abstract FU definition from MPEG VTL.

language. In the text specification, each FU is viewed as a black box, and in RVC–CAL implementation, each FU is viewed as a white box. Module granularity is included in the FU definition, and directly affects the reusability and reconfigurability of modules. For this reason, both implementation and granularity should be clearly defined.

2.1.2 Testability

Efficient testing and debugging of an implemented module is one of the goals of CMC. Adequate module granularity helps reduce testing and debugging work. In MPEG RVC black–box testing, golden responses are generated by analyzing the corners of a given FU. The black–box approach is taken to ensure that different module implementations can be tested in a standard way.

2.1.3 Interoperability

The standard definition of a media codec has, so far, been confined to the bitstream syntax, and parsing and decoding algorithms. Implementation of algorithms in a codec is unspecified. Industry fills this gap by enhancing the compression efficiency of encoding algorithms (through, for example, efficient mode decision) and enhancing the encoding and decoding algorithms with cost–effective implementations. An implementation designer can create customized algorithms, for example, a combined implementation of quantization and transform. Using CMC, interoperability between modules of different implementations may be possible if any implemented module conforms to the input and output behavior of the abstract module definition. Therefore, it is possible to produce a decoder comprising a combination of modules from different implementations. This has not been possible with conventional decoder implementation. In a multimedia framework such as DirectShow, the only visible component has been a decoder, not the modules to generate a decoder. In MPEG RVC, module–level interoperability is not yet supported because the first goal of MPEG RVC is to provide a framework, not the modules.

2.2 Case Study: MPEG Video Tool Library

The MPEG video tool library (VTL) (ISO/IEC 23002–4) is a collection of FUs and part of the MPEG RVC standard. The tools (or FUs) available in MPEG VTL are supported by the MPEG codec configuration representation (CCR) standard. Fig. 1 shows an abstract definition of an inverse–scan FU used in MPEG–4 AVC/H.264. Two important fields in the abstract are input and output. The input is a 4 × 4 BLOCK token, and the output is also a BLOCK token. The description field contains a brief description of what the FU does internally with input and output tokens. The exact behavior is not explicitly described. There could be various implementations of the FU. Fig. 2 shows a reference description and implementation in RVC–CAL.

In Fig. 1, the FU testing is very much like black–box testing. In Fig. 2, the testing is white–box testing. MPEG RVC is not clear about this issue yet, but the most important thing is that the input and output behavior is transparent to any implementation.

2.3 Module Connections

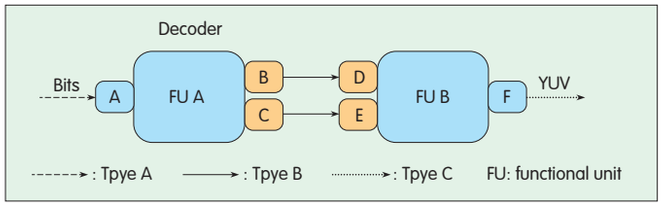
Once modules are defined, connections between modules have to be made in order to form a module network. When connecting the modules, any data transaction between modules should be defined clearly enough so that any implementation follows the specification. In MPEG, input and output data of FUs are called tokens. These are the basic elements for connecting FUs into an FU network. In a packet–switched network such as the Internet, a datagram is similar to a token. However, a token is different to a datagram in that the size and format of each token may be different to one another. A variety of token types influences the design of interconnections between modules. If too much information is carried in a token type, modularization may not be done with optimal granularity. Connections and traffic between modules

▲Figure 2. A reference implementation in RVC–CAL of the abstract FU definition from Fig. 1.

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P24

Configurable Media Codec Framework: A Stepping Stone for Fast and Stable Codec Development

Euee S. Jang



▲Figure 3. A simple FU network with two FUs, two internal connections, and two external connections.

```
<Instance id = "FU A">
    <Class name = "Algo_Example1" />
</Instance>
<Instance id = "FU B">
    <Class name = "Algo_Example2" />
</Instance>
```

Type A	<connection src = "FU A" src-port = "B" dst = "FU B" dst-port = "D" /> <connection src = "FU A" src-port = "C" dst = "FU B" dst-port = "E" />
Type B	<input src = "FU A" src-port = "A" />
Type C	<output src = "FU B" src-port = "F" />

▲Figure 4. An FND of Fig. 3.

should be therefore be minimized when defining modules.

In CMC, connections are described in a readable format because they could be essential information in implementations. The following information should be described: connections between module input and output ports, definition of the token type of each connection (e.g., block of 8 × 8, pixel, MB, 1–bit flag), token sequence or order, and parameters for specific implementations.

In MPEG RVC, the module network is descibed with an XML–like description called the FU network description (FND). The rules for describing connections are defined in the FU network language (FNL) in the MPEG RVC standard. A diagram is commonly used to describe the module connections. Fig. 3 shows an FU network in MPEG–RVC. There can be up to four different token types, that is, two external and two internal. Input and output ports that share the same connection should support the same token type.

The diagram helps the implementation designer understand the modular network, but it is also desirable to describe the network in language format. Fig. 4 shows an FND written in FNL.

2.4 Syntax Parser

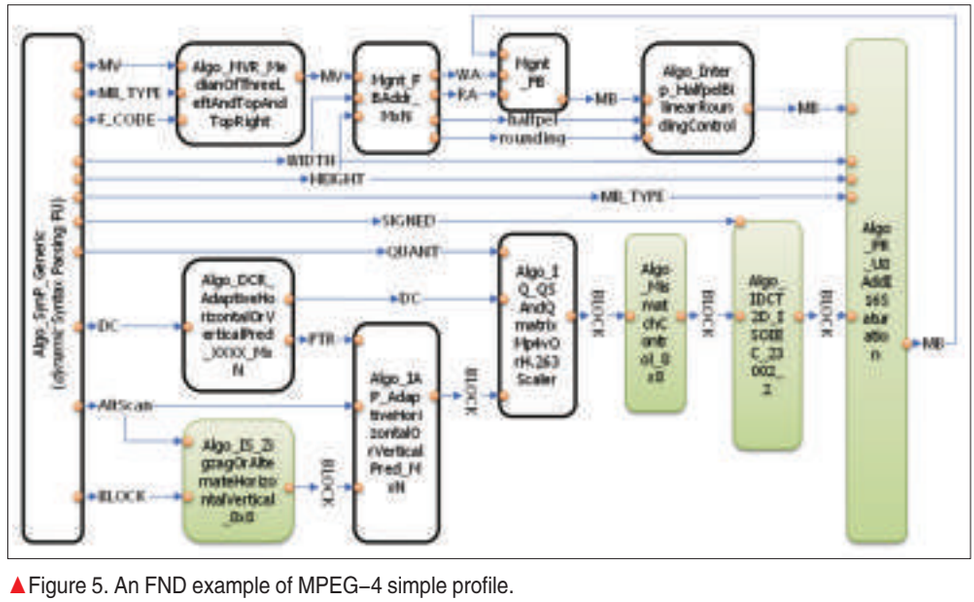
The bitstream syntax and parsing process is unique for each codec and usually includes entropy coding (variable–length decoding, arithmetic decoding). Unlike in the modular CMC approach, the syntax parser module is less likely to be reused by other codecs and is highly codec–dependent. The parser is usually the first module to process the bit stream. In MPEG–RVC, the bit–stream parser description (BSD) is part of the decoder description. Each decoder description contains FND and BSD, and a parser module can be generated from the BSD. The BSD format is RVC bit–stream syntax description language (RVC–BSDL), a variant of XML.

A syntax parser can run without necessarily engaging the decoding process. This means that the syntax parsing and entropy decoding process can be detatched from the decoding process, and conformance of bit–stream syntax to bit–stream semantics can be checked. In MPEG RVC, automatic generation of the bit–stream parser from the BSD is still an unresolved issue because generating the parser, including the entropy decoder, is difficult to describe in XML.

Fig. 5 shows an FND that includes all FUs needed to form an MPEG–4 simple–profile decoder. Each box is an FU. The FU on the far left is the syntax parser, which receives a bit stream and produces output tokens (e.g. entropy–decoded semantic data) for the other FUs.

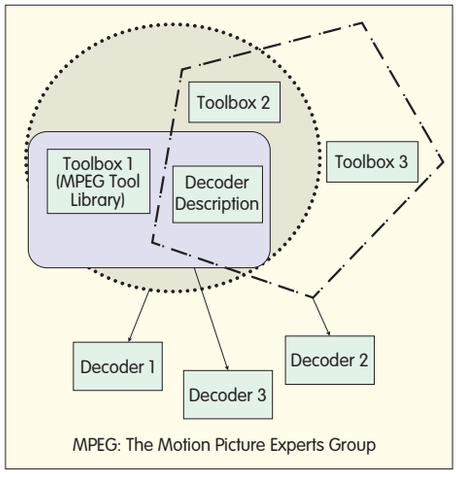
CMC has two parts: framework and toolbox. Fig. 6 shows how different toolboxes can be used to generate a decoder based on the MPEG RVC framework. Other than toolbox 1, the other toolboxes may be proprietary or non–MPEG standards. This opens the way for non–MPEG organizations to use the RVC framework for their own codec implementations and for MPEG codec implementations such as decoder 1 and decoder 2. The AVS group in China supports RVC and multiple toolboxes.

The toolbox approach also extends to other types of media coding, such as graphics coding. In reconfigurable graphics coding (RGC), the RVC framework supports graphics coding tools. Graphics coding is an area that can benefit from RVC. Many graphics applications are multimedia applications that encompass not only geometry data processing but also audio, image, and video data processing. To view a movie, two bit streams are needed, one for video and another for audio. For graphics applications such as games, many data



▲Figure 5. An FND example of MPEG–4 simple profile.

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P25



◀Figure 6. Toolbox concept in MPEG RVC.

sets need to be processed as components that include encoded graphics content. Many graphical object types share common coordinates, colors, and normals. As with many object–type compression methods, graphics data compression involves compressing primitives. For this reason, the division of codecs into modules is easy in graphics coding.

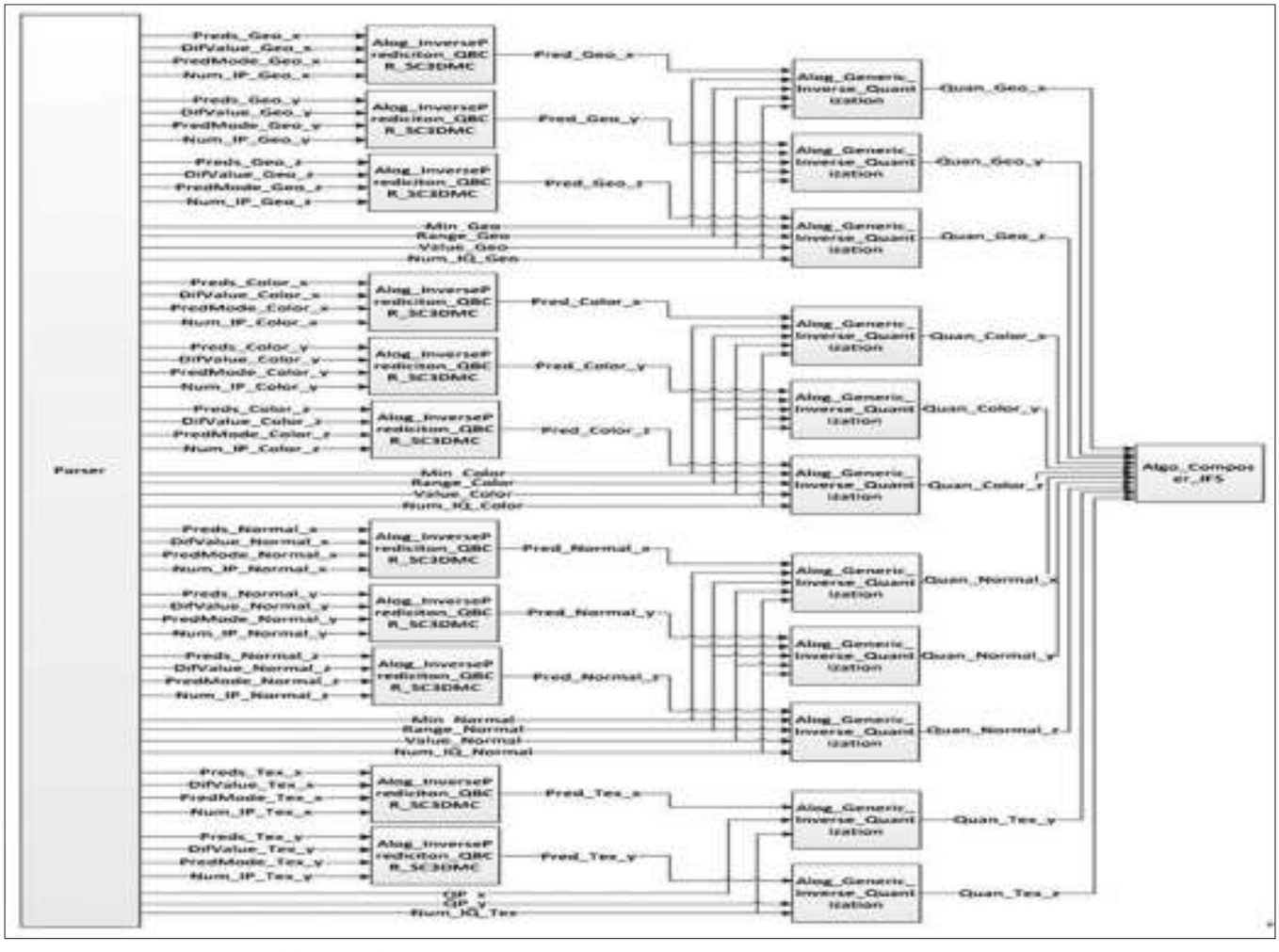
Fig. 7 shows an FND of an MPEG scalable complexity 3D mesh coding (SC–3DMC). Many FUs are reused in order to decode attributes such as coordinates, colors, normals, and texture.

3 Future Research Directions

Although many years have been spent researching and standardizing CMC, this field is relatively young and there is much room for improvement. This is one reason why MPEG RVC continues. This section describes issues that are open for future research.

3.1 Model of Computation

Coding tools are usually represented by algorithms, reference implementations, and textual specifications. In any representation format, the MoC is implicitly defined; otherwise, it would be hard to understand how a coding tool operates for a given functionality. MoC may differ from implementation to



▲Figure 7. An FND of an MPEG SC–3DMC decoder.

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P26

implementation. If there are three consecutive statements, that is, no branch or loop, in a C code, three statements are executed in sequence. Sequential execution may not be guaranteed if the implementation is done in hardware. Parallel execution of three statements may be possible if the statements are independent of each other. The choice of MoC directly affects implementation complexity, and MoC must be chosen carefully.

During the development of MPEG RVC, there have been many discussions about how to define MoC. The consesus is that the reference implementation language, RVC–CAL, should be used as a model to understand MoC in MPEG RVC. To confirm this recommendation, more experiments should be conducted on how to describe a network of modules, how input and output tokens behave in the network, and how a generic description on different implementations can be guaranteed.

3.2 Parser Generation

Bit–stream syntax parsing, including entropy decoding, usually consumes 20 to 40 percent of the decoding time, and this makes the parser one of the most time–consuming modules in the decoder. It is difficult to design a parallel algorithm to speed up the bitstream syntax parser because the parsing process is sequential. This is not the case with other modules. It is also difficult to subdivide the bit–stream syntax parser, which is likely to be the largest module in CMC and outputs the largest number of tokens.

Despite the importance of parser generation, it is not an automatic process yet. In MPEG RVC, there is BSD in the decoder description. However, BSD is not directly used to generate the parser module, and this is called a built–in approach. While it can support existing codecs, the parser is less flexible in generating new codecs as needed. One reason parser generation is not automatic is because the parser includes entropy decoding algorithms, such as variable–length decoding and arithmetic decoding. Entropy decoding requires a complex procedural description, and it may be difficult to define a generic description for any implementation. Future research on automatic parser generation is necessary.

3.3 Design–Time versus Run–Time Generation of the Decoder

There are two distinct approaches in CMC: design–time codec configuration and run–time codec configuration. Most efforts have been focused on design–time configuration. Run–time codec configuration is a challenging issue because of the run–time requirements. In design–time configuration, defined modules may be complex in terms of implementation and computation. This is not the case for run–time configuration, where reasonable performance is expected.

3.4 Granularity of Modules

One of the frontier research areas in CMC is defining proper granularity when designing a module. A decoder can be regarded as a module, and dividing a decoder into modules is only beneficial if there is a gain in the divide–and–conquer strategy. This means the sum of all the processes of modules in a decoder should be less than or equal to that of the decoder. This problem is challenging because the cost may be different from implementation to implementation, from one set of division of modules to another, and from one platform to another. More research has been focused on the framework than on the modules.

3.5 Evolution of the Media Codec

One unrealized but very interesting objective of CMC is the evolution of media codecs through module upgrade. There has been recent discussion within MPEG of a royalty–free video coding standard for Internet applications. To date, it has been a very difficult create a royalty–free standard because standards depend on patent holders. Even if only one algorithm is not royalty–free, there are only a few limited ways to make the entire codec free: wait for up to 20 years until the patent has expired or design a codec standard that circumvents the patented algorithm. Both scenarios are very costly and seldom chosen. With a CMC framework, it is possible to pinpoint the algorithm in a module or set of modules in a decoder. The bypass standard has to include a new set of modules that do not have the patented algorithm. If this approach becomes common in standardization, the number of codecs will grow quickly, and it will be necessary to keep track of tools and their configurations. Although interesting, this idea is yet to be tested and implemented.

4 Conclusion

Standardization of the CMC framework has mostly been the work of MPEG. There are still many issues to be resolved before a dependable framework can be created and modules can be properly defined. MPEG's research is important for fast and stable codecs in the future.

References

[1] MPEG homepage [Online]. Available: http://mpeg.chiariglione.org/
[2] Y.–K. Chen, Gwang–Gook Lee, Marco Mattavelli, Euee S. Jang, "Algorithm/Architecture Co–Exploration of Visual Computing on Emerging Platforms," IEEE Trans. Circuits Syst. Video Techn, vol. 19, no. 11, pp. 1573–1575, 2009.
[3] Sinwook Lee, Taehee Lim, Euee S. Jang, Ji Hyung Lee, Seungwook Lee, "MPEG Reconfigurable Graphics Coding Framework: Overview and Design of 3D Mesh Coding," in Proc. IEEE Visual Commun. And Image Processing (VCIP), Taiwan, Nov. 2011.

Manuscript received: January 26, 2012

Biography

Euee S. Jang received his BS degree from Jeonbuk National University, Korea, and his PhD degree from the State University of New York (SUNY), Buffalo. He is currently a professor at the College of Engineering, Hanyang University, Seoul. His research interests include image/video coding, reconfigurable video coding, and computer graphics objects. He has authored more than 150 MPEG papers and more than 30 journal and conference papers. He also has 35 patents, some of which are pending, and has contributed chapters to two books. He has received three ISO/IEC Certificates of Appreciation for his contributions to MPEG–4 development. He also received a Presidential Award from the Korean government for his contributions to MPEG standardization. Professor Jang is an IEEE Senior Member.

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P27

Lattice Vector Quantization Applied to Speech and Audio Coding

Minjie Xie
(ZTE USA Inc.,Richardson, TX 75080, USA)

Abstract

Lattice vector quantization (LVQ) has been used for real–time speech and audio coding systems. Compared with conventional vector quantization, LVQ has two main advantages: It has a simple and fast encoding process, and it significantly reduces the amount of memory required. Therefore, LVQ is suitable for use in low–complexity speech and audio coding. In this paper, we describe the basic concepts of LVQ and its advantages over conventional vector quantization. We also describe some LVQ techniques that have been used in speech and audio coding standards of international standards developing organizations (SDOs).

Keywords

Vector quantization; lattice vector quantization; speech and audio coding; transform coding

1 Introduction

Vector quantization is generally much more efficient than scalar quantization, and the efficiency of vector quantization improves as the vector dimension increases [1], [2]. With a conventional vector quantizer, however, the computational complexity of the quantization process increases exponentially as the dimension increases, and the storage requirement for the codebook can be very large [1], [3]. Lattice vector quantization (LVQ) can overcome this problem. It is a very promising signal compression technique that has advantages over conventional vector quantization [4]–[7]. In LVQ, the codebook is a finite subset of a regular–point lattice. Because of the regular structure of the lattice, the nearest codeword to the input vector can be found and indexed very efficiently. Another interesting feature of LVQ is that the codewords do not have to be stored because they can be algorithmically generated. Therefore, LVQ has a simple and fast encoding process and significantly reduces the amount of memory required. These advantages can substantially reduce the implementation complexity of a vector quantizer. Applying entropy coding to the quantization indices can further improve the efficiency of LVQ [8]–[10].

LVQ is suitable for use in low–complexity speech and audio coding. Low computational complexity is especially important in telecommunication applications such as video conferencing and mobile communications. For example, a low–complexity audio codec can free cycles for computationally intensive video coding and other audio processing, such as acoustic echo cancellation in video conferencing systems. Low computational complexity can also extend battery life in portable devices. Various LVQ schemes have been developed for speech and audio coding in the past decades [11]–[17], and some of these are used in the codecs that have been adopted as speech and audio coding standards by international standards developing organizations [18]–[23].

In section 2, we briefly review vector quantization and describe the advantages of LVQ over conventional vector quantization. In section 3, we describe some LVQ techniques used in speech and audio coding standards. In section 4, we summarize LVQ applications presented in this paper.

2 Lattice Vector Quantization

2.1 Vector Quantization

Let $\boldsymbol{x} \in \boldsymbol{R}^N$ be an arbitrary vector in N–dimensional Euclidean space $\boldsymbol{R}^N$. An N–dimensional vector quantizer Q with L–level is a function that maps the input vector $\boldsymbol{x}$ into a codeword (code vector) $\boldsymbol{y}_i$ that is selected from a finite codebook $C = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3, \dots, \boldsymbol{y}_L | \boldsymbol{y}_i \in \boldsymbol{R}^N\}$. That is,

$$Q: \boldsymbol{R}^N \rightarrow C \tag{1}$$

and

$$Q(\boldsymbol{x}) = \boldsymbol{y}_i, \text{ if } \boldsymbol{x} \in V_i, \ i = 1, 2, 3, \dots, L \tag{2}$$

where V_i is a partition (Voronoi region) of $\boldsymbol{R}^N$. The Voronoi region V_i is a nearest–neighbor region associated with the

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P28

codeword $\boldsymbol{y}_i$ and is given as

$$V_i = \{\boldsymbol{x} \in \boldsymbol{R}^N: \mathrm{d}(\boldsymbol{x}, \boldsymbol{y}_i) < \mathrm{d}(\boldsymbol{x}, \boldsymbol{y}_j),\ i \neq j\} \quad (3)$$

where $\mathrm{d}(\boldsymbol{x}, \boldsymbol{y}_i) = \|\boldsymbol{x} - \boldsymbol{y}_i\|^2 = \sum_{K=1}^{N}(x_k - y_{ik})^2$, $i = 1, 2, 3, \ldots, L$ is the minimum–squared error and is used as the distortion measure of quantization.

The codewords $\boldsymbol{y}_i$ are usually represented by their indices i, which are used for transmission or storage. A vector quantizer Q is specified by C, V_i, and the indexing of $\boldsymbol{y}_i$.

The rate R of Q is measured in bits per dimension and is given by

$$R = \frac{1}{N}\log_2 L \quad (4)$$

This rate can be used to measure the accuracy of quantization. Vector quantizers are traditionally designed by using K–means or Linde–Buzo–Gray (LBG) clustering algorithms [1], [3]. The conventional vector quantization is usually referred to as being either statistical or stochastic. In statistical vector quantization, the clustering algorithm generates a locally optimal codebook based on a large training database that is related to the signal to be quantized. For a given number of codewords, L, the codebook is designed in the following steps:

Step 1. Initialize the codebook with L codewords. The initial codewords can be obtained by first calculating the centroid of the training database then splitting the centroid into L codewords by using the clustering algorithm.

Step 2. Using the minimum–squared error distortion, associate each input vector with a codeword to determine L partitions (Voronoi regions).

Step 3. Calculate the total average distortion for the training database. If the distortion does not vary or varies only very slightly, the final codebook is obtained. Otherwise, continue.

Step 4. Recalculate the centroid of each partition and use the obtained centroids as the new codewords. Then repeat step 2 and step 3.

In statistical vector quantization, $\boldsymbol{x}$ must be compared with all L codewords in the codebook to find $\boldsymbol{y}_i$ that best matches $\boldsymbol{x}$ in terms of minimum–squared error. The number of codewords is usually $L = 2^{NR}$, and finding the best codeword, that is, the nearest neighbor to $\boldsymbol{x}$ requires $(2N - 1)2^{NR}$ additions, $N2^{NR}$ multiplications, and $(2^{NR} - 1)$ comparisons. The codebook storage requires a memory of $N2^{NR}$ units. The drawback of statistical vector quantization is the high computational complexity for codebook search and the large amount of memory required for codebook storage. Both computational complexity and storage memory required increase exponentially as N and R increase, so it is difficult to improve vector quantization by increasing the dimension or accuracy. This is also an important issue in real–time implementation of statistical vector quantization.

2.2 Lattice Vector Quantization

From a geometric standpoint, a lattice Λ_N is a regular arrangement of points in N–dimensional Euclidean space $\boldsymbol{R}^N$. From an algebraic standpoint, an N–dimensional lattice Λ_N is a collection of vectors $\boldsymbol{y}$ that forms a group under ordinary vector addition in $\boldsymbol{R}^N$, that is,

$$\Lambda_N = \{\boldsymbol{y} \mid \boldsymbol{y} = k_1\boldsymbol{v}_1 + k_2\boldsymbol{v}_2 + k_3\boldsymbol{v}_3 + \cdots + k_N\boldsymbol{v}_N\} \quad (5)$$

where $k_1, k_2, k_3, \ldots, k_N$ are integers, and $\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \ldots, \boldsymbol{v}_N$ are linearly independent vectors in $\boldsymbol{R}^N$ [4].

The simplest lattice is the integer lattice Z_N in which all the N components of the lattice points are integers. Another important lattice, D_N, consists of the lattice points of Z_N, which have integer components with an even–component sum:

$$D_N = \{(y_1, y_2, y_3, \ldots, y_N) \in Z_N \mid \sum_{i=1}^{N} y_i = \text{even}\} \quad (6)$$

The Gosset lattice E_8 is a well–known lattice in 8 dimensions and is defined as the union of the D_8 lattice and the coset $\{D_8 + (\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2})\}$. The Gosset lattice is given by

$$E_8 = D_8 U \{D_8 + (\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2})\} \quad (7)$$

The rotated Gosset lattice RE_8 is given as

$$RE_8 = 2D_8 U \{2D_8 + (1, 1, 1, 1, 1, 1, 1, 1)\} \quad (8)$$

LVQ, also called algebraic vector quantization (AVQ), is an efficient vector quantization technique in which a finite subset of lattice points is used as the codebook of quantizer. Because of the regular structure of the lattice codebook, the nearest codeword to a given input vector can be found and indexed very efficiently. Another interesting feature of LVQ is that the lattice codebook does not have to be stored because the codewords can simply be generated using algebraic rules. Therefore, LVQ has two main advantages: It has a simple and fast encoding process, and it significantly reduces the amount of memory required. These advantages can reduce substantially the implementation complexity of a vector quantizer.

Here, we describe the advantage of LVQ in terms of computational complexity. Suppose an input vector in $\boldsymbol{R}^N$ lies in the truncated lattice, which is used as the codebook of the quantizer. In LVQ based on the Z_N lattice, an N–dimensional vector is quantized by rounding each vector component individually. The fast–quantizing algorithms for LVQ based on D_N and E_8 are described in [24]. For D_N, quantization of an N–dimensional vector requires $(2N - 1)$ additions, $(N - 1)$ comparisons, and $(N + 1)$ rounding operations. In the case of E8, quantizing an 8–dimensional vector requires $4(2N - 1)$ additions, $2N$ multiplications, $(2N - 1)$ comparisons, and $2(N + 1)$ rounding operations. Table 1 shows the computational complexity for finding the best codeword for the input vector in various 8–dimensional vector quantization schemes.

In this example, computational complexity of statistical vector quantization increases exponentially, and the complexity of the LVQ schemes is constant as R increases. In the case of $R = 1$, statistical vector quantization requires 6143 operations to find the best codeword in the codebook. This

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P29

▼Table 1. Computational complexity for various 8–dimensional vector quantization schemes

Operation	Vector Quantization Scheme			
	Statistical	Z_8	D_8	E_8
Addition	15×2^{8R}		15	60
Multiplication	8×2^{8R}			16
Comparison	$2^{8R}-1$		7	15
Rounding		8	9	18
Total	$24 \times (2^{8R}-1)$	8	31	109

computational cost is much higher than those for LVQ in Table 1.

In LVQ there are input vectors lying outside the lattice, which is truncated for a given rate. Hence, additional operations are performed to quantize those input vectors. However, when the quantizer is optimally designed, the number of these vectors can be very small so that computational complexity of LVQ is still quite low. This is true for real–time speech and audio coding applications presented in section 3.

2.3 Voronoi Codes

As defined in (3), the Voronoi region of a lattice point $\boldsymbol{x}$ consists of all points in $\boldsymbol{R}^N$ that are at least as close to $\boldsymbol{x}$ as to any other lattice point. Given a lattice Λ in $\boldsymbol{R}^N$, the Voronoi region around the origin is called the Voronoi region of Λ and is denoted $V(\Lambda)$.

A Voronoi code [25] can be defined as

$$C_\Lambda(r, \boldsymbol{a}) = \Lambda \cap (rV(\Lambda) + \boldsymbol{a}) \quad (9)$$

where $r = 1, 2, 3, \ldots$ and $\boldsymbol{a}$ is a small offset vector that can avoid any lattice point on the boundary of the truncated Voronoi region. The code $C_\Lambda(r, \boldsymbol{a})$ consists of all lattice points in the Voronoi region $V(r\Lambda)$ shifted by $\boldsymbol{a}$ [25]. If r is a power of 2, that is, $r = 2R$ with integer $R > 0$, the code size is $\Lambda / r\Lambda = r^N = 2^{NR}$. Fig. 1 shows the Voronoi codes based on the hexagonal lattice A_2 [4] for $R = 2$ and $R = 3$.

Because there are fast search and indexing algorithms for the root lattices [24], [25], Voronoi codes can be used as codebooks in low–complexity variable–rate lattice vector quantization.

3 LVQ Applied to Speech and Audio Coding Systems

LVQ has not been widely applied to real–time speech and audio coding systems because of several difficulties. These difficulties include truncating a lattice for a given rate in order to create an LVQ codebook that matches the probability density function (PDF) of the input source, quickly translating the codewords of the LVQ codebook into their indices, and quantizing the source vectors "outliers" that are outside the truncated lattice. In this section, we describe two LVQ approaches that have been successfully used in speech and audio coding standards [13], [16]–[23]. We describe solutions to the problem of quantizing spectral vectors in transform–based speech and audio coding.

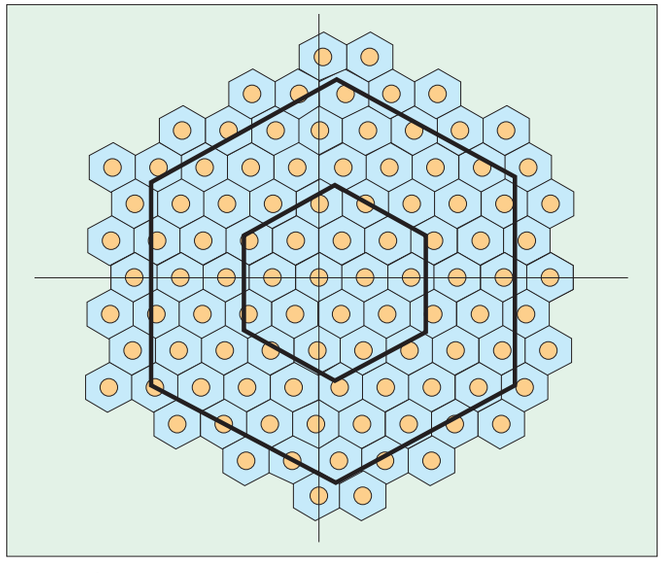
3.1 Embedded Algebraic Vector Quantization

Embedded algebraic vector quantization (EAVQ) is an LVQ application for speech and audio coding [13]. It can be used to quantize spectral vectors in transform coding. With EAVQ as the basis, a split multirate LVQ scheme was developed [16] and has been used in several speech and audio coding standards.

3.1.1 Overview of EAVQ

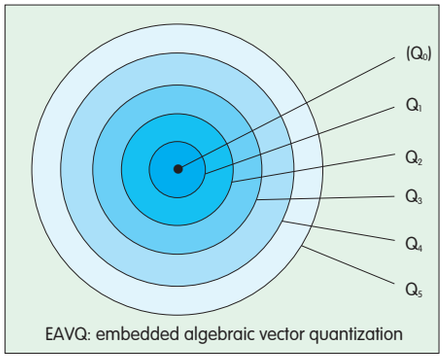
EAVQ is an RE_8–based variable–rate vector quantization scheme [13]. The points of RE_8 fall on concentric spheres with radius $2\sqrt{2r}$ centered at the origin, where r is a non–negative integer that can be used as a natural index for the spheres [26], [27]. In EAVQ, the sets of lattice points on the spheres constitute quantizer codebooks, and each codebook consists of the lattice points inside a specific sphere. The EAVQ quantizer has five subquantizer codebooks, Q_1, Q_2, Q_3, Q_4, and Q5, from low rate to high rate, and these are spherically embedded. A high–rate subquantizer codebook contains those of the lower–rate subquantizers. The subquantizers have codebook sizes of 16 (4 bits), 256 (8 bits), 4096 (12 bits), 65,536 (16 bits) and 1,048,448 (20 bits), respectively. Q_n is a $4n$–bit codebook and comprises 2^{4n} codewords. Additionally, the origin vector $Q_0 = [0, 0, 0, 0, 0, 0, 0, 0]$ is used as a codeword in EAVQ. Fig. 2 shows the structure of the EAVQ codebooks.

The codewords of the codebooks are generated from appropriate permutations of the components of "leaders." A leader is a vector whose components are arranged in descending order. The nearest neighbor of an input vector is easily found by permuting the leaders, and the index of the codeword is obtained by calculating its rank according to an algebraic method [27]. In EAVQ, only the leaders and some



▲Figure 1. Voronoi codes based on the hexagonal lattice A_2.

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P30



◀Figure 2. Structure of the EAVQ codebooks.

parameters are stored as a lookup table for generating and indexing the codewords. This lookup table requires a small number of memory units.

An arbitrary vector in 8 dimensions, denoted $\boldsymbol{x} = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)$, is quantized in the following steps [13]:

Step 1. Find the nearest neighbor $\boldsymbol{y}$ of $\boldsymbol{x}$ in RE_8 by using the fast quantizing algorithm in [24].

Step 2. Reorder the components of $\boldsymbol{y}$ in descending order and then find its leader $\boldsymbol{y}_0$ in a lookup table set up in advance.

Step 3. Scale down $\boldsymbol{x}$ with a predefined scaling factor when x lies outside the largest codebook Q_5, then repeat steps 1 and 2 until a codeword is found in Q_5.

Step 4. Compute the rank t of $\boldsymbol{y}$ as described in [27].

Step 5. Find the nearest neighbor $\boldsymbol{y}'$ of $\boldsymbol{x}$ in Q_1 when x is near the origin Q_0. Then select between $\boldsymbol{y}$ and $\boldsymbol{y}'$ the lattice point closest to $\boldsymbol{x}$ in terms of the mean–squared error (MSE).

Step 6. Compute the index k of the selected codeword and determine the subquantizer number n according to the sphere associated with $\boldsymbol{y}_0$.

Step 7. Stop.

The decoding algorithm is described as follows [13]:

Step 1. From the received index k and the subquantizer number n, find the leader $\boldsymbol{y}_0$ using the same lookup table as in the encoding operation.

Step 2. Find the vector $\tilde{\boldsymbol{x}}$ from k when Q_1 is used, then stop. Otherwise, continue.

Step 3. Compute the rank t according to k.

Step 4. Find the vector $\tilde{\boldsymbol{x}}$ from $\boldsymbol{y}_0$ and t [27].

Step 5. Stop.

3.1.2 Application to Wideband Speech Coding

The EAVQ technique was applied to 50–7000 Hz wideband speech coding at 16 kbit/s [13]. In this application, a speech signal sampled at 16 kHz was encoded by a transform coded excitation (TCX) coder [28], and EAVQ was used to quantize the target signal in the frequency domain. The TCX coder operates on frames of 6 ms corresponding to 96 samples at 16 kHz. For each frame, there are 96 available bits—12 bits are used for the LPC coefficients, 12 bits are used for the pitch parameters, and 72 bits are used for quantization of the target signal. The target signal is converted into a frequency–domain representation by discrete Fourier transform (DFT), and 48 complex coefficients are obtained.

Fig. 3 shows the principle of quantization for the target signal in the TCX coder. The norm (energy) E_x of the complex coefficient vector $\boldsymbol{x}$ is given by

$$E_x = \sum_{i=1}^{96} x_i^2 \tag{10}$$

This norm is then quantized by a 7–bit logarithmic scalar quantizer (SQ). The 96–dimensional coefficient vector $\boldsymbol{x}$ is normalized by the quantized norm E_{xq}. The normalized coefficient vector $\boldsymbol{x}_{nrm}$ is first scaled by a scaling factor to match the EAVQ codebook described in the previous section. The best scaling factor is obtained experimentally. Then, the $\boldsymbol{x}_{nrm}$ is split into 12 subvectors in 8 dimensions, and each subvector is quantized by EAVQ. Finally, the numbers of the 12 subquantizers used are entropy coded.

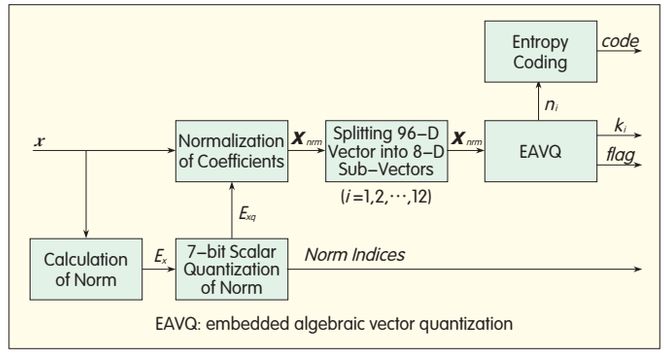
For performance evaluation, the two–dimensional statistical complex vector quantization (2D–CVQ) scheme in [29] was also used in the TCX coder with the same bit–allocation scheme to quantize the target signal in the frequency domain. The objective test results showed that the EAVQ performed slightly better than 2D–CVQ [13]. In addition, memory usage in EAVQ was much less than that in 2D–CVQ.

3.1.3 Split Multirate LVQ

The split multirate LVQ scheme [16] has been used in speech and audio standards such as 3GPP AMR–WB+ [18], ITU–T G.718 [20], G.711.1 Annex D [21], G.722 Annex B [22], and MPEG unified speech and audio coding (USAC) [23].

In this scheme, a modified version of EAVQ is used as a base quantizer, and an additional quantizer called Voronoi extension [16], [30], which is based on the same RE8 lattice, is used to extend the codebooks of the base quantizer when the nearest neighbor of the input vector lies outside these base codebooks. In the base quantizer, only the codebooks Q_0, Q_2, Q_3, and Q_4 in EAVQ are used, and some leaders of Q_3 and Q_4 are replaced [16]. With this modification, Q_2 and Q_3 are still embedded, but they are no longer subsets of Q_4. Actually, Q_4 is a complementary subquantizer codebook to Q_3.

The Voronoi extension is designed by using the Voronoi codes described in section 2.3, and its codebook size depends on the order of extension. For an R–order Voronoi–extension, the codebook size is 2^{8R}. When the nearest neighbor of an input vector cannot be found in the base codebooks, the Voronoi extension is applied, and the



▲Figure 3. EAVQ application to the wideband speech coding in TCX.

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT———1PPS/P31

selected lattice point is represented by the sum of two codevectors: the one from the base codebook Q_3 or Q_4, and the other from the extended codebook. In this case, an input vector is quantized with $12 + 2^{8R}$ or $16 + 2^{8R}$ bits.

In real speech and audio coding systems, outliers may appear when the bit budget is limited. This quantization scheme is usually combined with a gain–shape scheme [1]. The signal is normalized by a gain that is estimated over the current frame of the signal according to a predefined bit budget. Then the signal is quantized.

In the speech and audio codecs previously mentioned, quantization of the signal spectral parameters and transform coefficients is briefly described in the following steps:

Step 1. Normalize the signal by an estimated gain so that it fits within the predefined bit budget.

Step 2. Split the current frame of signal into 8–dimensional vectors.

Step 3. Find the nearest neighbor in RE_8 for each vector.

Step 4. Determine whether the selected lattice point is in the base codebooks. If it is, encode the index of the codebook used and stop. Otherwise, continue.

Step 5. Apply the Voronoi extension and find the codevectors from the base codebooks and extended codebook [16], [30].

Step 6. Compute and encode the index of the codebooks used.

Step 7. Stop or iterate over global gain to adjust overall bit consumption.

3.2 Fast Lattice Vector Quantization

Fast lattice vector quantization (FLVQ) is an LVQ technique applied to low–complexity audio coding and is designed to quantize transform coefficients in transform coding [17], [19].

3.2.1 Overview of Fast Lattice Vector Quantization

In FLVQ, the quantizer comprises two subquantizers: a D_8–based higher–rate lattice vector quantizer (HRQ) and an RE_8–based lower–rate lattice vector quantizer (LRQ). HRQ is a multirate quantizer designed to quantize the input vector at rates greater than 1 bit/dimension. LRQ quantizes the input vector at 1 bit/dimension and uses spherical codes based on RE8 as the codebook. The codebooks of the FLVQ quantizer are constructed from a finite region of lattice and match the probability density function (PDF) of the input vectors. The codewords of HRQ are algorithmically generated, and a fast quantization algorithm is used. The LRQ codebook of 256 codewords is stored in a structured lookup table so that a fast searching method is designed for indexing the codewords.

LVQ is optimal only for uniformly distributed sources. In transform coding, the distribution of transform coefficients is usually not uniform; therefore, entropy coding, such as Huffman coding, is applied to the quantization indices of HRQ to improve the efficiency of quantization in FLVQ.

3.2.2 Higher–Rate Quantization Based on D_8

HRQ is based on the Voronoi code of D_8 presented in section 2.3 and is designed to quantize input vectors at 2 bit/dimension to 9 bit/dimension with increments of 1 bit/dimension. The codebook of this subquantizer is constructed from a finite region of D_8 and is not stored in memory. The codewords can be generated using a simple algebraic method.

To minimize the distortion for a given rate, D_8 should be truncated and scaled. The input vectors are scaled instead of the lattice codebook so that the fast–search algorithm introduced in [24] can be used, and then the reconstructed vectors at the decoder are rescaled. However, this fast–search algorithm assumes an infinite lattice that cannot be used as the codebook in real–time audio coding systems. In other words, for a given rate, the algorithm cannot be used to quantize input vectors lying outside the truncated lattice region. Therefore, a fast method for quantizing these outliers is developed in HRQ.

For a given rate R bit/dimension, where $2 \leqslant R \leqslant 9$, an 8–dimentional vector $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_5, \boldsymbol{x}_6, \boldsymbol{x}_7, \boldsymbol{x}_8)$ is quantized as follows:

Step 1. Apply a small offset $a = 2^{-6}$ to each component of $\boldsymbol{x}$ in order to avoid any lattice point on the boundary of the truncated Voronoi region, that is, $\boldsymbol{x}_1 = \boldsymbol{x} - \boldsymbol{a}$, where $\boldsymbol{a} = (2^{-6}, 2^{-6}, 2^{-6}, 2^{-6}, 2^{-6}, 2^{-6}, 2^{-6}, 2^{-6})$.

Step 2. Scale $\boldsymbol{x}_1$ by the scaling factor α: $\boldsymbol{x}_2 = \alpha \boldsymbol{x}_1$. For a given R, the optimal scaling factor is experimentally selected.

Step 3. In D_8 , find the nearest lattice point $\boldsymbol{v}$ to $\boldsymbol{x}_2$. This can be done by using the searching algorithm described in [24].

Step 4. Suppose $\boldsymbol{v}$ is a codeword in the Voronoi region truncated with R and compute the index vector $\boldsymbol{k} = (k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8)$ of $\boldsymbol{v}$, where $0 \leqslant k_i < 2^R$ and $i = 1, 2, \cdots, 8$. The index $\boldsymbol{k}$ is given by

$$\boldsymbol{k} = (\boldsymbol{v}\boldsymbol{G}^{-1}) \bmod r,\ r = 2^R \tag{11}$$

where $\boldsymbol{G}$ is the generator matrix for D_8 and is defined as follows [4]:

$$\boldsymbol{G} = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{12}$$

and

$$\boldsymbol{G}^{-1} = \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{13}$$

Step 5. Compute the codeword $\boldsymbol{y}$ from $\boldsymbol{k}$ using the algorithm described in [25], then compare $\boldsymbol{y}$ with $\boldsymbol{v}$. If y and $\boldsymbol{v}$ are exactly same, $\boldsymbol{k}$ is the index of the best codeword to $\boldsymbol{x}_2$ and

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P32

stop here. Otherwise, $\boldsymbol{x}_2$ is an outlier and is quantized by the following steps:

Step 6. Scale down $\boldsymbol{x}_2$ by 2: $\boldsymbol{x}_2 = \boldsymbol{x}_2/2$.

Step 7. In D_8 , find the nearest lattice point $\boldsymbol{u}$ to $\boldsymbol{x}_2$. Then compute the index vector $\boldsymbol{j}$ of $\boldsymbol{u}$.

Step 8. Find $\boldsymbol{y}$ from $\boldsymbol{j}$ and then compare $\boldsymbol{y}$ with $\boldsymbol{u}$. If $\boldsymbol{y}$ is different from $\boldsymbol{u}$, repeat steps 6 to 8; otherwise, compute $\boldsymbol{w} = \boldsymbol{x}_2/16$. Because of the normalization of transform coefficients in transform coding, a few iterations may be performed to find a codeword to the outlier in the truncated lattice.

Step 9. Compute $\boldsymbol{x}_2 = \boldsymbol{x}_2 + \boldsymbol{w}$.

Step 10. In D_8, find the nearest lattice point $\boldsymbol{u}$ to $\boldsymbol{x}_2$. Then compute $\boldsymbol{j}$ of $\boldsymbol{u}$.

Step 11. Find $\boldsymbol{y}$ from $\boldsymbol{j}$ then compare $\boldsymbol{y}$ with $\boldsymbol{u}$. If $\boldsymbol{y}$ and $\boldsymbol{u}$ are exactly same, $\boldsymbol{k} = \boldsymbol{j}$ and repeat steps 9 to 11; otherwise, $\boldsymbol{k}$ is the index of the best codeword to $\boldsymbol{x}_2$ and stop.

The decoding procedure of HRQ is simple:

Step 1. Find $\boldsymbol{y}$ from the received $\boldsymbol{k}$ according to R.

Step 2. Rescale $\boldsymbol{y}$ by the same scaling factor α used in the quantization process: $\boldsymbol{y}_1 = \boldsymbol{y}/\alpha$.

Step 3. Add the same offset $\boldsymbol{a}$ used in step 1 of the quantization process to the rescaled codeword $\boldsymbol{y}_1$: $\boldsymbol{y}_2 = \boldsymbol{y}_1 + \boldsymbol{a}$, and then stop.

The quantization efficiency of HRQ can be further improved by Huffman coding, which is an entropy–coding method that is most useful when the source is unevenly distributed [31]. The transform coefficients are typically unevenly distributed; hence, using Huffman coding can improve the coding efficiency. In HRQ, Huffman coding is used to encode the quantization indices $\boldsymbol{k}$ and reduce the bit requirement.

3.2.3 Lower Rate Quantization Based on RE_8

LRQ is based on RE_8 presented in section 2.2 and is designed to quantize input vectors at the rate of 1 bit/dimension. The Gosset lattice E_8 (RE_8) is the best lattice in 8 dimensions for most purposes [4]. However, from Table 1, the computational complexity of the LVQ scheme based on E_8 (RE_8) is more than 3 times higher compared to that of the LVQ scheme based on D_8. To reduce the complexity, LRQ includes a table–based searching method and a table–based indexing method.

In LRQ the codebook consists of all 256 codewords of the subquantizer Q_2 of EAVQ described in subsection 3.1.2. However, the codewords are arranged in a particular order to develop the fast indexing method (Table 2).

For each 8–dimensional input vector $\boldsymbol{x} = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)$, quantization is performed as follows:

Step 1. Apply an offset $a = 2^{-6}$ to each component of the vector $\boldsymbol{x}$: $\boldsymbol{x}_1 = \boldsymbol{x} - \boldsymbol{a}$, where $a = (2^{-6}, 2^{-6}, 2^{-6}, 2^{-6}, 2^{-6}, 2^{-6}, 2^{-6}, 2^{-6})$.

Step 2. Scale $\boldsymbol{x}_1$ by the scaling factor α: $\boldsymbol{x}_2 = \alpha \boldsymbol{x}_1$. The optimal scaling factor is experimentally chosen.

Step 3. Obtain the new vector $\boldsymbol{x}_3$ by reordering the components of $\boldsymbol{x}_2$ in descending order.

Step 4. In Table 3, find the vector $\boldsymbol{l}$ that best matches $\boldsymbol{x}_3$ in terms of MSE. The vectors given in Table 3 are leaders of the

▼Table 2. Codebook of the lower–rate quantizer

Index	Codeword	Index	Codeword	Index	Codeword	Index	Codeword
0	-2 0 0 0 0 0 0 0	64	0 0 0 -2 0 0 2 0	128	-1 -1 1 1 1 1 1 1	192	1 1 1 -1 1 -1 -1 -1
1	0 -2 0 0 0 0 0 0	65	0 0 0 -2 0 0 0 2	129	-1 1 -1 1 1 1 1 1	193	1 1 1 -1 1 1 1 1
2	0 0 -2 0 0 0 0 0	66	0 0 0 0 -2 2 0 0	130	-1 1 1 -1 1 1 1 1	194	1 1 1 -1 1 -1 1 1
3	0 0 0 -2 0 0 0 0	67	0 0 0 0 -2 0 2 0	131	-1 1 1 1 -1 1 1 1	195	1 1 1 -1 1 1 -1 1
4	0 0 0 0 -2 0 0 0	68	0 0 0 0 -2 0 0 2	132	-1 1 1 1 1 -1 1 1	196	1 1 -1 1 1 -1 -1 -1
5	0 0 0 0 0 -2 0 0	69	0 0 0 0 0 -2 2 0	133	-1 1 1 1 1 1 -1 1	197	1 1 -1 1 1 -1 1 -1
6	0 0 0 0 0 0 -2 0	70	0 0 0 0 0 -2 0 2	134	-1 1 1 1 1 1 1 -1	198	1 1 -1 1 1 1 -1 1
7	0 0 0 0 0 0 0 -2	71	0 0 0 0 0 0 -2 2	135	1 -1 -1 1 1 1 1 1	199	1 1 -1 1 1 1 1 -1
8	2 0 0 0 0 0 0 0	72	2 -2 0 0 0 0 0 0	136	1 -1 1 -1 1 1 1 1	200	1 1 -1 1 1 1 -1 -1
9	0 2 0 0 0 0 0 0	73	2 0 -2 0 0 0 0 0	137	1 -1 1 1 -1 1 1 1	201	1 1 -1 1 1 -1 1 -1
10	0 0 2 0 0 0 0 0	74	2 0 0 -2 0 0 0 0	138	1 -1 1 1 1 -1 1 1	202	1 1 -1 1 1 -1 -1 1
11	0 0 0 2 0 0 0 0	75	2 0 0 0 -2 0 0 0	139	1 -1 1 1 1 1 -1 1	203	1 1 -1 1 -1 -1 1 1
12	0 0 0 0 2 0 0 0	76	2 0 0 0 0 -2 0 0	140	1 -1 1 1 1 1 1 -1	204	1 1 -1 1 -1 1 1 1
13	0 0 0 0 0 2 0 0	77	2 0 0 0 0 0 -2 0	141	1 1 -1 -1 1 1 1 1	205	1 1 -1 -1 1 1 1 -1
14	0 0 0 0 0 0 2 0	78	2 0 0 0 0 0 0 -2	142	1 1 -1 1 -1 1 1 1	206	1 1 -1 1 -1 1 1 -1
15	0 0 0 0 0 0 0 2	79	0 2 -2 0 0 0 0 0	143	1 1 -1 1 1 -1 1 1	207	1 1 -1 1 1 -1 -1 1
16	2 -2 0 0 0 0 0 0	80	0 2 0 -2 0 0 0 0	144	1 1 -1 1 1 1 -1 1	208	1 1 -1 1 1 1 1 -1
17	2 0 -2 0 0 0 0 0	81	0 2 0 0 -2 0 0 0	145	1 1 1 -1 1 1 1 -1	209	1 1 1 -1 -1 1 1 1
18	2 0 0 -2 0 0 0 0	82	0 2 0 0 0 -2 0 0	146	1 1 1 -1 -1 1 1 1	210	1 1 1 -1 1 -1 1 1
19	2 0 0 0 -2 0 0 0	83	0 2 0 0 0 0 -2 0	147	1 1 1 -1 1 -1 1 1	211	1 1 1 -1 1 1 -1 1
20	2 0 0 0 0 -2 0 0	84	0 2 0 0 0 0 0 -2	148	1 1 1 -1 1 1 -1 1	212	1 1 1 -1 1 1 1 -1
21	2 0 0 0 0 0 -2 0	85	0 0 2 -2 0 0 0 0	149	1 1 1 1 -1 1 1 1	213	1 1 1 1 -1 -1 1 1
22	2 0 0 0 0 0 0 -2	86	0 0 2 0 -2 0 0 0	150	1 1 1 1 -1 -1 1 1	214	1 1 1 1 -1 1 -1 1
23	0 -2 2 0 0 0 0 0	87	0 0 2 0 0 -2 0 0	151	1 1 1 1 -1 1 -1 1	215	1 1 1 1 -1 1 1 -1
24	0 -2 0 2 0 0 0 0	88	0 0 2 0 0 0 -2 0	152	1 1 1 1 -1 1 1 -1	216	1 1 1 1 -1 1 1 1
25	0 -2 0 0 2 0 0 0	89	0 0 2 0 0 0 0 -2	153	1 1 1 1 1 -1 1 1	217	1 1 1 1 1 -1 -1 1
26	0 -2 0 0 0 2 0 0	90	0 0 0 2 -2 0 0 0	154	1 1 1 1 1 -1 1 -1	218	1 1 1 1 1 -1 1 1
27	0 -2 0 0 0 0 2 0	91	0 0 0 2 0 -2 0 0	155	1 1 1 1 1 1 -1 -1	219	1 1 1 1 1 1 -1 1
28	0 -2 0 0 0 0 0 2	92	0 0 0 2 0 0 -2 0	156	1 1 1 1 1 1 1 -1	220	1 1 1 1 1 1 1 1
29	0 0 -2 2 0 0 0 0	93	0 0 0 2 0 0 0 -2	157	1 1 1 1 1 1 1 1	221	1 1 1 1 1 1 1 1
30	0 0 -2 0 2 0 0 0	94	0 0 0 0 2 -2 0 0	158	1 1 1 1 1 1 1 1	222	1 1 1 1 1 1 1 1
31	0 0 -2 0 0 -2 0 0	95	0 0 0 0 2 0 -2 0	159	1 1 1 1 1 1 1 1	223	1 1 1 1 1 1 1 1
32	0 0 -2 0 0 0 -2 0	96	0 0 0 0 2 0 0 -2	160	1 1 1 1 1 1 1 1	224	1 1 1 1 1 1 1 1
33	0 0 -2 0 0 0 0 2	97	0 0 0 0 0 2 -2 0	161	1 1 1 1 1 1 1 1	225	1 1 1 1 1 1 1 1
34	0 0 0 -2 -2 0 0 0	98	0 0 0 0 0 2 0 -2	162	1 1 1 1 1 1 1 1	226	1 1 1 1 1 1 1 1
35	0 0 0 -2 0 2 0 0	99	0 0 0 0 0 0 2 -2	163	1 1 1 1 1 1 1 1	227	1 1 1 1 1 1 1 1
36	0 0 0 -2 0 0 2 0	100	2 2 0 0 0 0 0 0	164	1 1 1 1 1 1 1 1	228	1 1 1 1 1 1 1 1
37	0 0 0 -2 0 0 0 2	101	2 0 2 0 0 0 0 0	165	1 1 1 1 1 1 1 1	229	1 1 1 1 1 1 1 1
38	0 0 0 0 -2 2 0 0	102	2 0 0 2 0 0 0 0	166	1 1 1 1 1 1 1 1	230	1 1 1 1 1 1 1 1
39	0 0 0 0 -2 0 2 0	103	2 0 0 0 2 0 0 0	167	1 1 1 1 1 1 1 1	231	1 1 1 1 1 1 1 1
40	0 0 0 0 -2 0 0 2	104	2 0 0 0 0 2 0 0	168	1 1 1 1 1 1 1 1	232	1 1 1 1 1 1 1 1
41	0 0 0 0 0 -2 2 0	105	2 0 0 0 0 0 2 0	169	1 1 1 1 1 1 1 1	233	1 1 1 1 1 1 1 1
42	0 0 0 0 0 -2 0 2	106	2 0 0 0 0 0 0 2	170	1 1 1 1 1 1 1 1	234	1 1 1 1 1 1 1 1
43	0 0 0 0 0 0 -2 2	107	0 2 2 0 0 0 0 0	171	1 1 1 1 1 1 1 1	235	1 1 1 1 1 1 1 1
44	2 2 0 0 0 0 0 0	108	0 2 0 2 0 0 0 0	172	1 1 1 1 1 1 1 1	236	1 1 1 1 1 1 1 1
45	2 0 2 0 0 0 0 0	109	0 2 0 0 2 0 0 0	173	1 1 1 1 1 1 1 1	237	1 1 1 1 1 1 1 1
46	2 0 0 2 0 0 0 0	110	0 2 0 0 0 2 0 0	174	1 1 1 1 1 1 1 1	238	1 1 1 1 1 1 1 1
47	2 0 0 0 2 0 0 0	111	0 2 0 0 0 0 2 0	175	1 1 1 1 1 1 1 1	239	1 1 1 1 1 1 1 1
48	2 0 0 0 0 2 0 0	112	0 2 0 0 0 0 0 2	176	1 1 1 1 1 1 1 1	240	1 1 1 1 1 1 1 1
49	2 0 0 0 0 0 2 0	113	0 0 2 2 0 0 0 0	177	1 1 1 1 1 1 1 1	241	1 1 1 1 1 1 1 1
50	2 0 0 0 0 0 0 2	114	0 0 2 0 2 0 0 0	178	1 1 1 1 1 1 1 1	242	1 1 1 1 1 1 1 1
51	0 -2 2 0 0 0 0 0	115	0 0 2 0 0 2 0 0	179	1 1 1 1 1 1 1 1	243	1 1 1 1 1 1 1 1
52	0 -2 0 2 0 0 0 0	116	0 0 2 0 0 0 2 0	180	1 1 1 1 1 1 1 1	244	1 1 1 1 1 1 1 1
53	0 -2 0 0 2 0 0 0	117	0 0 2 0 0 0 0 2	181	1 1 1 1 1 1 1 1	245	1 1 1 1 1 1 1 1
54	0 -2 0 0 0 2 0 0	118	0 0 0 2 2 0 0 0	182	1 1 1 1 1 1 1 1	246	1 1 1 1 1 1 1 1
55	0 -2 0 0 0 0 2 0	119	0 0 0 2 0 2 0 0	183	1 1 1 1 1 1 1 1	247	1 1 1 1 1 1 1 1
56	0 -2 0 0 0 0 0 2	120	0 0 0 2 0 0 2 0	184	1 1 1 1 1 1 1 1	248	1 1 1 1 1 1 1 1
57	0 0 -2 2 0 0 0 0	121	0 0 0 2 0 0 0 2	185	1 1 1 1 1 1 1 1	249	1 1 1 1 1 1 1 1
58	0 0 -2 0 2 0 0 0	122	0 0 0 0 2 2 0 0	186	1 1 1 1 1 1 1 1	250	1 1 1 1 1 1 1 1
59	0 0 -2 0 0 2 0 0	123	0 0 0 0 2 0 2 0	187	1 1 1 1 1 1 1 1	251	1 1 1 1 1 1 1 1
60	0 0 -2 0 0 0 2 0	124	0 0 0 0 2 0 0 2	188	1 1 1 1 1 1 1 1	252	1 1 1 1 1 1 1 1
61	0 0 -2 0 0 0 0 2	125	0 0 0 0 0 2 2 0	189	1 1 1 1 1 1 1 1	253	1 1 1 1 1 1 1 1
62	0 0 0 -2 2 0 0 0	126	0 0 0 0 0 2 0 2	190	1 1 1 1 1 1 1 1	254	1 1 1 1 1 1 1 1
63	0 0 0 -2 0 2 0 0	127	0 0 0 0 0 0 2 2	191	1 1 1 1 -1 -1 -1 -1	255	1 1 1 1 1 1 1 1

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P33

▼Table 3. Leaders of the codewords of LRQ

Index	Leader
0	0 0 0 0 0 0 0 –2
1	2 0 0 0 0 0 0 0
2	0 0 0 0 0 0 –2 –2
3	2 0 0 0 0 0 0 –2
4	2 2 0 0 0 0 0 0
5	1 1 1 1 1 1 –1 –1
6	1 1 1 1 –1 –1 –1 –1
7	1 1 –1 –1 –1 –1 –1 –1
8	–1 –1 –1 –1 –1 –1 –1 –1
9	1 1 1 1 1 1 1 1

▼Table 4. Flag vectors and index offsets of the leaders

Leader Index	Flag Vector	Index Offset
0	0 0 0 0 0 0 0 1	0
1	1 0 0 0 0 0 0 0	8
2	0 0 0 0 0 0 1 1	16
3	1 0 0 0 0 0 0 1	44
4	1 1 0 0 0 0 0 0	100
5	0 0 0 0 0 0 1 1	128
6	0 0 0 0 1 1 1 1	128
7	0 0 1 1 1 1 1 1	128
8	1 1 1 1 1 1 1 1	128
9	0 0 0 0 0 0 0 0	128

codewords, and any codeword in the codebook can be generated by permutation of its leader.

Step 5. Obtain the best codeword $\boldsymbol{y}$ by reordering the components of $\boldsymbol{l}$ in the original order.

Step 6. Find the flag vector of $\boldsymbol{l}$ in Table 4 and obtain the vector $\boldsymbol{z}$ by reordering the components of the flag vector in the original order. The flag vectors are defined as follows:

- If the leader consists of –2, 2, and 0, then –2 and 2 are indicated by 1, and 0 is indicated by 0
- If the leader consists of –1 and 1, then –1 is indicated by 1, and 1 is indicated by 0.

Step 7. Find the index offset K related to the leader $\boldsymbol{l}$ in Table 4.

Step 8. If $\boldsymbol{l}$ is (2, 0, 0, 0, 0, 0, 0, –2) and $\boldsymbol{y}$ has the component 2 with index lower than that of the component –2, the offset K is adjusted so that $K = K + 28$.

Step 9. Compute the vector dot product $i = \boldsymbol{z}\boldsymbol{p}^T$, where $\boldsymbol{p}$ = (1, 2, 4, 8, 16, 32, 64, 128).

Step 10. From i, find the index increment j related to $\boldsymbol{y}$ in Table 5.

Step 11. Compute the index k of $\boldsymbol{y}$: $k = K + j$, and then stop.

The following are the steps taken in the decoding procedure of LRQ:

Step 1. Find the codeword $\boldsymbol{y}$ in Table 2 from the received index k.

Step 2. Rescale the codeword $\boldsymbol{y}$ by the same scaling factor α used in the quantization process: $\boldsymbol{y}_1 = \boldsymbol{y}/\alpha$.

Step 3. Add the same offset a used in step 1 of the encoding procedure to the rescaled codeword $\boldsymbol{y}_1$: $\boldsymbol{y}_2 = \boldsymbol{y}_1 + \boldsymbol{a}$, and then stop.

▼Table 5. Index increments related to the codewords of LRQ

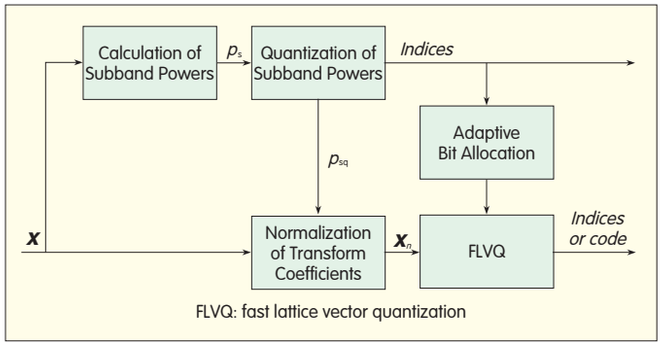
Index	Increment	Index	Increment	Index	Increment	Index	Increment
0	127	64	0	128	7	192	37
1	0	65	5	129	4	193	0
2	1	66	11	130	12	194	0
3	0	67	0	131	0	195	40
4	2	68	16	132	17	196	0
5	1	69	0	133	0	197	50
6	7	70	0	134	0	198	92
7	0	71	33	135	32	199	0
8	3	72	10	136	21	200	0
9	2	73	0	137	0	201	60
10	8	74	0	138	0	202	82
11	0	75	35	139	36	203	0
12	13	76	0	140	0	204	72
13	0	77	45	141	46	205	0
14	0	78	90	142	91	206	0
15	28	79	0	143	0	207	120
16	4	80	23	144	24	208	0
17	3	81	0	145	0	209	54
18	9	82	0	146	0	210	79
19	0	83	38	147	39	211	0
20	14	84	0	148	0	212	69
21	0	85	48	149	49	213	0
22	0	86	96	150	97	214	0
23	29	87	0	151	0	215	113
24	18	88	0	152	0	216	85
25	0	89	58	153	59	217	0
26	0	90	86	154	87	218	0
27	31	91	0	155	0	219	113
28	0	92	76	156	77	220	0
29	43	93	0	157	0	221	108
30	86	94	0	158	0	222	103
31	0	95	124	159	123	223	0
32	5	96	35	160	26	224	0
33	4	97	0	161	0	225	53
34	10	98	0	162	0	226	78
35	0	99	42	163	41	227	0
36	15	100	0	164	0	228	68
37	0	101	52	165	51	229	0
38	0	102	94	166	95	230	0
39	30	103	0	167	0	231	116
40	19	104	0	168	0	232	84
41	0	105	62	169	61	233	0
42	0	106	84	170	85	234	0
43	34	107	0	171	0	235	112
44	0	108	74	172	73	236	0
45	44	109	0	173	0	237	107
46	88	110	0	174	0	238	101
47	0	111	122	175	121	239	0
48	22	112	0	176	0	240	63
49	0	113	56	177	55	241	0
50	0	114	81	178	80	242	0
51	33	115	0	179	0	243	111
52	0	116	71	180	70	244	0
53	47	117	0	181	0	245	106
54	95	118	0	182	0	246	100
55	0	119	118	183	118	247	0
56	0	120	67	184	66	248	0
57	57	121	0	185	0	249	105
58	85	122	0	186	0	250	99
59	0	123	115	187	114	251	0
60	75	124	0	188	0	252	98
61	0	125	110	189	109	253	0
62	0	126	104	190	103	254	0
63	125	127	0	191	0	255	126

LRQ: lower–rate quantizer

3.2.4 Application to Low–Complexity Full–Band Audio Coding

FLVQ has been applied to 20 kHz audio coding in ITU–T Recommendation G.719 [19]. ITU–T G.719 is the first

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P34



▲Figure 4. FLVQ applied to audio coding in G.719.

▼Table 6. Computational complexity of FLVQ in G.719

Bit Rate (kbit/s)	Quantization		De–Quantization		Total	
	Average	Maximum	Average	Maximum	Average	Maximum
32	1.644	2.837	0.808	0.987	2.452	3.824
48	2.336	3.861	1.143	1.436	3.479	5.297
64	2.844	4.120	1.457	1.804	4.301	5.924

full–band audio codec of ITU–T and was developed for low–complexity full–band audio coding for high–quality conversational applications. The G.719 codec is based on transform coding and operates on frames of 20 ms corresponding to 960 samples at a sampling rate of 48 kHz. The codec provides an audio bandwidth of 20 Hz to 20 kHz, operating from 32 kbit/s up to 128 kbit/s, and has an algorithmic delay of 40 ms. The G.719 codec features very high audio quality and extremely low computational complexity compared with other state–of–the–art audio coding algorithms. It is suitable for use in applications such as videoconferencing, telepresence, teleconferencing, streaming audio over the Internet, and IPTV.

In the G.719 encoder, the input audio signal sampled at 48 kHz is converted by an adaptive time–frequency transform [19] from the time domain into the frequency domain. For every 20 ms, the input audio samples are transformed into 960 transform coefficients. FLVQ is used to quantize transform coefficients $\boldsymbol{x}$ (Fig. 4).

After the transform, the obtained transform coefficients are grouped into sub–bands— 8, 16, 24, or 32—of unequal length. Because the bandwidth is 20 kHz, only 800 transform coefficients are used. The 160 transform coefficients representing frequencies above 20 kHz are ignored. The power p_s of each sub–band is defined as the root–mean–square value of the subband and is given by

$$p_s=\sqrt{\frac{1}{N}\sum_{i=0}^{N-1}x_i^2},\ 0\leqslant s\leqslant 43 \tag{14}$$

where $\boldsymbol{x}$ is the transform coefficients, and N is the number of coefficients in the sub–bands, that is, 8, 16, 24, and 32. The resulting spectral envelope comprising the powers of all sub–bands is quantized and encoded. An adaptive bit–allocation scheme based on the quantized powers of the sub–bands is used to assign the available bits in a frame among the sub–bands. The number of bits assigned to each transform coefficient can be as large as 9 bits depending on the input signal. In each sub–band, the transform coefficients are normalized by the quantized powers p_{sq}.

Each sub–band consists of one or more vectors of 8–dimensional coefficients. Thus, the normalized coefficients $\boldsymbol{x}_n$ are quantized in 8–dimensional vectors by using FLVQ previously mentioned. If a sub–band is assigned 1 bit per coefficient, then the lower–rate quantizer LRQ is used to quantize the normalized coefficients of the sub–band; otherwise, the coefficients are quantized by the higher–rate quantizer HRQ. The 8–dimensional coefficient vectors have a high concentration of probability around the origin; therefore, Huffman coding is an option for the quantization indices of HRQ. When the rate is smaller than 6 bit/coefficient, the total of the bits needed for all sub–bands is added. If the Huffman coded bits are less than the allocated bits, Huffman coding is applied to the quantization indices, and a Huffman code flag is set. The saved bits are used to quantize the coefficients of the sub–bands assigned 0 bit. If the Huffman coded bits are not less than the allocated bits, then Huffman coding is not used, and the Huffman code flag is cleared. In each case, the Huffman code flag is transmitted as side information to the decoder. In this way, the best coding method is used.

Table 6 shows the computational complexity of FLVQ in 16/32–bit fixed–point for some bit rates in G.719. Computational complexity is measured in units of weighted million operations per second (WMOPS) by using the basic operators of ITU–T Software Tool Library STL2005 v2.2 in ITU–T G.191 [32]. The ROM memory usage of the LRQ and HRQ tables is shown in Table 7.

Low computational complexity and storage requirements are a major advantage of using FLVQ for transform–based audio coding.

In February 2008, subjective tests for the ITU–T G.719 Optimization/Characterization phase were performed by independent listening laboratories in English, French, and Spanish according to a test plan designed by ITU–T Q7/SG12 Speech Quality Experts Group (SQEG) [33]. Statistical analysis of the test results showed that the G.719 codec met all performance requirements [34]. An additional subjective listening test for G.719 was conducted later to evaluate the quality of the codec at rates higher than those described in the ITU–T test plan [35]. These test results showed that transparency was reached for critical material at 128 kbit/s.

The computational complexity of the G.719 codec in 16/32–bit fixed–point was estimated by encoding and decoding the source material used for the subjective test of the G.719 Optimization/Characterization phase. Using FLVQ, the computational complexity of G.719 is quite low, for

▼Table 7. ROM memory usage of FLVQ in G.719 (in 16–bit words)

LRQ	HRQ	Total
2554	40	2594

LRQ: lower rate quantization HRQ: higher rate quantizer

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P35

example, 15.397 WMOPS at 32 kbit/s, 18.060 WMOPS at 64 Kbit/s, and 21.000 WMOPS at 128 kbit/s [19].

4 Conclusion

LVQ has many advantages and is suitable for use in low–complexity transform–based speech and audio coding.

Embedded algebraic vector quantization (EAVQ) has been applied to speech and audio coding to efficiently quantize spectral vectors in transform coding, for example, TCX coding. Based on the EAVQ technique, split multirate LVQ has been developed and successfully used in several speech and audio coding standards, including 3GPP AMR–WB+, ITU–T G.718, G.711.1 Annex D, G.722 Annex B, and MPEG Unified Speech and Audio Coding (USAC).

Fast lattice vector quantization (FLVQ) has been applied to low–complexity full–band audio coding in ITU–T Recommendation G.719 and is designed to quantize transform coefficients in transform coding. The fast encoding algorithm is used, and an efficient method for quantizing outliers has been developed. Hence, the computational complexity of G.719 is quite low. In addition, Huffman coding is optionally applied to quantization indices to further improve the efficiency of the quantizer.

Acknowlegement

The author would like to thank Dr. Stéphane Ragot for valuable comments and discussions on this paper. The author also thanks the reviewers for their helpful suggestions in improving the presentation of the paper.

References

[1] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA: Kluwer, 1992.
[2] A. Gersho, "Asymptotically optimal block quantization," *IEEE Trans. Inform.* Theory, vol. IT–25, pp. 373–380, July 1979.
[3] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM–28, pp. 84–95, Jan. 1980.
[4] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices, and Groups*. New York: Springer–Verlag, 1988.
[5] M. V. Eyuboglu and G. D. Forney Jr., "Lattice and trellis quantization with lattice and trellis–bounded codebooks—High–rate theory for memoryless sources," *IEEE Trans. Inform. Theory*, vol. 39, pp. 46–59, Jan. 1993.
[6] J. D. Gibson and K. Sayood, "Lattice quantization," in *Advances in Electronics and Electron Physics*, vol. 72, pp. 259–330, 1988.
[7] T. R. Fischer, "A pyramid vector quantizer," *IEEE Trans. Inform. Theory*, vol. 32, pp. 568–583, July 1986.
[8] M. Barlaud, P. Solé, T. Gaidon, M. Antonini, and P. Mathieu, "Pyramidal lattice vector quantization for multiscale image coding," *IEEE Trans. Image Processing*, vol. 3, pp. 367–381, July 1994.
[9] Z. M. Yusof and T. Fischer, "An entropy–coded lattice vector quantizer for transform and subband image coding," *IEEE Trans. Image Processing*, vol. 5, pp. 289–298, Feb. 1996.
[10] P. Raffy, M. Antonini, and M. Barlaud, "Distortion–rate models for entropy–coded lattice vector quantization," *IEEE Trans. Image Processing*, vol. 9, pp. 2006–2017, Dec. 2000.
[11] C. Lamblin, J. P. Adoul, D. Massaloux, and S. Morissette, "Fast CELP coding based on the Barnes–Wall lattice in 16 dimensions," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Glasgow, Scotland, UK, May 1989, vol. 1, pp. 61–64.
[12] M. Xie and J.–P. Adoul, "Algebraic vector quantization of LSF parameters with low storage and computational complexity," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 4, pp. 234–239, May 1996.
[13] M. Xie and J.–P. Adoul, "Embedded algebraic vector quantizers (EAVQ) with application to wideband speech coding," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Atlanta, Georgia, May 1996, vol. 1, pp. 240–243.
[14] S. Ragot, R. Lefebvre, R. Salami, and J.–P. Adoul, "Stochastic–algebraic wideband LSF quantization," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Istanbul, Turkey, June 2000, vol. 2, pp. 1169–1172.
[15] S. Ragot, M. Xie, and R. Lefebvre, "Near–Ellipsoidal Voronoi Coding," *IEEE Trans. Inform. Theory*, vol. 49, pp. 1815–1820, July. 2003.
[16] S. Ragot, B. Bessette, and R. Lefebvre, "Low–complexity multi–rate lattice vector quantization with application to wideband TCX speech coding at 32 kbit/s," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Montreal, Canada, May 2004, vol. 1, pp. 501–504.
[17] M. Xie, A. Taleb, M. Briand, and P. Chu, "ITU–T G.719: A New Low–Complexity Full–Band (20 kHz) Audio Coding Standard For High–Quality Conversational Applications," in *Proceedings of 2009 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, New York, October 2009, pp. 265–268.
[18] *3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Audio codec processing functions; Extended Adaptive Multi–Rate – Wideband (AMR–WB+) codec; Transcoding functions*, 3GPP TS 26.290 V6.0.0, Mar. 2005.
[19] *Low–complexity full–band audio coding for high–quality conversational applications*, ITU–T Recommendation G.719, June 2008.
[20] *Frame error robust narrowband and wideband embedded variable bit–rate coding of speech and audio from 8–32 kbit/s*,ITU–T Recommendation G.718, June 2008.
[21] *Wideband embedded extension for G.711 PCM: New Annex D with superwideband extension*, ITU–T Recommendation G.711 Annex D, November 2010.
[22] *7 kHz audio–coding within 64 kbit/s: New Annex B with superwideband embedded extension*, ITU–T Recommendation G.722 Annex B, November 2010.
[23] *MPEG audio technologies–Part 3: Unified speech and audio coding*, ISO/IEC FDIS 23003–3:2011, 2011.
[24] J. H. Conway and N. J. A. Sloane, "Fast quantizing and decoding algorithms for lattice quantizers and codes," *IEEE Trans. Inform. Theory*, vol. IT–28, pp. 227–232, Mar. 1982.
[25] J. H. Conway and N. Sloane, "A fast encoding method for lattice codes and quantizers," *IEEE Trans. Inform. Theory*, vol. IT–29, pp. 820–824, Nov. 1983.
[26] J.–P. Adoul, "La quantification vectorielle des signaux : approche algébrique," *Annales des Télécommunications*, 41, pp. 158–177, 1986.
[27] C. Lamblin and J.–P. Adoul, "Algorithme de quantification vectorielle sphérique à partir du réseau de Gosset d'ordre 8," *Annales des Télécommunications*, 43, pp. 172–186, 1988.
[28] R. Lefebvre, R. Salami, C. Laflamme, and J.–P. Adoul, "High quality coding of wideband audio signals using Transform Coded eXcitation (TCX)," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Adelaide, South Australia, Apr. 1994, vol. 1, pp. 193–196.
[29] J.–P. Adoul and R. Lefebvre, *Speech Coding and Synthesis*, Chapter on Wideband Speech Coding. Amsterdam: Elsevier, 1995.
[30] S. Ragot, "New techniques of algebraic vector quantization based on Voronoi coding – Application to AMR–WB+ coding," Ph.D. dissertation (in French), University of Sherbrooke, Sherbrooke, Quebec, Canada, May 2003.
[31] D. A. Huffman, "A method for the construction of minimum–redundancy codes," in *Proceedings of IRE*, vol. 40, pp. 1098–1101, Sept. 1952.
[32] Software tools for speech and audio coding standardization, ITU–T Recommendation G.191, Sept. 2005.
[33] *G.722.1 Fullband extension optimization/ characterization Quality Assessment Test Plan*, ITU–T WP3/SG16 TD–323, Apr. 2008.
[34] *Reply LS on speech and audio coding matters (Addendum)*, ITU–T GEN/SG16 TD–525, Apr. 2008.
[35] *Additional information about the subjective performance of the G.722.1FB codec*, ITU–T SG12 COM12–C–165, , May 2008.

Manuscript received: January 16, 2012

Biography

Minjie Xie (minjie.xie@zteusa.com) received his MASc and PhD degrees in electrical engineering from the University of Sherbrooke, Quebec, Canada, in 1993 and 1996. He is a senior standard specialist at ZTE USA Inc., where he works on speech and audio coding standardization and algorithm development. Since 2002, he has been active in speech and audio coding standardization in ITU–T, MPEG, and 3GPP. From 2006 to 2008, he was the editor of ITU–T Recommendation G.722.1 and its Annexes and is currently the editor of the 3GPP SA4 EVS Codec. He was a major contributor to ITU–T Recommendations G.722.1C, the first ITU–T superwideband audio coding standard, and G.719, the first ITU–T full–band audio coding standard. His research interests include speech and audio coding, speech processing, vector quantization, and data compression. He has several patents and has published in several journal articles and conference proceedings.

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT———1PPS/P36

Noise Feedback Coding Revisited: Refurbished Legacy Codecs and New Coding Models

Stéphane Ragot, Balázs Kövesi, and Alain Le Guyader
(audiOvisual and sPEech foR quAlity (OPERA) Lab., Orange Labs, Lannion, France)

Abstract

Noise feedback coding (NFC) has attracted renewed interest with the recent standardization of backward–compatible enhancements for ITU–T G.711 and G.722. It has also been revisited with the emergence of proprietary speech codecs, such as BV16, BV32, and SILK, that have structures different from CELP coding. In this article, we review NFC and describe a novel coding technique that optimally shapes coding noise in embedded pulse–code modulation (PCM) and embedded adaptive differential PCM (ADPCM). We describe how this new technique was incorporated into the recent ITU–T G.711.1, G.711 App. III, and G.722 Annex B (G.722B) speech–coding standards.

Keywords

speech coding; noise shaping; noise feedback coding; G.711; G.722

1 Introduction

Noise shaping is a key technique for improving sound quality in telecommunications and multimedia. Analog noise reduction systems relied on companding/expanding in frequency bands to reduce recording noise from magnetic tapes [1]. Pulse–code modulation (PCM) in ITU–T G.711 uses companding/expanding on the time domain magnitude to achieve a near–constant signal–to–noise ratio (SNR) over a large range of input levels [2]. Even if PCM does not involve any noise shaping in the frequency domain, it exploits the human perception of an audio signal on a log scale.

In this paper, we describe noise feedback coding (NFC), a well–known noise shaping technique [3]–[5], [6], [7]. The goal of NFC is to modify the input signal of scalar quantization by feeding back the filtered coding noise. NFC was introduced to shape the spectrum of the quantization noise in sample–based waveform coders such as PCM coders. A similar principle is used in sigma–delta analog–to–digital (A/D) converters, except that the filter is in the feedforward path [8] instead of the feedback path. NFC has attracted renewed interest with the recent standardization of backward–compatible enhancements to ITU–T G.711 and G.722 [9], [10]. It has also been revisited with the emergence of proprietary speech codecs such as BroadVoice (BV16 and BV32) [11], [12] and SILK [13], that have structures not based on code–excited linear prediction (CELP) coding.

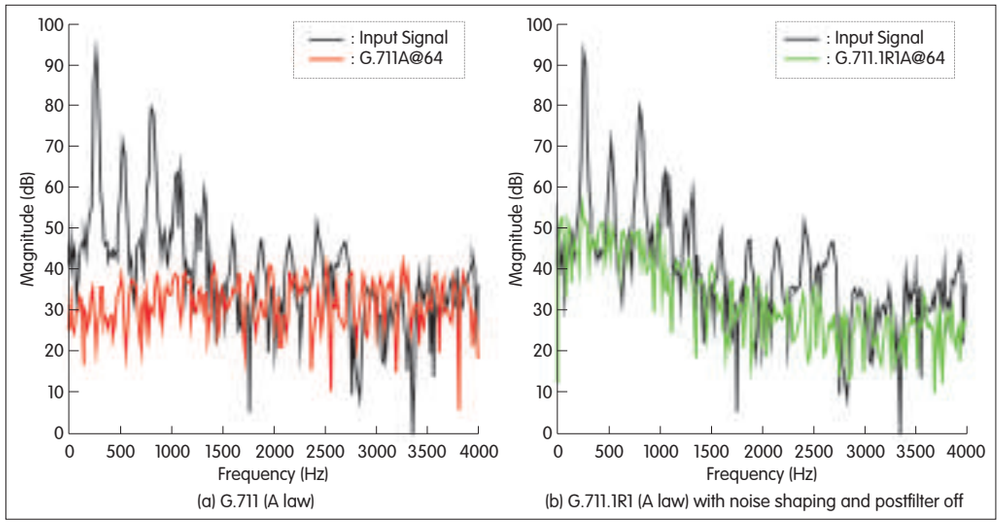
In this paper, we give an overview of NFC and describe a novel speech coding method that optimally combines NFC with embedded PCM or adaptive differential PCM (ADPCM). Important applications include the recent backward–compatible enhancements to G.711 and G.722.

In section 2, the principles of and approaches to noise shaping are discussed. In section 3, NFC is discussed in detail. In section 4, embedded coding and noise shaping enhancements to G.711 are described. In section 5, a counterpart to G.722 is proposed. In section 6, new codec structures inspired by NFC are discussed. Section 7 concludes the paper.

2 Noise Shaping Principles and Approaches

The basic idea of noise shaping is to exploit the limitations of the human auditory system, in particular, masking properties, to make coding noise inaudible. Different methods of noise shaping draw on the principles of psychoacoustics [14], [15]. In this section, we give an overview of noise shaping in speech coding but not noise shaping in perceptual audio coding. In perceptual audio coding, noise shaping is performed in the frequency or sub–band domains by a

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P37



▲Figure 1. Spectrum of a narrowband female speech sample coded at 64 kbit/s and resulting noise with and without noise shaping.

masking model that determines appropriate bit allocation [16]–[18].

2.1 Noise Shaping in Speech Coding: G. 711

Fig. 1 shows noise shaping for a real–speech sample taken from a female who was speaking French. Two short–term spectral noise shapes are shown for PCM codecs operating at 64 kbit/s. In Fig. 1(a), 64 kbit/s PCM coding in G.711 (A law) is used [19]. The coding noise spectrum, shown in red, is nearly flat, and coding noise fills the low–energy spectral valleys of the input signal, which means the noise is clearly audible. In Fig. 1(b), a modified G.711 encoder is used that is backward–compatible with G.711 decoders. The modified G.711 encoder is defined in G.711.1 and has the same 64 kbit/s bit rate as G.711 [9], [19]. The coding noise spectrum, shown in green, approximates the envelope of the signal spectrum, which means noise is present but barely audible because it has been properly shaped.

From Fig. 1(a) and (b), a principle can be derived for perceptually optimizing speech codecs: The coding noise spectrum should approximate the signal spectrum so that a sufficient signal–to–noise ratio is maintained over each frequency range, including ranges where the signal has low energy.

2.2 Noise Shaping at the Encoder: Predictive Speech Coding

Noise shaping in speech coding is implemented according to the underlying coding model. For time–domain waveform coders such as PCM and ADPCM coders, noise can be shaped using NFC [3], [5]; however, related standards, such as G.711 and G.722, do not incorporate NFC [2], [20].

The most common low–bit–rate speech coders are based on linear predictive coding (LPC) (short–term prediction) and pitch (long–term prediction). To the best of our knowledge, the first temporal predictive coder that included a short–term and long–term adaptive predictor was the adaptive predictive coder (APC) [21]. In this coder, long–term prediction is performed first. If short–term prediction was performed first, the CELP synthesis model would be obtained [22]. Atal and Schroeder recognized the necessity of shaping the quantization noise spectrum:

"The quality of the reconstructed speech can thus be improved by a suitable shaping of the spectrum of the quantizing noise so that the signal–to–noise ratio (SNR) is more or less uniform over the entire frequency range of the input speech signal." [21]

To achieve this goal, they suggested using a fixed pre–emphasis filter at the encoder side and an inverse filter at the decoder side.

Two important contributions to noise masking in speech coders were [23], [24], and these were extended by [25]–[27]. These contributions showed that noise could be shaped at the encoder by subtracting the filtered noise from the quantizer input in a feedback loop. Hence, the technique was called noise feedback coding.

The main significance of [25] was that it showed how quantization noise $Q(z)$ of an APC coder could be shaped by a prediction filter $B(z)$–1 and how a linear predictive feedback of noise could be added to the input signal to create the quantizer input that is forwarded to an APC coder with noise shaping (APC–NS) [25, Fig. 9]. The reconstructed speech $\tilde{S}(z)$ in the local decoder and in the (distant) decoder is obtained by adding the quantization noise, which is filtered by a moving–average (MA) FIR filter, to the input speech. The reconstruction speech is given by

$$\tilde{S}(z) = S(z) + B(z)Q(z) \quad (1)$$

where $S(z)$ is the input speech, $B(z)$ is the filtered quantization noise, and $Q(z)$ is the quantization noise.

In [6] and [7], the noise masking in ADPCM coding is implemented as in [25, Fig. 9]. In [26], the quantization noise is filtered by a predictor $A(z)$, and the reconstruction noise is filtered by a predictor $B(z)$. These noises are subtracted to the input signal in the feedback loop [26, Fig. 3]. Using the notation in [26], an autoregressive moving average (ARMA) noise shape is given by

$$\tilde{S}(z) = S(z) + \frac{1-A(z)}{1-B(z)} Q(z) \quad (2)$$

The configuration in [26, Fig. 3] leads to the same noise shaping as that in [25, Fig. 9] when $B(z)$ is set to 0. An improvement to [26, Fig. 3] is shown in [26, Fig. 7], and this improvement gives rise to a generalized predictive coder,

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P38

including a long–term predictor, with the following short–term noise–shaping characteristics:

$$\tilde{S}(z) = S(z) + \frac{1-F_s(z)}{1-P_s(z)} Q(z) \quad (3)$$

where $P_s(z)$ is the linear predictor filter:

$$P_s(z) = \sum_{k=1}^{P} a_k z^{-k} \quad (4)$$

and $F_s(z)$ is the weighted filter derived from $P_s(z)$ using the weighting factor γ:

$$F_s(z) = P_s(z/\gamma) = \sum_{k=1}^{P} a_k \gamma^k z^{-k} \quad (5)$$

The ARMA noise shaping of the quantization noise $Q(z)$ is usually done by the more flexible filter $H_s(z)$, given by

$$H_s(z) = \frac{A(z/\gamma_2)}{A(z/\gamma_1)} = \frac{1+\sum_{k=1}^{P} a_k \gamma_2^k z^{-k}}{1+\sum_{k=1}^{P} a_k \gamma_1^k z^{-k}} \quad (6)$$

The same type of noise–shaping filter has been used in the subsequent generation of speech coders, known as analysis–by–synthesis (AbS) coders, first in multipulse coding [28] and then in CELP coding [22]. AbS is equivalent to minimizing the following CELP criterion [29]:

$$\varepsilon = \frac{1}{2\pi j} \oint_C \left| \left[S(z) - \frac{\hat{C}(z) + g_c C_k(z)}{(1-g_P z^{-T})\hat{A}(z)} \right] W(z) \right|^2 \frac{dz}{z} \quad (7)$$

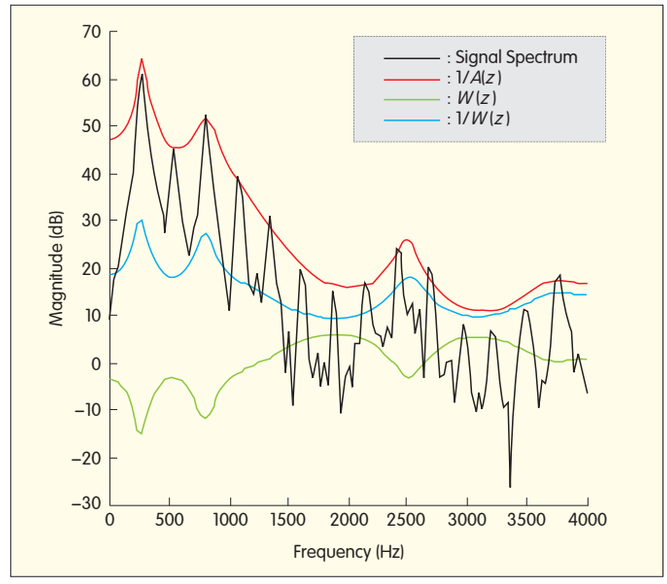
where $\hat{C}(z)$ is the past reconstructed excitation signal, g_c is the fixed codebook gain, $C_k(z)$ is the code vector, g_P is the adaptive codebook gain, T is the adaptive codebook lag, $\hat{A}(z)$ is the quantized LPC filter, and $W(z)$ is the noise weighting filter (also called perceptual weighting filter). The distortion criterion ε is minimized in order to whiten the weighted reconstructed noise (in square brackets, equation 7) and create a coding noise with a spectrum shape $|H_s(z)|^2 = |1/W(z)|^2$.

Fig. 2 shows the perceptual filter in narrowband predictive coding, and G.729A is used as an example.

The filter $W(z)$ in Fig. 2 is optimized so that formants get relatively lower weights; that is, more noise is tolerated in frequency ranges where the signal has more energy and can mask more noise. This masking method assumes that the masker (signal) is sufficiently well reconstructed.

Most narrowband CELP speech coders, operating at 8 kHz, use a noise–shaping filter with the form $W(z) = A(z/\gamma_1)/A(z/\gamma_2)$ or a version based on the quantized LPC filter $\hat{A}(z)$. Either γ_1 or γ_2 can be set to 0.

In G.729, the filter $W(z) = A(z/\gamma_1)/A(z/\gamma_2)$ is based on the unquantized LPC filter $A(z)$ with adaptive values for γ_1 and γ_2. In G.729A, $W(z)$ is replaced by $\hat{A}(z)/A(z/\gamma)$, where $\gamma = 0.75$ and $\hat{A}(z)$ is a quantized LPC filter used to reduce complexity. In 3GPP AMR–WB, $\gamma_2 = 0$ and the weighting filter $A(z/\gamma_1)$, where $\gamma_1 = 0.92$ in the signal domain, corresponds to the effective weighting filter used for CELP coding in a pre–emphasized signal domain $A(z/\gamma_1) / (1-\alpha z^{-1})$, where



▲Figure 2. Perceptual weighting filter $W(z)$(γ =0.75) in G.729A. The coding noise spectrum approximates.

$\alpha = 0.68$. The cascade of pre–emphasis and LPC filters in AMR–WB is a special case of a technique introduced in [30]. The pre–emphasis is set at $1-\alpha z^{-1}$, where α is not adaptive, and the weighting filter is computed only on the second filter of the cascade.

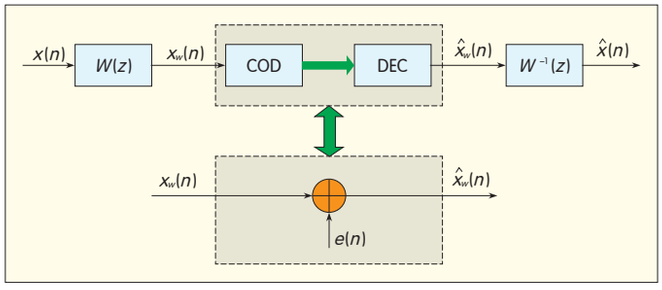
For details on predictive speech coding and AbS speech coding (including CELP) see [31]–[34]. Starting with the first CELP coding scheme in [22], the synthesis filters and perceptual weighting filters can be combined in one filter [35], which results in the filtering matrix H and a matrix form of the CELP error criterion. The CELP codebook search can be further improved by backward filtering [36], using binary algebraic codes [36]–[38], or by using sparse algebraic codes that give rise to an algebraic CELP (ACELP) model [39], [31, section 17.11]. The first real–time fixed–point implementation of an algebraic CELP coder, including noise shaping, was reported in [38].

2.3 Noise Shaping at the Decoder Side

Noise shaping for coded speech can be performed at the decoder side using a postfilter after the speech decoder [40]. A review of postfilering for CELP coders, including formant postfiltering, pitch postfilering, and gain control is given in [41]. Postfiltering is based on speech–signal model parameters (LPC and pitch) that are available in typical LPC–based speech decoders but not in PCM or ADPCM coders. The underlying principle of postfiltering is to reinforce signal components and redistribute coding noise. Such filtering is described in G.729 and G.729A. In 3GPP AMR–WB, some quality–enhancement techniques are used inside the speech decoder, that is, in the LPC residual domain prior to LPC synthesis.

Noise reduction can also be used to improve quality at the decoder side. When a statistical model is available for coding

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P39



▲Figure 3. Noise shaping by pre/postprocessing.

noise, good results can be obtained [42], [43]. Strictly speaking, this technique is not a form of noise shaping but is a reduction of the noise level by adaptive spectral attenuation, and extra delay is usually required for frequency–domain analysis and processing. If the noise is shaped at the encoder side, the resulting noise shapes need to be taken into account in decoder–side postprocessing.

2.4 Joint Encoder and Decoder Noise Shaping

Fig. 3 shows the principle of noise shaping by prefiltering and postfiltering. If the codec is modeled using an additive noise model, the decoded signal is

$$\hat{X}(z) = X(z) + \frac{E(z)}{W(z)}. \quad (8)$$

Noise shaping is decoupled from actual quantization or coding. The (inner) encoder–decoder can be optimized to minimize mean–square error, and the resulting coding error $E(z)$ has a nearly flat spectrum. The overall coding noise is then shaped according to the frequency response of $W^{-1}(z)$. In practice, this basic linear model is only an approximation because the filter coefficients are adapted per frame, and the actual process is non–linear.

Joint encoder/decoder processing is used in the transform coded excitation (TCX) model [44] and transform predictive coding (TPC) model [45]. The perceptual filter $W(z)$ is typically defined for LPC–based noise shaping:

$$W(z) = \frac{\hat{A}(z/\gamma_1)}{\hat{A}(z/\gamma_2)}. \quad (9)$$

The decoder needs to revert the preprocessing; therefore, the coefficients of $\hat{A}(z)$ are quantized of the linear predictive filter. This contrasts with the perceptual filters discussed in section 3, which can use unquantized coefficients.

A similar idea was explored in [46] for audio coding. In [47], an LPC filter derived from a model–based masking curve (a function of LPC and pitch) was investigated. In [48], the LPC–based perceptual weighting filter in TCX coding is modified in the frequency domain by adaptive low–frequency emphasis. This is done to improve the quality of some high–pitched music signals. In [49], combined companding and expanding at 48 kbit/s in the spectral domain results in the same quality as G.722 at 64 kbit/s [20], [50]. However, this technique implies that backward compatibility with existing coders is lost with the introduction of the new, improved coder, and extra delay and complexity are required for frequency–domain processing.

3 Noise Feedback Coding

In this section, we define NFC by detailing different filter structures. We also address the related problems of noise–shaping filter estimation and loop stability. The following notation is used:

$x(n)$: input signal to (outer) encoder
$x'(n)$: input signal to (inner) encoder (modified signal including noise feedback)
$\tilde{x}(n)$: signal reconstructed by (inner) local decoder
$e(n) = \tilde{x}(n) - x(n)$: overall coding noise (or reconstruction noise)
$q(n) = x(n) - x'(n)$: inner coding noise (or quantization noise)

3.1 Moving–Average Structure

Fig. 4 shows noise fed back using a moving–average (MA) filter structure. The quantization noise $q(n)$ is the error introduced by the inner encoder–local decoder (COD–Local DEC) and is given by

$$q(n) = \tilde{x}(n) - x'(n). \quad (10)$$

The input signal $x(n)$ is modified by adding the quantization noise filtered by $H_A(z) - 1$. The signal $X'(z)$ is given by

$$X'(z) = X(z) + (H_A(z) - 1)Q(z). \quad (11)$$

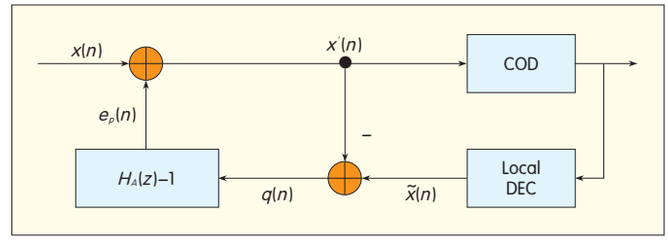
Replacing $X'(z)$ with $\tilde{X}(z)$ and $Q(z)$ gives

$$\tilde{X}(z) - X(z) = H_A(z)Q(z) \quad (12)$$

Therefore, the overall reconstruction error $\tilde{X}(z) - X(z)$ is the quantization noise $q(n)$ shaped by the filter $H_A(z)$. If the bitrate is sufficient, the quantization noise can be considered white noise, and the spectrum of the reconstruction noise is close to $|H_A(z)^2|$.

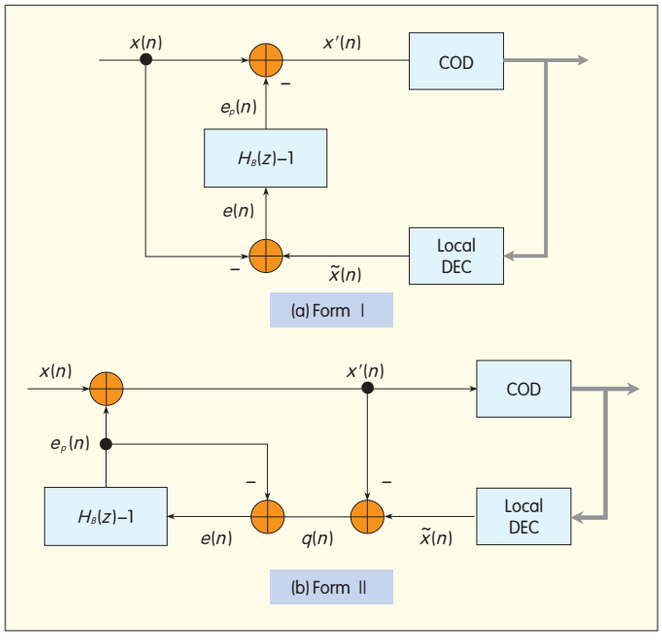
The value $e_p(n)$ corresponds to a prediction because the filter $H_A(z) - 1$ has a zero coefficient in z^0, that is, for sample n. This property indicates that the loop filter is an MA predictor for the next quantization error, and this prediction is based on past samples of the quantization error signal $q(n)$.

As in an APC coder, the inner coding noise $\tilde{x}(z) - x'(n)$ is the quantization noise $q(n)$. Fig. 4 can be considered a generalization of [25, Fig. 9] and [26, Fig. 3 where $A = 0$] because the noise feedback loop has been moved outside the coder. This is true when COD is a linear predictive coder without noise shaping, as in the original APC [21], PCM [2],



▲Figure 4. Noise feedback with moving–average structure.

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P40



▲Figure 5. Noise feedback with auto–regressive structure.

and ADPCM [20] schemes. The structure in Fig. 4 allows the introduction of noise shaping in these legacy codecs as external pre–processing, and the COD–Local DEC loop is not affected.

3.2 Auto–Regressive Structure

Fig. 5 shows two equivalent auto–regressive (AR) filter structures. In form I, the decoded signal $\tilde{X}(z) = X'(z) + Q(z)$ can be expanded:

$$\tilde{X}(z) = X(z) - (H_B(z) - 1)(\tilde{X}(z) - X(z)) + Q(z) \quad (13)$$

which yields

$$\tilde{X}(z) - X(z) = \frac{1}{H_B(z)} Q(z). \quad (14)$$

Form II gives the same result. The main advantage of form I is that it saves one subtraction per sample compared with form II.

Fig. 5(a) is a special case of [26, Fig.3], where $A = 0$ and the feedback loop has again been moved outside the COD–Local DEC loop. Form II is the same as that in Fig. 4 when

$$H_A(z) = \frac{H_B(z) - 1}{H_B(z)} + 1 \quad (15)$$

Here, the loop filter is an AR predictor for the next quantization error, and the prediction is based on past samples of the coding noise $e(n)$.

3.3 Auto–Regressive Moving–Average Structure

Combining the structures in Figs. 4 and 5(a) gives a general auto–regressive moving–average (ARMA) structure (Fig. 6). The coding noise is given by

$$\tilde{X}(z) - X(z) = \frac{H_A(z)}{H_B(z)} Q(z) \quad (16)$$

The noise shaping in Fig. 6 can also be obtained using the structure in Figs. 4 and 5(a) by introducing an ARMA predictor in the feedback loop and computing the noise prediction from the past quantization noise and reconstructed noise, as in [51, Figs. 4 and 5]. In [51, Fig. 5] ARMA long–term noise shaping was included in the feedback loop to improve the performance of legacy PCM and ADPCM coders for voice signals.

3.4 Loop Filter Design: HA and HB

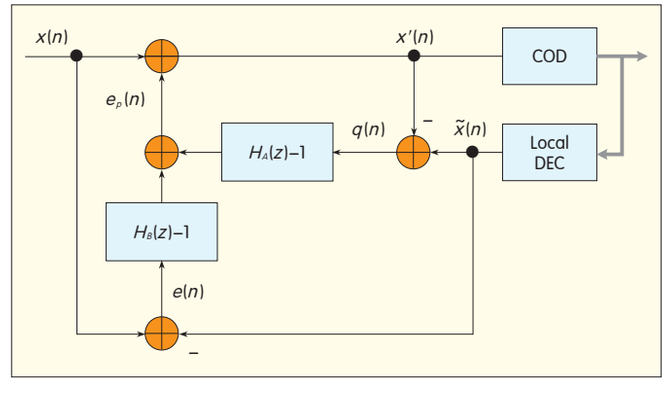
The simplest design for the loop filter includes a fixed filter, such as $H_A(z) = 1 + z^{-1}$. In this case, the effective quantization noise is colored in a predefined way. For speech or audio signals, this approach is generally not optimal, and it is better to adapt the loop filter to the short–term spectral properties of the input signal [5]. A detailed analysis using high–rate distortion theory (i.e. small errors) shows that, with certain assumptions, the optimal loop filter whitens the input signal [5]. Hence, a linear predictive (LP) analysis can be used for loop–filter estimation.

In [25] and [26], LPC techniques were used to obtain the weighting filters from the linear predictive coefficient. In [52], the weighting filter is an LPC filter computed on the adaptively pre–emphasized speech signal.

Because noise feedback is implemented at the encoder side, the loop filter estimation can be based on the original signal $x(n)$. In cases such as embedded PCM, where a core coder and enhancement stages depend on the loop filter at both the encoder and decoder sides, it can be beneficial to estimate the filter on the decoded core signal $\tilde{x}(n)$. This is the case of G.711.1, where the loop filter is estimated through an LP analysis on the decoded core signal $\hat{S}_{LO}(n)$ at the encoder and decoder. In G.722, the LP analysis is done on the input signal $x(n)$.

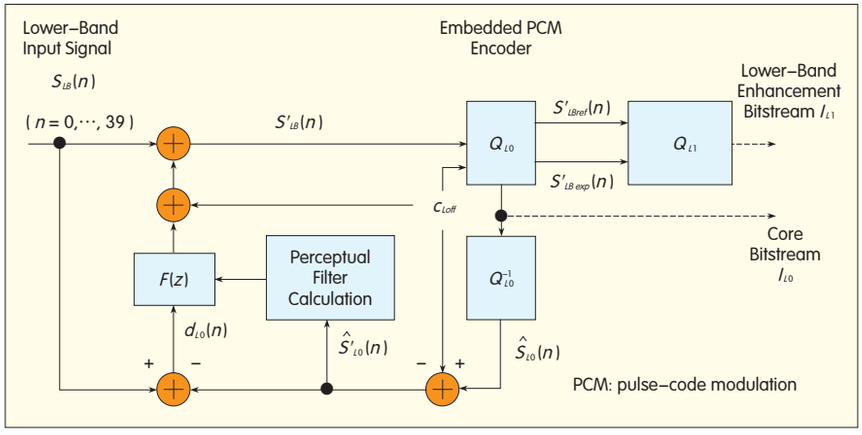
3.5 Loop Stability

As in any system with feedback, the noise feedback loop may become unstable in certain conditions. The quantization process is non–linear, and this makes it difficult to model the



▲Figure 6. Noise feedback with ARMA structure.

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P41



▲Figure 7. A G.711.1 encoder [58].

loop behavior. For the loop transfer function to be stable, the NFC transfer function must have its poles inside the unit circle. A stability analysis of the APC coder with a parametric model of the APC system can be found in [53].

In [54], stability was analyzed for the embedded ADPCM with noise feedback. Perceptual input/output SNR of the coder is given by

$$SNR_P = G_P\left[\frac{SNR_Q - 1}{E_D} + 1\right] \quad (17)$$

where G_P is the ADPCM prediction gain, SNR_Q is the SNR of the quantizer (about 24 dB for a 5–bit quantizer), and E_D is the impulse response energy of the masking filter H_s (in the general case). When SNR_P is low, the ADPCM coder is close to being unstable.

Because of this problem, the order of the loop filter is usually small. In recent G.711.1 and G.722B standardization work, an AR structure is used with a loop filter of order 4, and some heuristic methods have been developed that limit loop instability by detecting specific signals, for example, chirps and combinations of sinusoids and frequency hops. These signals create instability or degrade performance compared with having no feedback. Stability issues are mitigated by detecting such signals, and the noise–shaping filter resonances are attenuated or the loop is turned off for a period of time [9], [10], [54].

4 Improving G. 711 by Using Embedded PCM Coding with Noise Shaping

G.711 [2] is the most widely deployed speech codec in fixed networks and voice–over–IP (VoIP) networks. In [2], the input linear PCM is coded at 8 bits/sample using either A–law or a μ–law. In 2007, ITU–T SG16 started a work item called G.711–WB (wideband) under the initiative of NTT. The objective of this work was to develop a 64–80–96 kbit/s embedded extension of G.711 that was low–delay, low–complexity, and capable of 50–7000 Hz wideband [55]. The principles of embedded PCM, ADPCM, CELP coders, and hierarchical coders such as G.729.1 [56] can be found in [57].

4.1 Principle of G. 711.1

The G.711.1 encoder receives 8 kHz or 16 kHz input signals. The input is divided into 5 ms frames. The encoding bit rate can be set at 64 kbit/s, 80 kbit/s, or 96 kbit/s. The input signal is decomposed into two sub–bands using a quadrature mirror filter bank (QMF). The 0–4000 Hz lower band is coded by an embedded PCM encoder that is interoperable with G.711 (Fig. 7). The 4000–8000 Hz higher band is coded in the modified discrete cosine transform (MDCT) domain.

The bitstream structure of G.711.1 is defined in Table 1. More details on G.711.1 can be found in [9], [58].

4.2 Embedded PCM Coding Without Noise Shaping

The PCM coding in [2] is similar to the scientific notation; a given number of bits (resolution) in linear PCM is retained to form the mantissa, and an exponent value gives the correct scaling. In G.711, which has 8 bits per sample, 5–bit precision is used for the signed mantissa, and this results in approximately 38 dB SNR (except for the smallest values in the first segment). The most significant bit (MSB) is always 1 in natural binary decomposition; therefore, it does not need to be transmitted. Only the next 4 bits (after MSB) are sent to the decoder; one bit is reserved for the sign, and the remaining 3 bits are used to encode 8 possible exponent values (segment indices).

In G.711.1, a 16 kbit/s, 2 bit/sample enhancement layer was introduced to enhance the 0–4000 Hz lower band. In [55], two extra mantissa bits of the binary representation of input samples are extracted and transmitted to the extension layer, which gives an overall resolution of 6 bits in the first segment and 7 bits in the other segments. In the final tuning phase of G.711.1 standardization, a dynamic bit allocation was taken to allocate 1, 2 or 3 bits per sample to the frame bit budget [9].

4.3 Embedded PCM Coding with Noise Shaping

In the 0–4000 Hz lower band of G.711.1, noise shaping was introduced to PCM coding to improve the quality of the interoperable core bitrate mode at 64 kbit/s and to improve the quality of the enhanced high–bitrate mode at 80 kbit/s. In

▼Table 1. Bitstream structure of G.711.1

Mode	Sampling Rate (kHz)	Core Layer I_{L0}	Lower–Band Enhancement Layer I_{L1}	Higher–Band Enhancement Layer I_{L2}	Overall Bit Rate (kbit/s)
		64 kbit/s	16 kbit/s	16 kbit/s	
R1	8	x	–	–	64
R2a	8	x	x	–	80
R2b	16	x	–	x	80
R3	16	x	x	x	96

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P42

this way, the 64 kbit/s core layer of G.711.1 can be decoded by a legacy G.711 decoder, and the quality of G.711 encoding and decoding can be significantly improved, especially for low–level clean speech.

In the G.711.1 encoder, noise shaping is performed on the core–layer signal using the form I AR structure (Figs. 5(a) and 7). The decoded core coder output is given by

$$\hat{S}_{L0}(z) = S_{LB}(z) + \frac{1}{1+F(z)} Q(z) \quad (18)$$

The enhancement layer is defined in section 4.1, and noise shaping had to be performed on the enhancement signal at the decoder before being added to the core decoded signal. To prevent a mismatch between the decoder and the encoder, the perceptual filter $F(z)$ in Fig. 7 is calculated using the past decoded core signal available at both encoder and decoder side [51], [52].

In the encoder, the quantization noise of the core layer filtered by $F(z)$ is fed back to be added to the input signal encoded using embedded PCM coding (Figs. 5a and 7). Therefore, if the decoder decodes only the core layer, the quantization noise is already shaped when it is decoded by a G.711 decoder. If the enhancement layer is also decoded, the enhancement–layer contribution, that is, the difference between the core bitrate output and the higher bitrate output, has to be shaped at the decoder. It is thus shaped by the noise shaping filter $1/(1+F(z))$ prior to being added to the core layer output. Noise shaping is performed at the encoder side for the core layer and at the decoder side for the enhancement layer [51], [52].

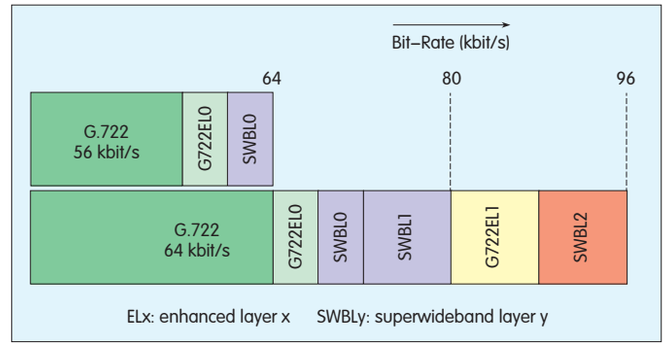
In the G.711.1 core layer, the ordinary log–PCM encoding is replaced by a dead–zone quantizer encoding scheme [9, clause 7.3.3]. Another way to enhance the quality of the PCM coding and reduce the perceptual impact of PCM quantization noise is to postfilter quantization noise at the decoder [42]. This technique is detailed in [9, Appendix I].

4.4 G.711 App. III: Noise Shaping as Preprocessing to G. 711 and Other Tools

Noise shaping was incorporated into the 64 kbit/s core coder in G.711.1; however, it can also be performed externally as preprocessing to G.711. This preprocessing approach, together with other tools such as the postfiltering of G.711.1 App. I, form G.711 Appendix III [19].

5 Improving G.722 by Using Embedded ADPCM Coding with Noise Shaping

G.722 was the first normalized 50–7000 Hz wideband (WB) speech codec [20]. This coder is based on a sub–band embedded ADPCM coding scheme [50]. The input signal of G.722, sampled at 16 kHz, is decomposed by a QMF into two sub–bands: 0–4000 Hz and 4000–8000 Hz. Each sub–band is coded separately using ADPCM coding. The total bit rate is 64, 56 or 48 kbit/s, depending on the number of bits allocated to the lower band. The block diagram of the G.722 encoder and decoder can be found in [50, Figs. 9 and 12]. G.722 is the wideband codec specified for DECT new Generation (DECT–NG) terminals, which are also called Cordless Advanced Technology—Internet and quality terminals. This codec provides better audio quality than the legacy narrowband DECT codec (G.726). To further improve call quality, 50–14,000 Hz superwideband (SWB) coding is the next step. In 2008, the ITU–T launched the G.711.1/G.722 SWB work item under the initiative of France Telecom and NTT. This activity resulted in G.722 Annex B (G.722B) [10], [59]. We focus here on improvements made to G.722 as part of G.722B development. The principle of embedded coding is explained in [57].



▲Figure 8. Bitstream structure of G.722B.

5.1 Principle of G. 722 Annex B

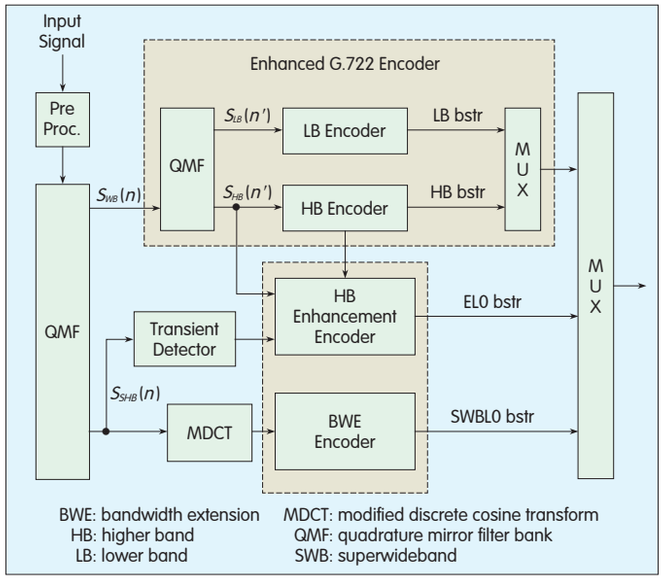
G.722–SWB standardization is aimed at developing an embedded scalable SWB extension of G.722 operating at 64, 80, and 96 kbit/s. The 64 kbit/s bit rate of G.722B was specifically designed to fit into existing 64 kbit/s CAT–iq transport channels which comprise wideband G.722 at 56 kbit/s and 8 kbit/s for an SWB enhancement layer. Fig. 8 shows the embedded bitstream structure of G.722B. Here we only discuss the part of G.722B that corresponds to the G.722 core bitstream layer at 56 kbit/s and 64 kbit/s.

The main objective of G.722B is to extend G.722 to SWB. As with G.711.1, it was necessary to improve G.722 because artifacts in the lower band are reinforced when higher–band extensions are added. At 64 kbit/s, G.722B operates with a 56 kbit/s G.722 core coder that has an 8 kbit/s enhancement layer split into a G.722 high–band enhancement (EL0) and an SWB enhancement (SWBL0). At 80 or 96 kbit/s, G.722B operates with a 64 kbit/s G.722 core coder that has two high–band enhancement layers: G722EL0 and G722EL1, and three SWB enhancement layers: SWBL0, SWBL1, and SWBL2.

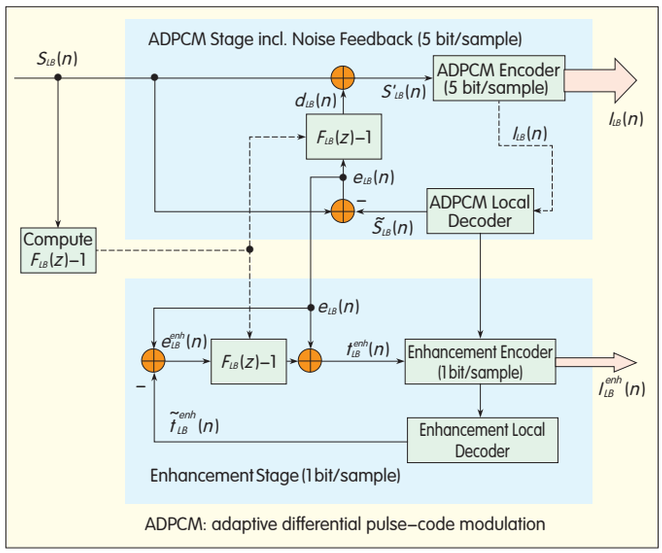
As shown in Fig. 9, the G.722B coder includes an enhanced G.722 coder for the wideband (WB) part, which enhances the G.722 4000–8000 Hz higher band, and a bandwidth extension in the MDCT domain.

In the original G.722 coder [20], [50], the lower sub–band is coded by an embedded ADPCM coder at 6, 5 or 4 bit/sample, and the higher sub–band is coded by an ADPCM coder at 2 bit/sample. G.722 relies on plain ADPCM coding with no noise shaping. In the following, we describe the G.722 lower–band (LB) coder and the G.722 higher–band (HB)

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P43



▲Figure 9. Block diagram of G.722B.



▲Figure 10. Lower band coding in G.722B.

coder, including noise shaping.

5.2 Improved 0–4000 Hz Lower–Band Embedded ADPCM Coding Including Noise Feedback

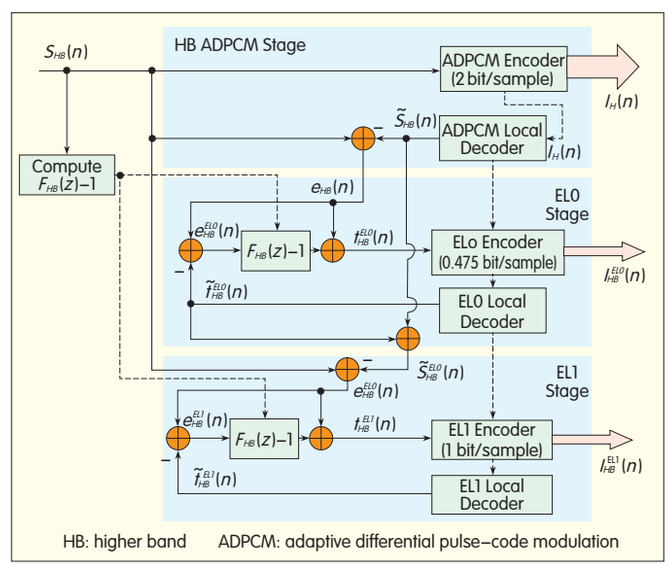
Fig. 10 shows a modified LB ADPCM coder, including noise shaping. In the case of G.722B at 64 kbit/s, the G.722 LB encoder operates at 5 bit/sample, and in the case of G.722B at 80 or 96 kbit/s, the G.722 LB encoder operates at 6 bit/sample. To shape the coding noise at 5 and 6 bit/sample in a coherent and optimal way, embedded ADPCM coding is modified so that it operates in two optimal stages. A noise feedback loop is included in the first stage, and AbS is done in the second stage. Therefore, a noise feedback loop is used with the ADPCM coder operating at a reduced bitrate of 40 kbit/s (5 bit/sample). In Fig. 10, noise shaping is performed on the core layer signal using form I of the AR structure shown in Fig. 5(a). The COD is an ADPCM coder with 5 bit/sample, and the Local DEC is an ADPCM local decoder with 5 bit/sample. An enhancement encoder using AbS and operating at 1 bit/sample brings the bit rate to 48 kbit/s (6 bits/sample) [10], [60].

Thus, the modified G.722 LB encoder relies on embedded scalar quantization, which is done by AbS in a perceptually weighted domain. Both stages use the same noise shaping filter $F_{LB}(z)-1$ derived from a linear predictive coding (LPC) filter $F_{LB}(z)$ with an order of 4.

5.3 Improved 4000–8000 Hz Higher–Band ADPCM Coding (G.722 HB and EL0/EL1)

Fig. 11 shows the HB encoder. The core G.722 HB encoder corresponds to the G.722 HB ADPCM encoder operating at 16 kbit/s (2 bit/sample). The ADPCM signal–to–noise ratio in the higher band is low because the allocated bitrate is 2 bit/sample as opposed to 5 or 6 bit/sample for G.722 LB ADPCM coding. This difference in bitrate means that contrary to the G.722 lower band, no noise feedback is used in higher–band ADPCM coding. For noise feedback to be efficient, a bit rate higher than 2 bit/sample is desirable. If additional bit rate is available to improve the quality of the G.722, these extra bits are allocated to the higher band. The quantization resolution can be improved by the enhanced G.722 HB encoder, which relies on embedded scalar quantizers of 0.475 bit/sample (EL0) and 1 bit/sample (EL1).

The G.722 HB ADPCM encoder can be extended by a scalable or embedded bitstream. This extension is done in two embedded stages using AbS, which is similar to the approach used for low–band enhancement. The first extension stage, G722EL0, is performed at 3.8 kbit/s, and 19 of the 40 samples are used only in non–transient frames to enhance quality with 1 bit/sample. This forms the G722EL0



▲Figure 11. Higher band coding in G.722B.

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P44

layer, which is used in all superwideband layers (Fig. 8). The G722EL0 layer is disabled in the case of transient signal segments, where the spare 19 bits are allocated to SWB extension. The second extension stage, G722EL1, is performed at 8 kbit/s (1 bit/sample) to further refine the quantization of the G.722 higher band, and this forms the G722EL1 layer that is only used in G.722B at 96 kbit/s.

5.4 Analyzing the Core Coder Enhancement Layer with Analysis–by–Synthesis in Embedded ADPCM Coding

Here, we explain how the LB embedded ADPCM coder in G.722 Annex B [10] is modified to allow backward–compatible noise shaping. As described in Section V.B of the G.722 standard, the G.722 LD encoder is split into a core coder with noise shaping at 56 kbit/s (5 bit/sample) and an AbS enhancement to reach 6 bit/sample.

The legacy ADPCM local decoder at 5 bit/sample reconstructs the output signal by adding the scaled ADPCM quantizer output at 5 bit/sample to the prediction obtained at 4 bit/sample. To obtain the 6th bit in the legacy coder (without noise shaping), the embeddedness of the ADPCM quantizer can be used to find the two scaled and adequate 6–bit quantizer levels. The predicted signal can be added to the quantizer output to give two possible synthesized signals, and then a search can be done to determine which of the two signals is the closest to the input signal $s_{LB}(n)$. AbS is done to find the extension layers of the core coder bit by bit, and it does not involve any noise shaping.

To introduce a weighted–error criterion [60], the extension–stage error criterion is

$$E_{LB}=\frac{1}{2\pi j}\oint_{c}\left|\left\{[S_{LB}(z)-\tilde{S}_{LB}(z)]-\tilde{\xi}^{i}_{LB}(z)\right\}W(z)\right|^{2}\frac{dz}{z},\ i=0,1 \qquad (19)$$

where $\tilde{S}_{LB}(z)$ is the core coder reconstructed signal, $\tilde{\xi}^{i}_{LB}(z)$ is the z–transform of the enhancement output signal, and $W(z)$ is the weighting filter equal to $F_{LB}(z)=1+\sum_{i=1}^{4}a_{LBi}z^{-i}$.

The signal $\tilde{\xi}^{i}_{LB}(z)$ is equal to the past outputs of the enhancement layer for $n'<n$, that is, $\tilde{t}^{enh}_{LB}(n)$ in Fig. 10, and is equal to the ith enhancement candidate $\xi^{i}_{LB}(n')$ for $n'=n$ [10, B.6–32]. The noise spectrum of $S_{LB}(z)-\tilde{S}_{LB}(z)-\tilde{\xi}^{i_{opt}}_{LB}(z)$ at theoutput of the enhancement stage has the shape $\left|\frac{1}{F_{LB}(z)}\right|^{2}$. The signal $S_{LB}(z)-\tilde{S}_{LB}(z)$ in the bracket in the error criterion is equal to $e_{LB}(n)$ in Fig. 10.

Although the coding structure in Figs. 10 and 11 use AbS, this technique is quite different from CELP coding because the sample–by–sample gain is provided by the core layer. Also, for each sample n, the enhancement codebook is obtained from the ADPCM index of the core layer and the possible preceding enhancement layers. It is obtained using the embeddedness of the ADPCM quantizers [10, B.6–32].

The same method can be used for the high–band ADPCM encoding. Because the method is based on AbS with noise shaping, the results can be extended to minimize error of (Eq. 19) on a block basis and to increase the efficiency of the noise shaping. If care is taken in embedded quantization, the reconstructed, enhanced codebook with two entries $\xi^{i}_{LB}(n)$ $i=0, 1$ (19) can be extended to a binary algebraic tree codebook, which allows fast algorithms to be derived for minimizing the error criterion [61]. Computation load is only slightly increased. Therefore, it is possible to further improve the quality of the G722EL0 and G722EL1 layers.

5.5 Experimental Results

In the official ITU–T tests for G.722B, only SWB cases were tested. In [62], some additional wideband tests were done to assess the backward compatibility of G.722B with G.722.

6 New Codec Structures

The principles of noise feedback were recently revisited in an effort to develop new codec structures, for example, BV16, BV32 [11], [12] and SILK [13].

6.1 BV16 and BV32 Coders

The BV16 and BV32 coders [11], [12] are based on a synthesis model that includes a long–term (LTP) synthesis filter and a short–term (LPC) synthesis filter. The first predictive speech codecs that used noise shaping [25], [26] relied on a scalar quantization and a short–term noise–feedback filter to shape the spectral envelope of the coding noise. In contrast, the two–stage noise feedback coding (TSNFC) in [11] and [12] relies on two NFC stages in a nested loop. In the first NFC stage, short–term prediction and short–term noise spectral shaping (spectral envelope shaping) is performed. In the second (nested) NFC stage, long–term prediction and long–term noise spectral shaping (harmonic shaping) is performed.

In Fig. 12, the long–term noise shaping predictor $N_l(z)-1$ is located around the quantizer, as in [25, Fig. 7]. This implies MA long–term noise shaping by $N_l(z)$. The short–term noise–shaping predictor $F_s(z)$ acts on the noise of the long–term quantizer loop so that the noise is shaped by $1-F_s(z)$. Thus, the short term filter $1-P_s(z)$ is a pre–emphasis, and the coding noise is shaped by $\frac{1}{1-P_s(z)}$ [21]. To more precisely describe the z–transform of the overall system, the output speech is given in terms of the input:

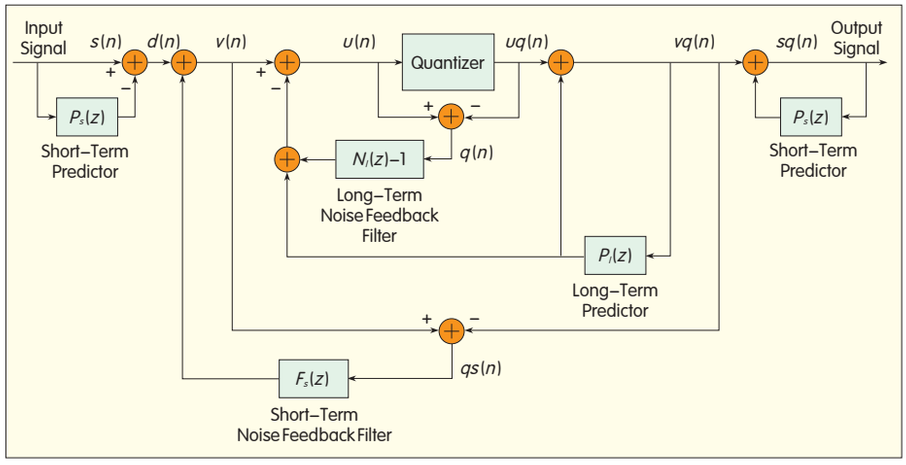
$$S_Q(z)=S(z)-N_l(z)\frac{1-F_s(z)}{1-P_s(z)} \qquad (19)$$

MA long–term noise shaping and ARMA short–term noise shaping depend on the LPC predictor $P_s(z)$, so the amount of noise shaping is obtained from the damping factor γ used in $F_s(z)$. An alternative ARMA long–term noise shaping and more flexible ARMA short–term noise shaping is described in [51, Fig. 5]. Further details on the BV16 or BV32 coders can be found in [11], [12] and [31].

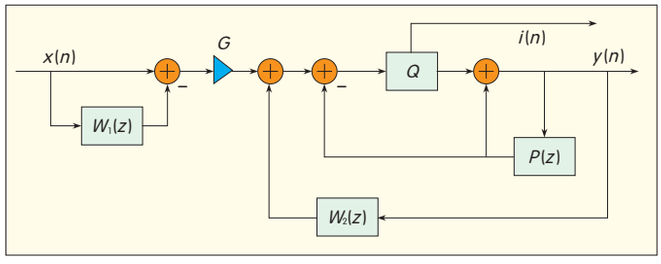
6.2 SILK Coder

SILK is a speech codec designed for VoIP [13]. The synthesis model for a SILK coder is based on a long–term synthesis filter and a short–term synthesis filter. The excitation

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P45



▲Figure 12. Noise shaping in BV32 coder.



▲Figure 13. SILK coder.

signal is obtained by range–coded quantization, which is similar to arithmetic–coded quantization. Fig. 13 shows the noise shaping of the SILK codec.

The noise–shaping predictor $W_2(z)$ is in a feedback path around an inner coder. This inner coder has a generalized predictor filter $P(z)$ that includes LPC and LTP. Therefore, the quantization noise $Q(z)$ is filtered by $\frac{1}{1-W_2(z)}$. The filter $1-W_1(z)$ acts as a pre–filter whose inverse is not present in the local decoder, as in the classical pre–filter in [21]. As a consequence, the spectral content of the input speech is modified by the codec. This can be confirmed by writing the input/output z–transform of Fig. 13 as

$$Y(z) = G\frac{1-W_1(z)}{1-W_2(z)}X(z) + \frac{1}{1-W_2(z)}Q(z) \qquad (20)$$

If G is set at 1 and $W_1(z) = W_2(z)$, the coder of Fig. 5(a) is obtained. Noise shaping in SILK involves modifying the input signal. This is similar to postfiltering the input as in [41], where the filter has a short–term and long–term section and gain is controlled by G, and shaping the noise, as in NFC.

The SILK noise shaping structure can be obtained by transforming the structure in Fig. 5(a). In the structure in Fig. 5(a), noise is shaped in the vector case by minimizing the weighted error criterion between the input and output signals. The structure in Fig. 13 allows the levels of the decoded speech formants to be matched with the levels of the original speech formants. The levels of the spectral valleys are decreased relative to the levels of the spectral peaks, including speech formants and harmonics. To obtain both long–term and short–term noise shaping, $W_1(z)$ and $W_2(z)$ should be of the form

$$W_{i=1,2}(z) = [Sh_{i=1,2}(z) + LT_{i=1,2}(z)[1-Sh_{i=1,2}(z)]] \qquad (21)$$

where $SH_{i=1,2}(z)$ is the short–term predictor and $LT_{i=1,2}(z)$ is the long–term predictor. Then $W_{i=1,2}(z)$ appears to be the product of the short–term and long–term inverse filters minus one.

7 Conclusion

Two applications of NFC were described: the improvement of existing coders, for example, G.711 and G.722 by making them backward–compatible, and the development of new codec structures that are alternatives to the ubiquitous CELP coding model.

The improvement of legacy codecs is an important research topic, and significant gains are possible with refurbished codecs. Versions of G.711 and G.722 using both encoder–side and decoder–side techniques have been standardized by the ITU–T [9], [10], [19].

As part of enhanced voice standardization (EVS), 3GPP SA4 is also considering improving the 3GPP AMR–WB standard by making it bit–stream interoperable.

Acknowledgment

The authors would like to thank colleagues involved in the ITU–T G.711.1 and G.711.1/G.722–SWB work as well as one reviewer for helpful comments to improve this article.

References

[1] R. M. Dolby, "An audio noise Reduction System," *JAES*, vol. 15, no. 4, pp. 383–388, Oct. 1967.
[2] *Pulse code modulation (PCM) of voice frequencies*, ITU–T G.711, Dec. 1972.
[3] C. C. Cutler, "Transmission systems employing quantization," U.S. Patent 2 927 962, 1960.
[4] H. A. Spang and P. M. Schultheiss, "Reduction of quantizing noise by use of feedback," *IRE Trans. Communications Systems*, vol. 10, issue 4, pp. 373–380, Dec. 1982.
[5] N. S. Jayant and P. Noll, "Noise feedback coding," Chapter 7 in *Digital Coding of Waveforms*, New Jersey: Prentice Hall, Mar. 1984.
[6] H. S. Lee and C. K. Un, "On the performance of speech waveform coder with noise spectral shaping", in *IEEE Trans. Commun.*, vol. 33, no.7, pp. 742–746, July 1985
[7] H. D. Kim and C.K. Un, "Embedded ADPCM with noise shaping for packet voice transmission," in *IEEE Electronic Letters*, vol. 3, no. 5, Feb. 1987
[8] H. Inose, Y. Yasuda, and J. Marakami, "A telemetering system by code modulation: modulation," *IRE Transations on Space Electronics Telemetry*, vol. SET–8, pp. 204–209, Sept. 1962.
[9] *Wideband embedded extension for G.711 pulse code modulation*, ITU–T G.711.1, March 2008.
[10] N*ew Annex B with superwideband embedded extension,* ITU–T G.722 Amd. 1, Nov. 2010.
[11] J. H. Chen, "Novel codec structures for noise feedback coding of speech," in Proc. *Int. Conf. Acoustic, Speech and Signal Processing (ICASSP)*, Toulouse, 2006, vol. 1, pp. 1.
[12] J. H. Chen and J. Thyssen, "The broadvoice speech coding algorithm," in Proc. *Int. Conf. Acoustic, Speech and Signal Processing (ICASSP)*, Honolulu, HI, 2007, vol. 4, pp. 537–540.
[13] K. Vos, S. Jensen, and K. Soerensen. (2010, Sept.). *SILK speech codec* [Online].

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P46

Available: http://tools.ietf.org/html/draft-vos-silk-02
[14] E. Zwicker and H. Fastl, *Psychoacoustics: Facts and Models*, 2nd edition, Springer, 1999.
[15] W. H. Hartmann, *Signals, Sounds, Sensation*, New York: Springer, 1997.
[16] K. Brandenburg, "Perceptual coding of high quality digital audio," Chapter 2 in *Applications of Digital Signal Processing to Audio and Acoustics*, Dordrecht: Kluwer Academic Publishers, 2002.
[17] M. Bosi and R. E. Goldberg, *Introduction to Digital Audio Coding and Standards*, New York: Springer, 2002.
[18] J. Herre and M. Lutzky, "Perceptual audio coding of speech signals", Chapter 18 in *Springer Handbook of Speech Processing*, (Benesty, Sondhi, Huang Editors), New York: Springer, 2008.
[19] *Audio quality enhancement toolbox*, ITU-T G.711 App. III, Nov. 2009.
[20] *7 kHz audio-coding within 64 kbit/s*, ITU-T G.722, Nov. 1988.
[21] B. S. Atal and M. R. Schroeder, "Adaptive predictive coding of speech signals," *Bell System Tech. J.*, vol. 49, no. 8, pp.1973-1986, Oct. 1970.
[22] M. R. Schroeder and B. S. Atal, "Coded-excited linear prediction (CELP): high quality speech at low bit rates," in Proc. *Int. Conf. Acoustic, Speech and Signal Processing (ICASSP)*, Tampa, FL, vol. 10, pp. 937-940, 1985.
[23] M. Berouti and J. Makhoul, "High quality adaptive predictive coding of speech," in Proc. *Int. Conf. Acoustic, Speech and Signal Processing (ICASSP)*, Tulsa, AZ, pp. 303-306, Apr. 1978.
[24] B. S. Atal and M. R. Schroeder, "Predictive coding of speech signals and subjective error criteria," in Proc. *Int. Conf. Acoustic, Speech and Signal Processing (ICASSP)*, Tulsa, AZ, pp. 573-576, Apr. 1978.
[25] J. Makhoul and M. Berouti, "Adaptive noise spectral shaping and entropy coding in predictive coding of speech," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol.27, no. 1, pp. 63-73, Feb. 1979.
[26] B. S. Atal and M. R Schroeder, "Predictive coding of speech signals and subjective error criteria," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. 27, no. 3, pp. 573-576, June 1979.
[27] M. R. Schroeder, B. S. Atal, and J. L. Hall, "Optimizing digital speech coders by exploiting masking properties of the human ear," *J. Acoust. Soc. Amer.*, vol. 66, issue 6, pp. 1647-1652, 1979.
[28] B. S. Atal and J. R. Remde, "A new model of the LPC excitation for producing natural-sounding speech at low bit rates," in Proc. *Int. Conf. Acoustic, Speech and Signal Processing (ICASSP)*, Paris, pp. 614-617, 1982.
[29] A. Le Guyader, R. Di Francesco, and C. Lamblin, "Derivation of efficient CELP coding algorithms using the Z-transform approach," *Proc. ICASSP*, Toronto, vol. 1, pp. 209-212, 1991.
[30] C. Quinquis, A. Le Guyader, "Method of analysing by linear prediction an audio frequency signal, and its application to a method of coding and decoding an audio frequency signal," Patent EP0782128, Jul. 2, 1997.
[31] J. H. Chen and J. Thyssen, "Analysis-by-synthesis speech coding", Chapter 17 in *Springer Handbook of Speech Processing* (Eds. Benesty, Sondhi, Huang), Springer, 2008.
[32] A. M. Kondoz, *Digital Speech Coding for Low Bit Rate Communication Systems*, New York: Wiley, 2004.
[33] P. Vary and R. Martin, *Digital Speech Transmission – Enhancement, Coding & Error Concealment*, New York: Wiley, 2006.
[34] N. Moreau, *Tools for Signal Compression: Applications to Speech and Audio Coding*, New York: Wiley, 2011.
[35] I. M. Trancoso and B. S. Atal, "Efficient procedures for finding the optimum innovation in stochastic coders," in Proc. *Int. Conf. Acoustic, Speech and Signal Processing (ICASSP)*, Tokyo, vol. 11, pp. 2375-2378, 1986.
[36] J. P. Adoul, P. Mabilleau, M. Delprat, and S. Morisette, "Fast CELP coding based on algebraic codes," in Proc. *Int. Conf. Acoustic, Speech and Signal Processing (ICASSP)*, Dallas, TX, pp. 1957-1960, 1987.
[37] D. Massaloux, A. Le Guyader, and J. F. Zurcher, "A new fast algorithm used in a vector adaptive predictive coder," CNETreport, ref:296/LAA/TSS/CMC, 1986.
[38] A. Le Guyader, D. Massaloux, and J. F. Zurcher, "A robust and fast CELP coder at 16 kbit/s," *Speech Communication*, vol.7, no. 2, pp. 217-226, Jul. 1988.
[39] R. Salami and al., "Design and description of CS-ACELP: a toll quality 8 kb/s speech coder," *IEEE Trans. Speech and Audio*, vol. 6, no. 2, pp. 116-130, Mar. 1998.
[40] V. Ramamoorthy and N. Jayant, "Enhancement of ADPCM speech by adaptive post-filtering," *AT&T Bell Laboratories Tech. J.*, vol. 63, no. 8, pp. 1465-1475, Oct. 1984.
[41] J. H. Chen and A. Gersho, "Adaptive postfiltering for quality enhancement of coded speech," *IEEE Trans. on Speech and Audio Processing*, vol. 3, no. 1, pp. 59-71, Jan. 1995.
[42] J. Garcia, C. Marro, and B. Kövesi, "A PCM coding noise reduction for ITU-T G.711.1," in Proc. *Interspeech*, Brisbane, Australia, pp. 57-60, Sept. 2008.
[43] C. M. Konate, "Enhancing speech coder quality: improved noise estimation for postfilters," M.Eng. thesis, Dept. Elec. & Comp. Eng., McGill University, June 2011.
[44] R. Lefebvre, R. Salami, C. Laflamme, and J. P. Adoul, "High quality coding of wideband audio signals using transform coded excitation (TCX)," in Proc. *ICASSP*, Adelaide, vol. 1, pp. 193-196, 1994.
[45] J. H. Chen and D. Wang, "Transform predictive coding of wideband speech signals," in Proc. *Int. Conf. Acoustic, Speech and Signal Processing (ICASSP)*, Atlanta, GA, vol. 1, pp. 275-278, 1996.
[46] B. Edler and G. Schuller, "Audio coding using a psychoacoustic pre- and post-filter," Proc. *ICASSP*, Istanbul, vol. 2, pp. 881-884, 2000.
[47] S. Bruhn, V. Grancharov, B. Kleijn, J. Klejsa, M. Li, H. Pobloth, and S. Ragot, "The FlexCode Speech and Audio Coding Approach," *8th ITG Conference Speech Communication*, Aachen, Germany, pp. 1-4, Oct. 2008.
[48] *Audio Codec Processing Functions; Extended Adaptive Multi-Rate – Wideband (AMR-WB+) Codec; Transcoding Functions*, 3GPP TS 26.290, 2004.
[49] R. Lefebvre and C. Laflamme, "Spectral amplitude warping (SAW) for noise spectrum shaping in audio coding," in Proc. *Int. Conf. Acoustic, Speech and Signal Processing (ICASSP'97)*, Munich, vol. 1, pp. 335-338, 1997.
[50] X. Maitre, "7 kHz audio coding within 64 kbit/s," *IEEE Trans. on Selected Areas in Communications*, vol. 6, no. 2, pp. 283-298, Feb. 1988.
[51] B. Kovesi, S. Ragot, and A. Le Guyader, "Coding of digital audio signals," FR200700557792 patent application.
[52] J. Lapierre, R. Lefebvre, B. Bessette, V. Melanovsky, and R. Salami, "Noise shaping in an ITU-T G.711-interoperable embedded codec", Proc. *EUSIPCO*, Lausanne, Switzerland, Aug. 2008.
[53] M. Krasner, M. Berouti, and J. Makhoul, "Stability analysis of APC systems," in Proc. *Int. Conf. Acoustic, Speech and Signal Processing (ICASSP)*, Atlanta, GA, vol. 6, pp. 611-614, 1981.
[54] S. Ragot, B. Kovesi, and A. Le Guyader, "Controlling a noise-shaping feedback loop in a digital audio signal encoder," FR20100055037 patent application.
[55] B. Kövesi, S. Ragot, and A. Le Guyader, "An 64-80-96 kbit/s scalable wideband speech coding candidate for ITU-T G.711-WB standardization," in Proc. *Int. Conf. Acoustic, Speech and Signal Processing (ICASSP'08)*, Las Vegas, NV, pp. 4801-4804, 2008.
[56] S. Ragot et al, "ITU-T G.729.1: An 8-32 kbit/s scalable coder interoperable with G.729 for wideband telephony and voice over IP," *Proc. ICASSP*, Honolulu, vol. 4, pp. 529-532, Apr. 2007.
[57] B. Geiser, S. Ragot, and H. Taddei, "Embedded speech coding: from G.711 to G.729.1," Chapter 8 in *Advances in Digital Speech Transmission*, (R. Martin, U. Heute, C. Antweiler, eds.), New York: Wiley, Jan. 2008, pp. 201-248.
[58] Y. Hiwasaki et al, "G.711.1: a wideband extension to ITU-T G.711," *Proc. EUSIPCO*, Lausanne, Switzerland, Aug. 2008
[59] L. Miao et al, "G.711 Annex D and G.722 Annex B-New ITU-T superwideband codecs," in Proc. *Int. Conf. Acoustic, Speech and Signal Processing (ICASSP)*, Prague, pp.5232-5235, 2011.
[60] B. Kövesi, S. Ragot, and A. Le Guyader, "Encoding of an audio signal with noise transformation in a scalable encoder," FR20080057839 patent application.
[61] B. Kövesi, S. Ragot, and A. Le Guyader, "Encoding with noise shaping in a hierarchical encoder," FR20100053851 patent application.
[62] B. Kövesi et al, "Re-engineering ITU-T G.722: Low delay and complexity superwideband coding at 64 kbit/s with G.722 bitstream watermarking," in Proc. *Int. Conf. Acoustic, Speech and Signal Processing (ICASSP)*, Prague, pp. 5248-5251, 2011.

Manuscript received: February 27, 2012

Biographies

Stéphane Ragot (stephane.ragot@orange.com) received his diplôme d´ingénieur in telecommunications engineering from Telecom Bretagne, France, in 1997. He received his MSc and PhD degrees in electrical engineering from the University of Sherbrooke, Canada, in 2000 and 2003. From 1997 to 2003, he was a research assistant at the University of Sherbrooke. From 2000 to 2003, he was a research engineer at VoiceAge, Canada. Since 2003, he has been with France Telecom R&D/Orange Labs, France. He has contributed to the standardization of speech/audio coders in 3GPP and ITU-T. Since 2008, he has been vice chair of 3GPP SA4. His main research interests include source coding and speech/audio processing.

Balázs Kövesi (balazs.kovesi@orange.com) received his degree in electrical engineering from the Technical University of Budapest in 1992. He received his MSc degree from Telecom Bretagne, France, in 1993 and his PhD degree from the University of Rennes I, France, in 1997. He joined the Speech and Audio Coding Group of France Telecom/Orange as a postdoctoral fellow in 1997 and as a research engineer in 1998. His main research interests include speech and audio compression.

Alain Le Guyader (alain-le-guyader@orange.fr) received his doctorate in electronic engineering from Rennes University, Rennes, France, in 1978. In 1977, he joined CNET/France Telecom R&D, France. His main research interests include speech and audio coding and audio watermarking. Since 2009, he has been a part-time lecturer in speech and audio coding at the University of Rennes 1/ENSSAT, France.

D:\EMAG\2011-02-29/VOL9\CONTETN.VFT———1PPS/P47

MMT: The Next–Generation Media Transport Standard

Gerard Fernando
(ZTE USA Inc., 2425, N. Central Expressway, TX 75080, USA)

Abstract

In this paper, we discuss the development of MPEG media transport (MMT), which is a next–generation media transport standard effort by ISO/MPEG. The architecture and functional areas of MMT are described. The functionality of existing media transport is analyzed to determine whether there is a need for this new media standard. From this analysis, potential areas for standardization in MMT have been identified.

Keywords

MPEG; RTP; media transport

1 Introduction

The International Standards Organization/Moving Picture Experts Group (ISO/MPEG) has started developing a new media transport standard called MPEG media transport (MMT). ZTE Corporation and other companies are involved in this work. In this paper, we report on the progress of MMT standardization. We also analyze existing standards and protocols to determine whether there is indeed a need for this new standard.

It has been nearly 20 years since MPEG developed the MPEG–2 transport stream (TS) standard [1]. This standard is used widely in media–delivery solutions, such as cable, satellite, and terrestrial delivery of entertainment video. It is also used in some stored–media solutions. However, some aspects of MPEG–2 TS require updating to accommodate changed conditions. In the following, we discuss the reasons for updating MPEG–2 TS.

As with MPEG–2 TS, the real–time transport protocol (RTP) of the Internet Engineering Taskforce (IETF) [2] was developed in the mid 1990s. Since then, RTP has been regularly updated with adaptation formats for recently developed media compression standards. However, there are some key functionalities missing from both RTP and MPEG–2 TS, and this is arguably a good reason for developing MMT.

In section 2, we describe the features of existing media–delivery standards and highlight key changes in MPEG–2 TS and RTP. In section 3, we describe the architecture of MMT. In section 4, we list the functional areas of MMT. In section 5, we describe some key features that would be included in MMT. In section 6, we make some concluding comments on the viability of MMT.

2 Review of Existing Media–Delivery Standards

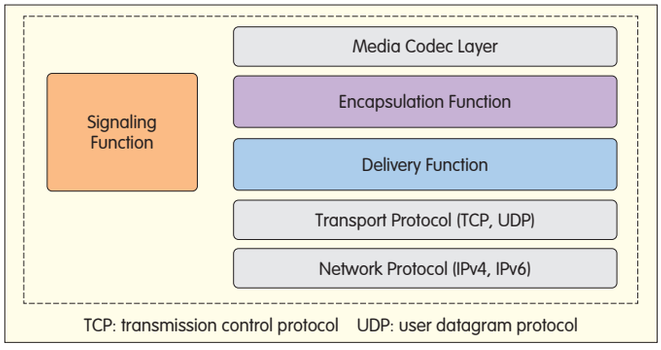
2.1 MPEG–2 Transport Stream

The MPEG–2 TS standard supports the combining of one or more elementary audio–video and data streams into single or multiple streams that are suitable for storage or transmission. This standard provides details about decoder buffer management to ensure media can be played back without buffer overflow or underflow. MPEG–2 TS provides the delivery clock as an in–band data channel that can be used by the receiving client to determine delay and jitter in the network.

MPEG–2 TS generation involves two stages. In the first stage, the media data is put into packetized elementary stream (PES) packets. This is analogous to encapsulation in MMT (section 4). Elementary–stream media data is packetized into PES streams according to access–unit boundaries, and presentation and decoding timestamps are inserted into the PES packet header. The second stage is delivery packetization. PES packets corresponding to audio, video, and other data formats are further packetized into smaller–sized packets of 188 bytes, and delivery timestamps are added.

MPEG–2 TS has withstood the test of time and is still highly relevant to the media–delivery industry. The main media–delivery services at the time of MPEG–2 TS

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT———1PPS/P48



▲Figure 1. MMT Architecture.

standardization were terrestrial broadcasting, and cable and satellite delivery. Since then, IPTV over multicast and, very recently, media delivery over Hypertext Transport Protocol (HTTP) have become important. Both HTTP live streaming (HLS) from Apple Computer and the DASH specification from ISO/MPEG [3] use MPEG–2 TS as a media format that is segmented for delivery over HTTP. Additionally, there are emerging hybrid services that use combinations of traditional delivery methods, and these require complex control and related operations on the part of the delivery standard. There are two areas where MPEG–2 TS is lacking: error control and support for network quality of service (QoS).

MPEG–2 TS has been updated several times (updates are referred to as amendments by the ISO) to include media types beyond those in the initial MPEG–2 TS. For some complex media formats, such as multiview coding (MVC) and scalable video coding (SVC), the original buffer model is inadequate. Furthermore, when MPEG–2 TS was developed, the required bitrates were on the order of 2 Mbit/s to approximately 20 Mbit/s, and this was for video with resolutions up to 1920 × 1080. With ultrahigh definition (UHD), the maximum bitrate to be supported is now near to 100 Mbit/s, and for these bitrates, the TS and PES packet sizes are not suitable.

All these reasons point towards the need for an updated MPEG–2 TS, and this is the justification for MMT.

2.2 Real–Time Transport Protocol

The real–time transport protocol (RTP) provides end–to–end delivery services for real–time data such as interactive audio and video. These services include payload type identification, sequence numbering, delivery timestamp insertion, and delivery monitoring. Initially, the developers of RTP did not want to support multiplexing of media in RTP; however, because there was a demand, several proposals were put forward that included multiplexing in RTP.

A key feature missing from RTP is quality of service (QoS) guarantee, and this is a feature that will be included in the new MMT standard.

3 MMT Architecture

The architecture for MMT is orthodox in terms of networking and media transport. Fig. 1 shows the architecture that has been agreed upon by experts participating in MMT standardization.

The architecture is divided into three functional areas: encapsulation, delivery, and signaling.

The encapsulation function defines the format for the encapsulation of encoded media data to be stored or to be carried as the payload of delivery protocols and networks. The delivery function provides formats and functionalities for transferring encapsulated media data from one network entity to another. The signaling function signals and controls delivery and consumption of the media.

4 Functional Areas of MMT

There is a clear separation of the functions of media encapsulation, media delivery and signaling within MMT.

4.1 Encapsulation

Encapsulation involves the following operations:

- Media packetization
- Media fragmentation
- Media synchronization
- Media multiplexing
- Insertion of timestamps to enable media synchronization, like lip synchronization
- Insertion of composition information. This includes spatial and temporal location of media objects in a given scene.
- Content protection. This includes conditional access and digital rights management.
- Container format that can be stored or packetized for delivery.

Encapsulation defines a media container that is not itself a physical storage format. Instead, this container may be stored or with further processing in the delivery layers of MMT, it may be ready to deliver. The main functions of the encapsulation layer are similar to PES encapsulation in MPEG–2 TS. The output of encapsulation is an MMT package.

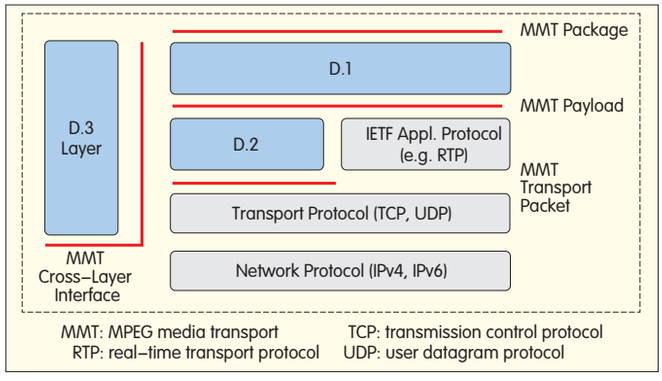
The MMT group has done much valuable work in the area of scene composition. The group have taken existing standards such as SMIL 2.0 [4] and LaSeR [5] as the basis for MMT composition and only made additions when there is broad consensus that such functional additions would enhance the MMT solution.

4.2 Delivery

In delivery, the encapsulated MMT package is taken as input, and the following operations are performed:

- Network packetization
- Network flow multiplexing
- Insertion of delivery stamps for use by client device to determine jitter etc.
- QoS operations
- Error handling. This includes application–layer forward–error correction (AL–FEC) and retransmission–based error handling, which is often referred to as automatic repeat request for re–transmission (ARQ).

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P49



▲Figure 2. Layers of the delivery functional area.

The delivery functional area is subdivided into D.1, D.2 and D.3 layers (Fig. 2).

Each delivery operation is performed in one of these three layers. The D.3 layer is not the same as the other two layers because its main function is to deliver messages between the other two delivery layers to enable cross–layer optimization.

4.2.1 D.1 Layer

The main data fields inserted into the D.1 layer are payload identification, fragmentation and aggregation of transport packets, information to enable content protection, and AL–FEC and information to enable random–access operations The D.1 layer generates the MMT payload.

Payload identification is required to determine the type of payload, including whether it is a media or signaling payload. Fragmentation and aggregation information allows the encapsulated media packets to be suitably structured according to the specific transport environment. Content protection and AL–FEC are required for media delivery; hence, the D.1 layer header indicates these functions. For random access operations to be performed efficiently on media data, information about random–access capability for a given media packet needs to be available at the transport level, and this obviates the need to inspect the media payload to determine random–access capability.

4.2.2 D.2 Layer

The D.2 layer header provides the delivery timestamp and QoS parameters. The D.2 layer function generates the MMT transport packet.

MMT requires a delivery timing model that provides a timestamp for synchronizing media streams and that calculates delay and jitter in networks. In the event required timing tolerances are not satisfied, any element on the delivery path should be able to readjust the timing relationships. At the current stage of the MMT development, it is assumed that each element on the delivery path has access to the universal time clock (UTC) from a remote clock source that operates the network time protocol (NTP) [6]. An alternative approach is for the delivery clock to be transmitted in–band with the media data, which is similar to using a program clock reference to derive the delivery clock in MPEG–2 TS solutions. With in–band clock delivery, there is no requirement for each element in the delivery path to have access to the UTC clock from a remote NTP server. Arguably, relaxing this requirement could make MMT more widely deployable.

In the D.2 layer, QoS fields are available so that network filtering can be performed based on these fields. It is expected that the QoS fields from the D.2 layer will be mapped to the corresponding fields in the IPv4 or IPv6 protocols.

4.2.3 D.3 Layer

The D.3 layer is also referred to as the cross–layer function because it provides the means of supporting cross–layer optimization. This requires exchanging QoS–related information between the application layer and underlying network layers. QoS–related information could be used for QoS management and adaptation such as flow control, session management, session monitoring, and error control.

4.3 Signaling Function

The signaling function is divided between the S.1 layer and S.2 layer (Fig. 3).

4.3.1 S.1 Layer

This layer is used for presentation session management. Signaling messages are exchanged between applications in the client device for media presentation, session management, and provision of information for media consumption.

4.3.2 S.2 Layer

This layer manages delivery sessions, which includes managing signaling messages that are exchanged between delivery end–points. These signaling messages are used for flow control, delivery session management, delivery session monitoring, error control, and hybrid network synchronization control. This is an important function for media delivery over hybrid networks.

5 Support for Error Control in MMT

Media delivery services need to work effectively in error–prone networks. However, where media delivery occurs

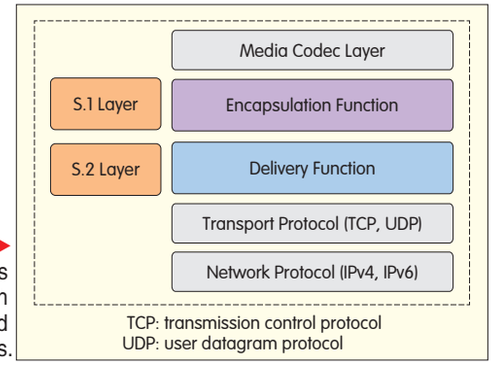


Figure 3. ▶ Signaling functions associated with encapsulation and delivery functions.

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P50

in a pull data model, with TCP as the underlying transport protocol, there is no need for explicit error control because this function is inherent in the packet loss detection and re–transmission in the TCP protocol.

For data delivery using the push model, where error control is not built into the underlying transport protocol, there is a need for an error–control function, and this is an area where the MMT standard could provide valuable added functionality. Any packet loss that may occur in the delivery stage is detected at the client device, then there are several methods that could be followed to mitigate packet loss. The following is a list of solutions to address such packet loss:

- AL–FEC
- ARQ
- Built–in error resilience at the codec level. This includes data portioning and redundant data generation. Scalable video coding could also be structured to enable error resilience.
- Error concealment at the client device

AL–FEC and ARQ require explicit signaling as well as extra media or supporting data to be delivered for such functions to be effective. Using ARQ for error control is very effective but is limited to services that do not need real–time response becauase this method requires retransmission of packets.

Error concealment and error recovery, which invlove introducing error resilience at the codec level, are commonly used in today's media delivery solutions. These techniques are reviewed in [7]. Because these approaches do not require any explicit signaling or extra media data transport there is no need to go into further detail here.

5.1 AL–FEC

With AL–FEC, the server first adds some redundant data to the transmitted packets using a predetermined FEC algorithm. There are several contenders for such FEC algorithms. At the receiving client device, once packet loss has been detected, missing information may be reconstructed. There are two different methods of signaling the client device about the specific FEC algorithm being used: The signaling information can be carried out–of–band or the signaling information can be carried in band. The use case would determine which of the two methods is applicable for a given media delivery solution. The server deployment for in–band signaling is simpler compared to that for out–of–band FEC signaling. As the bandwidth requirement increases for in–band signaling, the FEC parameters need to be signaled frequently in order to enable functions such as channel switching.

These two approaches to FEC signaling are prime candidates for standardization in MMT. Signaling data would be delivered within the signaling layers. Whether this signaling data is delivered in the same physical channel as the media data or whether it is delivered over a separate physical channel depends on the deployment scenarios.

5.2 ARQ

ARQ has particular benefits for media delivery services that do not require real–time response. In ARQ, the client device constantly sends acknowledgements to the server. If the server does not receive the acknowledgements in an expected time interval, it re–transmits the particular media data packet. The parameters of the ARQ process, for example, timeout duration, need to be signaled to the client device from the server. In this respect, the process of ARQ parameter signaling is similar to that for FEC signaling. The same trade–offs as for FEC are applicable for ARQ.

Hence, the same principle for standardization of the signaling information can be adopted in MMT.

6 Conclusions

In this paper, we have described the work that is currently being done within ISO/MPEG on the development of the MMT standard. The two main media–delivery standards that are in use today are MPEG–2 TS and RTP, but these have certain limitations. There is justification for MMT standardization focusing on the limitations that have been identified in this paper.

In addition to encapsulation and delivery functionalities, MMT also has signaling and composition functions. With these two areas, similar gap analyses with respect to existing standards and protocols need to be carried out in order to determine whether MMT should include such functionality or whether it is possible to rely on existing standards and protocols.

The MMT effort is a work in progress. Success of MMT depends to a large extent on whether the standards so far developed fill an actual gap in technology. ZTE Corporation and other companies are playing an active role in this standards effort, and this will help ensure MMT gives rise to a technology that is useful.

References

[1] *Information Technology — Generic Coding of Moving Pictures and Associated Audio Information: Systems*, ISO/IEC 13818–1: 2007.
[2] *RTP: A Transport Protocol for Real–Time Applications*, RFC3550, 2003.
[3] Information Technology— Dynamic Adaptive Streaming over HTTP (DASH)— Part 1: Media Presentation Description and Segment Formats, ISO/IEC 23009–1:2012.
[4] W3C. (2005). *Synchronized Multimedia Integration Language (SMIL 2.0), Second Edition* [Online]. Available: http://www.w3.org/TR/SMIL2/
[5] *Information technology — Coding of audio–visual objects — Part 20: Lightweight Application Scene Representation (LASeR) and Simple Aggregation Format (SAF)*, ISO/IEC 14496–20: 2008.
[6] *Network Time Protocol (Version 3) Specification, Implementation and Analysis*, RFC1305, 1992.
[7] Zhiji Zhao and Joern Ostermann, "Low complexity error control methods for scalable video streaming," accepted for publication in *ZTE Communications*, vol. 10, no. 2, Jun. 2012.

Manuscript received: February 29, 2012

Biography

Gerard Fernando (gerard.fernando@zteusa.com) received his PhD degree in electrical engineering from Imperial College, London. He has worked at Philips Research Labs and Sun Microsystems Inc. He is currently working at ZTE Corporation and is focused on standards activities in multimedia transport in ISO/MPEG and in IETF. He has worked on medical imaging, video signal processing, media transport over Internet, digital rights management and video codecs. He has authored several publications and also lead standards activities in ISO/MPEG and in IETF. He has several patents.

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P51

Low–Complexity Error–Control Methods for Scalable Video Streaming

Zhijie Zhao and Jörn Ostermann
(Institut für Informationsverarbeitung Leibniz Universität, Hannover D–30167, Germany)

Abstract

In this paper, low–complexity error–resilience and error–concealment methods for the scalable video coding (SVC) extension of H.264/AVC are described. At the encoder, multiple–description coding (MDC) is used as error–resilient coding. Balanced scalable multiple descriptions are generated by mixing the pre–encoded scalable bit streams. Each description is wholly decodable using a standard SVC decoder. A preprocessor can be placed before an SVC decoder to extract the packets from the highest–quality bit stream. At the decoder, error concealment involves using a lightweight decoder preprocessor to generate a valid bit stream from the available network abstraction layer (NAL) units when medium–grain scalability (MGS) layers are used. Modifications are made to the NAL unit header or slice header if some NAL units of MGS layers are lost. The number of additional packets that a decoder discards as a result of a packet loss is minimized. The proposed error–resilience and error–concealment methods require little computation, which makes them suitable for real–time video streaming. Experiment results show that the proposed methods significantly reduce quality degradation caused by packet loss.

Keywords

error resilience; error concealment; SVC; MDC

1 Introduction

Real–time video streaming over packet–switched networks can be impeded by packet loss, which often produces undesirable effects at the decoder. Usually, packet loss can make part of a frame, a whole frame, or even several frames undecodable using a standard decoder. Therefore, error–resilient coding and error–concealment techniques are widely used in video streaming systems to reduce the effect of transmission errors and to minimize end–to–end distortion in error–prone environments.

Error–resilient coding in the encoder produces redundancy, and this limits packet loss. Error–resilient coding tools for scalable video coding (SVC) can be classified as standard or non–standard. SVC supports several standard error–resilient coding tools, including intra–MB/picture refresh, slice coding, parameter sets, flexible MB order, and redundant slices/pictures [1]. Loss–aware rate–distortion–optimized mode decision [1], forward–error correction, and multiple–description coding (MDC) are non–standard error–resilient coding tools. MDC is used to code a video sequence into two or more bit streams, called descriptions, and these descriptions are transmitted using independent paths. Each description can be decoded independently so that the reproduction of the original source reaches a basic level of quality. A high level of quality can be achieved when all descriptions are reconstructed together. Some early forays into MDC include multiple–description scalar quantizer [2], MDC with pairwise correlating transform [3], multiple–description compensation schemes [4], multiple–state video coding [5], and multiple–description coding based on forward–error correction [6].

Alternatively, error concealment can be used in the decoder to passively reduce transmission errors. In this way, available and correctly decoded information is used without modifying source– and channel–coding schemes. The tools and structure of a special codec are also used to reduce video quality degradation. The temporal and spatial correlation between frames or within a frame is frequently used to conceal the artifacts caused by transmission errors. Motion data is one of the most important types of data for decoding a frame in hybrid video codecs, so motion copy and motion prediction are widely used.

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P52

With SVC, a video sequence is coded into one or more layers, and data–rate adaptation is allowed. This is an attractive solution for dealing with heterogeneous networks and different terminal capacities. The scalable extension of H.264/AVC is the latest SVC standard [7]. SVC provides temporal, spatial, and quality scalability that can be combined for greater adaptability to different network conditions or terminal capacities. In this paper, we describe low–complexity error–control methods for SVC. In particular, we propose a flexible, standard–compatible MDC method for error–resilient coding [8] and a low–complexity error–concealment method for SVC at the decoder when an MGS layer is used [9]. In section 2, we give an overview of work related to SVC and introduce our proposed methods. In section 3, we present simulation results. Section 4 concludes the paper.

2 Scalable Video Coding and Our Proposed Methods

2.1 Overview of Scalable Video Coding

The SVC extension of H.264/AVC incorporates the key features of H.264/AVC as well as new techniques to improve scalability and coding efficiency. Temporal scalability can be achieved by using hierarchical prediction structures, for example, hierarchical B–pictures or non–dyadic hierarchical prediction structures. Scalable quality can be achieved by using coarse–grain quality–scalable coding (CGS) and medium–grain quality–scalable coding (MGS). Spatial scalability can be achieved by using multilayer coding, and each layer corresponds to a supported spatial resolution. Redundancy between spatial layers can be further exploited by interlayer prediction mechanisms such as interlayer motion prediction, interlayer residual prediction, and interlayer intraprediction.

H.264/AVC has a video coding layer (VCL) and a network abstraction layer (NAL). In the VCL, a coded representation of the input video signal is generated, and in the NAL, this coded representation is fragmented. The NAL provides header information to ease the use of VCL data. Being an extension of H.264/AVC, SVC has the H.264/AVC

MGS supports packet–based quality scalability by distributing the transform coefficients of a slice. When MGS is used to provide quality scalability, an access unit can include several MGS NAL units. With MGS, the enhancement layer transform coefficients can be distributed between a maximum of 16 slices, and each slice corresponds to an NAL unit. MGS layers inside each dependency layer are identified by a quality identifier.

2.2 Work Related to Scalable Video Coding

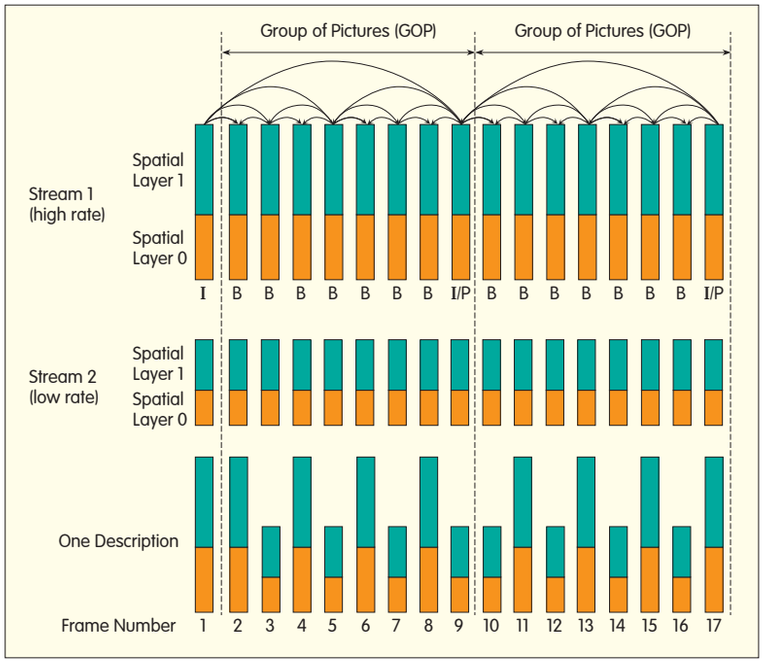
Combined SVC and MDC has attracted the interest of researchers. Multiple–state video coding [5] splits the input video into subsequences in the temporal domain and encodes each subsequence as an independent description. This coding can be used to generate scalable multiple descriptions with half the temporal resolution. In [10], two complementary descriptions are generated for the high–pass frame of each enhancement layer. These descriptions are generated by assigning only half the motion vectors and texture information of the original coded stream in each description. Alternatively, two descriptions are generated for the base layer of SVC by downsampling the residual data [11]. A general multirate allocation scheme for multiple–description coding is proposed in [12]. This scheme has been used in JPEG 2000 and H.264/AVC to produce multiple descriptions [13]. A fully redundant unbalanced MDC scheme for SVC is proposed in [14]. In [15], an SVC bit stream produces two balanced descriptions by assigning MGS NAL units to one of the descriptions on alternating frames with a period of two group of pictures (GOPs). The spatial scalability of SVC is not taken into account.

Several error–concealment methods have been proposed for SVC [1], [16]. Intralayer error concealment and interlayer error concealment deal with frame loss when there are two spatial layers. Error–concealment algorithms copy a picture from another view, generate a motion vector from the same spatial layer, or upsample motion and residual data from the available base layer to generate a lost picture. With a slice support, the error–concealment scheme in [16] not only uses reference–frame information but also uses correctly received information from the same frame and from higher spatial layers. In this way, packet loss can be concealed on the base layer of the compressed stream [17]. In [18], a frame–loss error–concealment algorithm based on hallucination is proposed for the spatial–enhancement layer. A training database of hallucinations for missing enhancement frames is generated from the two most–recently decoded I or P frames. This algorithm performs better than the motion and residual upsampling proposed in [16]. Another error–concealment method for whole–picture loss in hierarchical B–picture coding is proposed [19]. This method performs better than the error–concealment method in SVC reference software. In contrast, a low–complexity error–concealment algorithm can be used in the network abstraction layer of SVC [20]. The algorithm in [20] recognizes the bit–stream structure and creates a valid sequence of packets from the received packets. In this paper, we call this method NAL unit removal. Unlike other error–concealment algorithms, NAL unit removal does not generate missing frames. In [20], the case of two spatial layers with fine–grain scalability (FGS) is considered. A similar work is [21], which builds on the work in [16] by supporting quality scalability using FGS layer [22].

2.3 Standard–Compatible MDC for SVC

In SVC, each quality base layer and quality–enhancement layer of a spatial layer is usually quantized using different step sizes. The coefficients of a quality refinement picture are quantized with a quantization parameter (QP) and can be distributed over several layers. Each of these layers contains partial refinement coefficients that use MGS. QPs can be cascaded over the temporal levels according to a given pattern, or default QP cascading can be used.

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P53



▲Figure 1.Two descriptions are generated by combining bit streams.

The parameters for quantization step sizes are stored in several places, including the picture parameter set (PPS), slice header (SH), and macroblock layer. The lum quantization parameter is initially QP_Y, and this value is used for all macroblocks in the slice until modified by *mb_qp_delta* in the macroblock layer. QP_Y is given as

$$QP_Y = 26 + pic\_init\_qp\_minus\ 26 + slice\_qp\_delta \quad (1)$$

where *pic_init_qp_minus* 26 is the initial QP_Y of –26 for each slice and is stored in the PPS, and *slice_qp_delta* is a value that changes the quantizer step size at each slice and is stored in the SH. The slice header contains a codeword that indicates the PPS to be used, and the PPS includes the identifier of the active sequence parameter set (SPS). An active SPS remains unchanged throughout a coded video sequence, and an active PPS remains unchanged within a coded picture.

Two standard–compatible scalable descriptions can be produced by combing streams pre–encoded at different bit rates. To do this, we change only the quantization step size in order to generate low– and high–bit–rate steams. Moreover, in order to combine NAL units of different descriptions at the decoder, both descriptions use the same PPSs and SPSs. Different quantization step sizes are indicated by *slice_qp_delta* parameter in the SH. Each description generated by combined pre–encoded streams supports spatial, temporal, and quality scalability and can be decoded by a standard SVC decoder.

To generate balanced multiple descriptions, different bit streams are combined so that high– and low–rate NAL units from alternating frames can be assigned to a description over a period of two GOPs. Fig. 1 shows the proposed combination scheme for generating descriptions.

For the first description in Fig. 1, the even–numbered frames of the first GOP come from a high–bit–rate stream, and the odd–numbered frames come from a low–bit–rate stream. The even–numbered frames of the second GOP come from a low–bit–rate stream, and the odd–numbered frames come from a high–bit–rate stream. For the second description, the even–numbered frames of the first GOP come from a low–bit–rate stream, and the odd–numbered frames come from a high–bit–rate stream. The even–numbered frames of the second GOP come from a high–bit–rate stream, and the odd–numbered frames come from a low–bit–rate stream. This produces two descriptions that are balanced in terms of bit rate and quality.

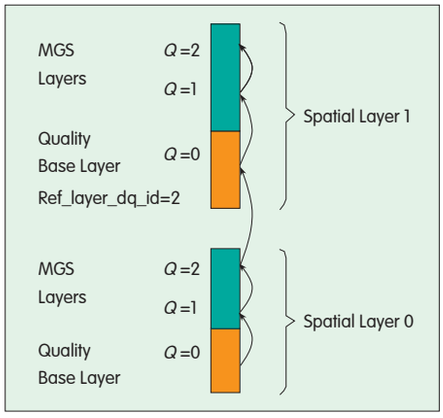
At the decoder, a preprocessor is placed before a standard SVC decoder to parse newly arrived packets and extract the packets from the highest–quality bit stream. However, without the preprocessor, each description is still decodable using a SVC decoder. In our proposed scheme, a side decoder is not needed for an MDC description. The received packets from both descriptions are parsed and arranged into a new stream that is passed to an SVC decoder.

2.4 Low–Complexity Error Concealment for SVC

In H.264/AVC and SVC, an NAL unit starts with a single–byte header that signals the type of contained data, for example, NAL unit. In SVC, a three–byte extension header is used to indicate the scalable information for the coded slice in a scalable extension and for the prefix NAL unit. The parameter's dependency identifier (D), temporal identifier (T), and quality identifier (Q) determine which spatial layer, temporal layer, and quality layer an NAL unit belongs to. An access unit corresponds to one picture after decoding and comprises several consecutive NAL units with specific properties. In the SVC design, MGS is used and not FGS; however, no research has been done on MGS quality scalability. We therefore propose extending the NAL–unit–removal algorithm to deal with packet loss in the MGS layer. Our approach is motivated by multilayer adaptation for an MGS–based SVC bit stream [23]. NAL unit headers or slice headers are parsed to produce a valid bit stream from the available NAL units at the receiver. When a frame belonging to the highest temporal level is lost, the handling method of the NAL–unit–removal algorithm is changed.

When MGS is used to provide scalable quality, an access unit can include several MGS NAL units. With the MGS, the enhancement–layer transform coefficients can be distributed between a maximum of 16 slices, and each slice corresponds to an NAL unit. MGS layers inside each dependency layer are identified by a quality identifier. For the quality base layer of a spatial–enhancement layer, a syntax element called *ref_layer_dq_id* in the slice header is used to signal which MGS layer is used for interlayer prediction (assuming that

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P54



◀Figure 2. A layer–prediction structure in an access unit.

interlayer prediction is enabled). For quality–refinement MGS layers with quality identifier $Q > 0$, the preceding quality layer with quality identifier $Q-1$ is used for interlayer prediction. Fig. 2 shows a layer–prediction structure in an access unit with two spatial layers and two MGS layers. If an MGS layer is employed as reference layer for interlayer prediction and is lost, the received bit stream becomes invalid for a standard decoder. For example, the decoding of MGS layer $Q = 2$ in spatial layer 0 depends on the MGS layer $Q = 1$ in spatial layer 0. If MGS layer $Q = 1$ in spatial layer 0 is lost and the other NAL units are received, the bit stream cannot be decoded by a standard decoder. The packets of MGS layer $Q = 2$ in spatial layer 0 and the whole spatial layer 1 can be discarded [20]. In the following, we discuss how to deal with MGS–layer loss.

To simplify the description without losing generality, we consider the case where the source video is coded with two spatial layers and two MGS layers within each spatial layer. Table 1 shows the NAL unit order in a bit stream for a group of pictures (GOP) of size four that has three temporal levels. With the NAL unit–removal method, if a NAL unit of a GOP is lost, a valid NAL unit order with lower spatial resolution and/or lower frame rate is chosen. With multiple–quality–layer coding, if an NAL unit not from the highest MGS layer is lost, a valid NAL unit order with a lower–quality layer is chosen. For example, if the 11th NAL unit (MGS layer 1) of a GOP in Table 1 is lost, the 12th NAL unit, which belongs to dependant MGS layer 2, is also discarded to create a valid bit stream, even if the 12th NAL unit is correctly received. Because the slice data of the MGS quality–refinement layers include different distributions of transform coefficients, the 12th NAL unit can still be used to improve the decoded image quality. Therefore, we do not discard the higher MGS layers if one or several lower MGS layers are lost. At the client, we use the same layer–dependent modification described in [23], which is made for data–rate adaptation at the server. If the lost MGS layers belong to spatial layer 1, only the *quality_id* parameter of the NAL headers of the remaining MGS NAL units in spatial layer 1 need to be modified so that continuity of *quality_id* values is maintained. Because an NAL unit header is not compressed, the modification requires very low computing power.

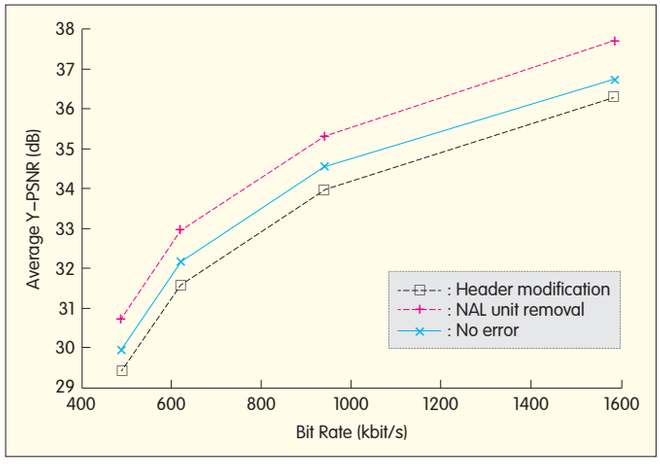
When the 15th NAL unit in Table 1 is lost and the layer–prediction structure in Fig. 2 is used, NAL units 16 to 18 are discarded to create a valid bit stream. To use the received higher MGS layers of a spatial–enhancement layer and to maintain a standard decoder–compliant bit stream, both NAL unit header and slice header are modified. When an MGS NAL unit in spatial layer 0 is lost, the header of the NAL units within spatial layer 0, and the slice header of the quality base layer in spatial layer 1, need to be changed. If the maximum–quality identifier in spatial layer 0 is changed, *ref_layer_dq_id* in the slice header of the quality base layer in spatial layer 1 is updated according to maximum–quality identifier. The slice header is coded using Exp–Golomb codes in SVC, and parsing and modification is not time consuming. Fig. 3 shows the reconstructed video quality of the joint test sequence used in section 0. We discard the first MGS layer of the spatial enhancement layer. In our proposed method, the NAL header of the second MGS layer is modified in order to maintain a valid bit stream. In the NAL unit–removal method, only the quality base layer of the spatial–enhancement layer is kept in order to maintain a valid bit stream. The proposed method improves average luma PSNR by 0.57 dB. Our method may introduce drift, so a

▼Table 1. NAL unit order in a bit stream for a GOP (four frames) with two spatial layers, three temporal layers and, two MGS layers

No.	D	T	Q	MGS
1	0	0	0	
2	0	0	1	X
3	0	0	2	X
4	1	0	0	
5	1	0	1	X
6	1	0	2	X
7	0	1	0	
8	0	1	1	X
9	0	1	2	X
10	1	1	0	
11	1	1	1	X
12	1	1	2	X
13	0	2	0	
14	0	2	1	X
15	0	2	2	X
16	1	2	0	
17	1	2	1	X
18	1	2	2	X
19	0	2	0	
20	0	2	1	X
21	0	2	2	X
22	1	2	0	
23	1	2	1	X
24	1	2	2	X

D: dependency identifier Q: quality identifier
MGS:medium–grain scalability T: temporal identifier

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P55



▲Figure 3. Average distortion when the first MGS layer in spatial layer 1 is discarded.

two–alternative forced–choice test was performed to assess the subjective quality of our method and the NAL unit–removal method in case the first MGS layer of the spatial–enhancement layer is lost. Two short videos were shown sequentially, and observers had to choose the one they thought was higher quality. For low and medium qualities, the proposed method is preferred, but for higher qualities, the NAL unit–removal method is preferred because of its smoother motion rendition.

With the NAL unit–removal method, if a quality–base–layer NAL unit of the highest temporal layer is lost, an entire temporal layer is removed. For example, if the 13th NAL unit is lost, NAL units 14 to 24 are discarded to arrange a valid bit stream, even if these units are received. However, if hierarchical B picture is used for temporal scalability, the highest temporal layers are B pictures and are not used as reference frames. This means that if one frame of the highest temporal layer is lost, it does not affect the other frames in the temporal layers. In our proposal, the remaining highest temporal layers are retained. The missing frame can be concealed using frame copy or other error–concealment methods for whole–frame loss.

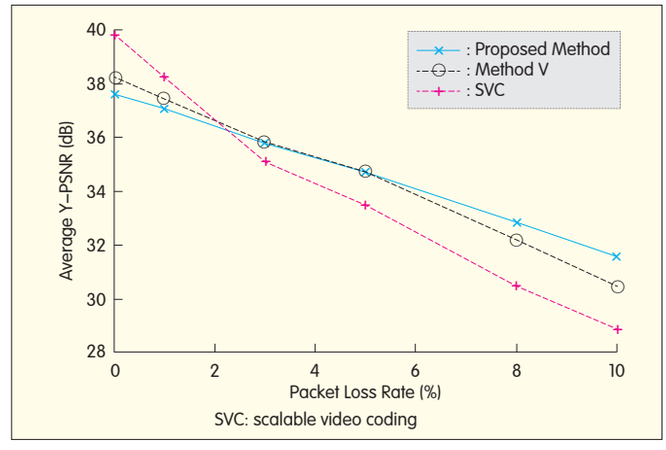
3 Experiment Results

In this section, we present experiment results for the proposed error–resilient and error–concealment methods. JSVM 9.18 SVC reference software was used to encode the input sequences [21]. The tested bits streams had three video sequences—Foreman, Mobile, and Akiyo—combined into a single sequence to produce long test–bit streams. Spatial layer 0 had quarter common intermediate format (QCIF) resolution, and spatial layer 1 had common intermediate format (CIF) resolution. The joint sequence contained 897 frames; the GOP size was 8 frames; and an I frame was used as the key picture. The RTP packet size was limited to 1400 bytes, and packet loss in the transmission channel was simulated by a two–state Markov model—where a good state means packets are received correctly and promptly, and a bad state means packets are lost.

3.1 Experiment Results for the Proposed MDC Method

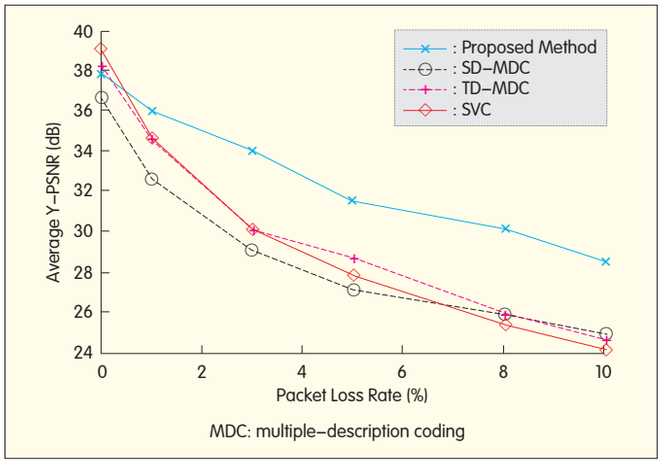
In this experiment, we used a streaming scenario with path diversity. The two descriptions are delivered through independent paths. In case of packet loss, all these paths have the same packet–loss probability. Five packet–loss ratios were used: 1%, 3%, 5%, 8% and 10%. We assume that parameter sets are conveyed using a reliable transport mechanism. If spatial scalability is supported, a coded bit stream contains two spatial layers (QCIF and CIF), four temporal levels, and one quality layer. CIF resolution is used if spatial scalability is not considered. For simplicity, spatial base layers of the pre–encoded streams are quantized by the same QP, and only the spatial–enhancement layers or quality–enhancement layers are quantized using two different QPs. Where SVC cannot decode the base layer, or the SMDC receiver lacks both descriptions, one or several frames cannot be decoded. In this case, frame copy is used as error concealment.

First, we compare our proposed SMDC scheme with single–description SVC and method V proposed in [15]. Method V is based on SVC with both descriptions containing the active base layer and every other quality–enhancement layer. Therefore, in method V, the redundancy is only the base layer. In the case of no packet loss, the proposed scheme and method V pay a penalty of reduced coding efficiency compared with single–description SVC. Method V has slightly less coding–efficiency loss than the proposed method. Fig. 4 shows the average luma PSNR as a function of the network packet–loss rate for the joint sequence of Foreman, Mobile, and Akiyo. The proposed scheme outperforms single–description SVC when the packet–loss rate is greater than 3% and outperforms method V when the packet–loss rate is greater than 5% (Fig. 4). At 10% of the packet–loss rate, the gain over single–description SVC is 2.6 dB, and the gain over method V is about 1.1 dB. When the packet–loss rate is less than 2%, the proposed SMDC scheme is inferior to

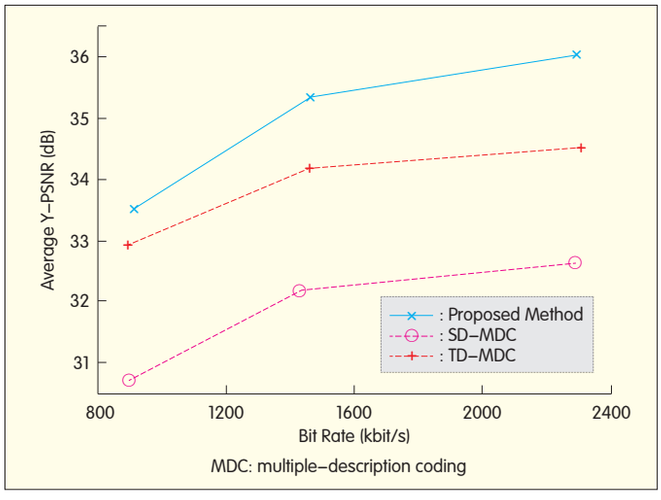


▲Figure 4. Average Y–PSNR vs. packet loss rate, without spatial scalability.

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT———1PPS/P56



▲Figure 5. Average Y–PSNR vs. packet loss rate, with spatial scalability.



▲Figure 6. Y–PSNR compared to TD–MDC and SD–MDC with 1% packet loss rate.

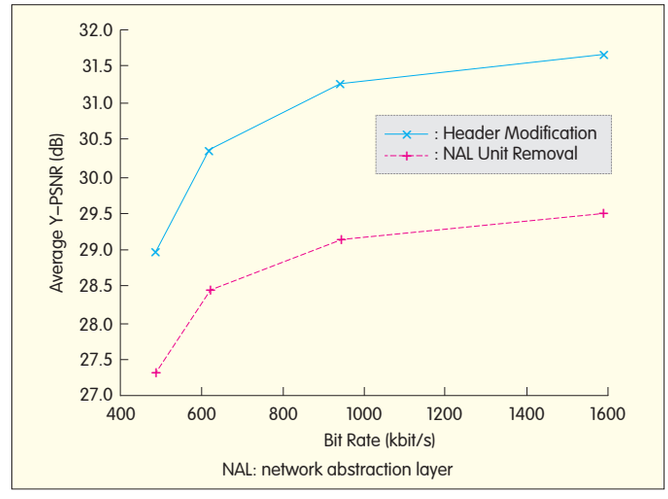
single–description SVC, and when the packet–loss rate is less than 3%, the proposed SMDC method is inferior to method V. The additional redundancy introduced in the proposed scheme plays a minor role at low packet–loss rates. However, method V in [15] cannot be extended to support spatial scalability.

To test performance when supporting spatial scalability, we compare our proposed scheme with single–description SVC, spatial downsampling (SD–MDC), and temporal downsampling MDC (TD–MDC). In SD–MDC and TD–MDC, an original video is first downsampled into two subsequences in the spatial and temporal domain, respectively. Then, the two subsequences are independently encoded by an SVC encoder. Fig. 5 shows how our scheme performs compared with single–description SVC, SD–MDC, and TD–MDC in terms of average Y–PSNR() versus packet–loss rate. The proposed scheme performs best at a packet–loss loss rate of 1–10%. At a 10% loss rate, the proposed scheme outperforms the SD–MDC and TD–MDC by approximately 3.5 dB and 3.9 dB, respectively. Because the descriptions of SD–MDC and TD–MDC are separately encoded, the losses of packets from one description cannot be effectively compensated from the received packets of the other description. However, the proposed scheme can still produce a whole spatial and temporal resolution video when one description is corrupted. Hence, the redundancy introduced in the proposed scheme is more beneficial than SD–MDC and TD–MDC in the case of packet loss. Although single–description SVC has the highest coding efficiency, the proposed scheme has a similar gain over single–description SVC. The gain is 4.3 dB at a 10% packet–loss rate. Fig. 6 shows the average Y–PSNR versus bit rate compared with SD–MDC and TD–MDC at 1% packet–loss rate. The results show that the proposed method is superior to SD–MDC and TD–MDC over the encoding bit rates 912 kbit/s, 1460 kbit/s, and 2294 kbit/s, where the redundancies are 28%, 31%, and 33%, respectively.

3.2 Experiment Results for the Proposed Error–Concealment Method

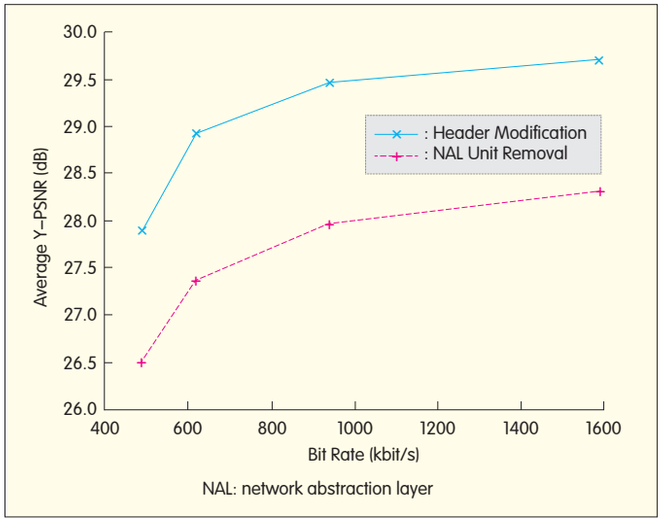
The proposed method is implemented as a preprocessing unit before a standard decoder in order to arrange a valid bit stream from the received packets. In these tests, each spatial layer has four temporal layers and two MGS layers. The first MGS layer contains four transform coefficients, and the second MGS layer includes 12 transform coefficients for each spatial layer. The QP difference between the quality base layer and the quality–enhancement layer is set to three. The default cascading of quantization parameters over the temporal levels is used. Hierarchical B picture is also used. In the experiments, we assume that packets of the quality base layer in spatial layer 0 are protected and not lost. Packet–loss ratios of 3%, 5%, and 10% are used. For decoded frames with a spatial resolution of QCIF, we use the upsampling filter in SVC to produce the spatial–resolution CIF.

Fig. 7 shows the rate–distortion curves for the

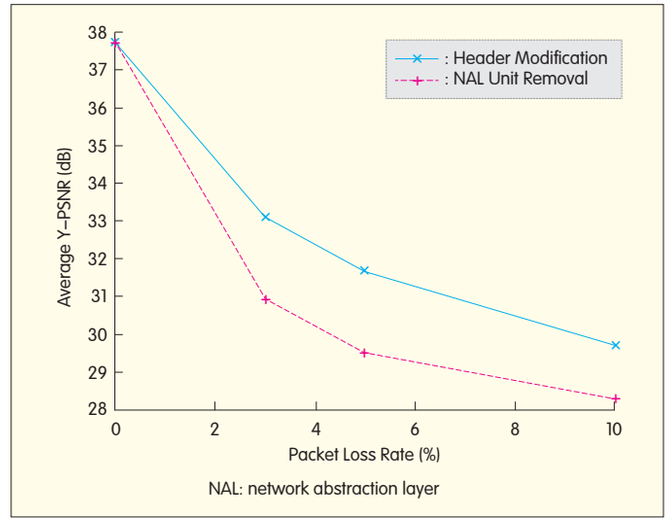


▲Figure 7. Average distortion, as a function of encoding rate, with 5% packet loss rate (Joint sequence Foreman, Mobile and Akiyo).

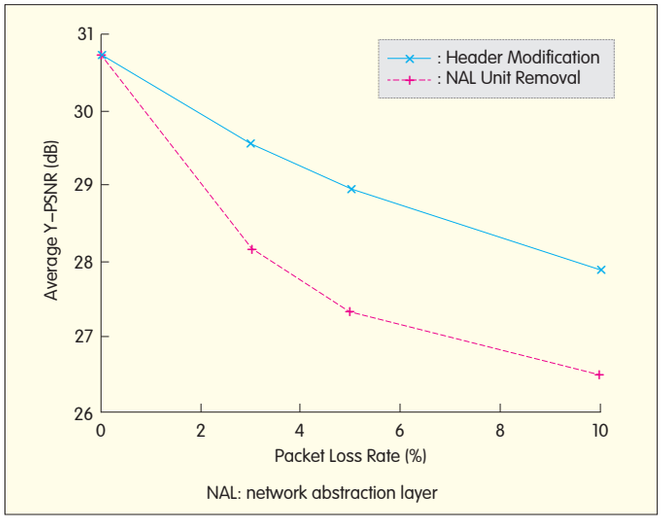
D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P57



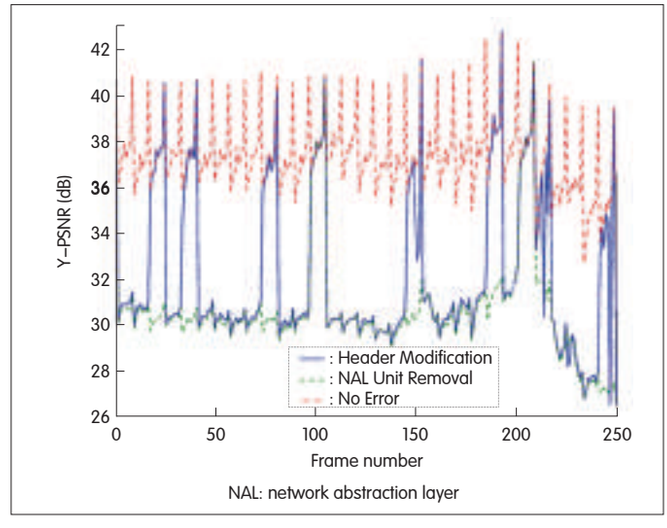
▲Figure 8. Average distortion vs. encoding rate for 10% packet–loss rate (Joint sequence Foreman, Mobile and Akiyo).



▲Figure 9. Average distortion vs. packet–loss rate (Joint sequence Foreman, Mobile and Akiyo, 1590 kbit/s).



▲Figure 10. Average distortion vs. packet–loss rate (Joint sequence Foreman, Mobile and Akiyo, 488 kbit/s).



▲Figure 11. Y–PSNR between NAL unit–removal and proposed method at 10% packet–loss rate (Foreman, from the joint sequence at 1590 kbit/s).

header–modification and NAL unit–removal methods with 5% packet–loss rate, and Fig. 8 shows the rate–distortion curves for the header–modification and NAL unit–removal methods with 10% packet–loss rate. We compare the proposed method with the NAL unit–removal method only because the proposed method does not substitute methods for concealing a frame loss but only complements them. Fig. 7 and Fig. 8 show that header–modification outperforms NAL unit–removal over the entire considered range of bit rates. For a 5% packet–loss rate, header modification gains 2.16 dB on average, and for a 10% packet–loss rate, NAL unit removal gains 1.55 dB on average for the joint sequence.

To further evaluate the performance of the proposed method, we determine how the average luma PSNR changes in relation to packet–loss rate. Figs. 9 and 10 show that the proposed method outperforms the NAL unit–removal method at all the three simulated packet–loss rates. The proposed method gains a maximum of 1.64 dB over the NAL unit–removal method when the packet–loss rate for the 488 kbit/s stream is 5%, and this can be as high as 2.16 dB when the packet–loss rate is 5% for the 1590 kbit/s stream. At 10% packet–loss rate, the proposed method improves average luma PSNR by 1.4 dB over the NAL unit–removal method at both bit rates.

Fig. 11 shows luma PSNR against the number of frames. It shows how luma PSNR changes for a Foreman sequence taken from the joint sequence coded at 1590 kbit/s with 10% packet–loss rate. The proposed method still uses the correctly received packets of higher MGS layers in case a lower MGS layer is lost, so the proposed method provides

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P58

much better video quality than NAL unit removal.

4 Conclusion

In this paper, we have proposed a standard–compatible MDC scheme for SVC based on combined pre–encoded streams. This scheme is designed for video streaming applications in error–prone environments. At the decoder, an error–concealment method in the NAL in case that MGS is used for the scalable extension of H.264/AVC is presented. Experiment results show that the proposed MDC and error–concealment methods can improve video quality in error–prone environments. The proposed methods have low computational complexity and require low computing power. Hence, they are suitable for real–time scalable video streaming.

References

[1] Y. Guo, Y. Chen, Y. K. Wang and et al., "Error resilient coding and error concealment in scalable video coding," in *IEEE Trans. on Circ. Syst. for Video Tech.*, vol. 19, no. 6, pp. 781–795, 2009.
[2] V.A. Vaishampayan, "Design of multiple description scalar quantizers," in I*EEE Trans. Info. Theory*, vol. 39, no. 3, pp. 821–834, 1993.
[3] Y. Wang, M.T. Orchard, V.A. Vaishampayan, A.R. Reibman, "Multiple description coding using pairwise correlating transforms," in *IEEE Trans. Image Processing*, vol. 10, no. 3, pp. 351–366, 2001.
[4] C.S. Kim, S.U. Lee, "Multiple description motion coding algorithm for robust video transmission," in *Proc. IEEE Int. Symp. Circ. Syst*., Geneve, Switzerland, Mar. 2000, pp. 717–720.
[5] J. G. Apostolopoulos, "Reliable video communication over lossy packet networks using multiple state encoding and path diversity," in *Visual Commun. and Image Processing (VCIP)*, vol. 4310, Jan. 2001, pp. 392–409.
[6] R. Puri, K. Ramchandran, K.W. Lee, V. Bharghavan, "Forward error correction (FEC) codes based multiple description coding for internet video streaming and multicast," in *Signal Processing: Image Communication*, vol. 16, no. 8, pp. 745–762, May 2001.
[7] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard, " in *IEEE Trans. Circ. Syst. for Video Tech.*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.
[8] Z.J. Zhao, J. Ostermann, "Video Streaming Using Standard–Compatible Scalable Multiple Description Coding Based on SVC," in *Proc. 17th IEEE Int. Conf. on Image Processing*, Hong Kong, Sep. 2010, pp. 1293–1296.
[9] Z.J. Zhao, J. Ostermann, "Error Concealment in the Network Abstraction Layer for Medium Grain Scalability of SVC," in *Visual Commun. and Image Processing 2010*, Proc. SPIE, vol. 7744, pp: 77442P–7744P–8, July 2010.
[10] H. Mansour, P. Nasiopoulus, V. Leung, "A flexible multi–rate allocation scheme for balanced multiple description coding applications," in *Proc. IEEE Int. Symp. Signal Processing and Info.* Tech., Athens, Greece, Nov. 2005, pp. 1–4.
[11] Z.J Zhao, J. Ostermann and H.X Chen, "Low Complexity Multiple Description Coding for the Scalable Extension of H.264/AVC," in *Proc. Picture Coding Symposium (PCS'09)*, Chicago, IL, May 2009, pp. 261–264.
[12] T. Tillo, E. Baccaglini and G. Olmo, "A flexible multi–rate allocation scheme for balanced multiple description coding applications," in *Proc. 7th IEEE Workshop on Multimedia Signal Processing*, Nov. 2005.
[13] T. Tillo, E. Baccaglini and G. Olmo, "Multiple descriptions based on multirate coding for JPEG 2000 and H.264/AVC," in *IEEE Trans. on Image Processing*, vol. 19, no. 7, pp. 1756–1767, Jul. 2010.
[14] P. Schelkens, A. Gavrilescu, A. Munteanu and et al., "Error–Resilient Transmission of H.264 SVC Streams over DVB–T/H and WIMAX Channels with Multiple Description Coding Techniques," in *Proc. 15th European Signal Processing Conference*, Poznan, Poland, Sep. 2007, pp. 1995–1999.
[15] T. Berkin Abanoz and A. Murat Tekalp, "SVC–based scalable multiple description video coding and optimization of encoding configuration," in *Signal Processing: Image Communication*, vol. 24, no. 9, pp. 691–701, Oct. 2009.
[16] Y. Chen, K. Xia, F. Zhang and et al., "Frame loss error concealment for svc," in J. *Zhejiang University Science A*, vol. 7, no. 5, pp. 677–683, 2006.
[17] T. Keränen, J. Vehkaperä, and J. Peltola, "Error concealment for svc utilizing spatial enhancement information," in *Proc. 4th Int. Mobile Multimedia Commun. Conf. (MobiMedia '08)*, Oulu, Finland, July 2008, article 10.
[18] Q.R. Ma, F. Wu and M.T. Sun, "Error concealment for spatial scalable video coding using allucination," in *Proc. IEEE Int. Symp. Circ. Syst. (ISCAS'09)*, Valencia, Spain, pp. 129–132, 2009.
[19] X.Y. Ji, D.B. Zhao and W. Gao, "Concealment of whole–picture loss in hierarchical b–picture scalable video coding," in *IEEE Trans. Multimedia*, vol. 11, no. 1, pp. 11–22, 2009.
[20] D.T. Nguyen, M. Shaltev, M and J. Ostermann, "Error concealment in the network abstraction layer for the scalability extension of H.264/AVC," in *Proc. Int. Conf. Commun. Electronics (ICCE06)*, Beijing, pp. 274–278, 2006.
[21] M. Stoufs, A. Munteanu, J. Cornelis and P. Schelkens, "Error concealment for the scalable extension of H.264/mpeg–4 avc," in P*roc. Picture Coding Symp. (PCS2007)*, Lisbon, Portugal, Nov. 2007.
[22] J. Reichel, H. Schwarz and M. Wien, Joint scalable video model 11 (jsvm 11)," Joint Video Team, doc.JVT–X202, July 2007.
[23] T. C. Thang, J. W. Kang, J. J. Yong and J. G. Lee, "Multilayer adaptation for MGS–based SVC bitstream", in *Proc. 16th ACM Int. Conf. on Multimedia*, Vancouver, BC, pp. 689–692, 2008

Manuscript received: January 17, 2012

Biographies

Zhijie Zhao(zhao@tnt.uni–hannover.de) received his MSc degree in communications and information systems from Jilin University. He is currently working towards his PhD degree at the Insitut für Informationsverarbeitung, Leibniz University, Hannover, Germany. His research interests include video streaming and video coding.

Jörn Ostermann has studied electrical engineering and communications engineering at the University of Hannover and Imperial College London. He received his Dipl.–Ing. and Dr.–Ing. degrees from the University of Hannover in 1988 and 1994. From 1988 to 1994, he was also a research assistant at the Institut für Theoretische Nachrichtentechnik and conducted research on low bit–rate, object–based analysis–synthesis video coding. From 1993 to 1994, he chaired the European COST 211 sim group coordinating research in low–bitrate video coding. From 1994 to 1995 he worked on video coding in the Visual Communications Research Department at AT&T Bell Labs. From 1996 to 2003, he was a member of Image Processing and Technology Research Team within AT&T Labs–Research. In 1998, he received the AT&T Standards Recognition Award and the ISO award. Since 2003, he has been a full professor and head of the Institut für Informationsverarbeitung at Leibniz Universität, Hannover, Germany. In 2007, he became head of the Laboratory for Information Technology at the same university. Since 2008, he has been the chairperson of the MPEG Requirements Group (ISO/IEC JTC1 SC29 WG11). Jörn was a scholar of the German National Foundation.

Dr. Ostermann has organized the evaluation of video tools to start defining the MPEG–4 standard. He chaired the Adhoc Group on the coding of arbitrarily–shaped objects in MPEG–4 video. He is a fellow of the IEEE and member of the IEEE Technical Committee on Multimedia Signal Processing. He has been the chair of the IEEE CAS Visual Signal Processing and Communications (VSPC) Technical Committee and has also been a Distinguished Lecturer of the IEEE CAS Society. He has published more than 100 research papers and book chapters. He is coauthor of a graduate–level text book on video communications and holds more than 30 patents.

AD Index

Back Cover:
ZTE Corporation

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P59

Key Technologies in Mobile Visual Search and MPEG Standardization Activities

Ling–Yu Duan, Jie Chen, Chunyu Wang, Rongrong Ji, Tiejun Huang, and Wen Gao
(Institute of Digital Media, Peking University, Beijing 100871, China)

Abstract

Visual search has been a long–standing problem in applications such as location recognition and product search. Much research has been done on image representation, matching, indexing, and retrieval. Key component technologies for visual search have been developed, and numerous real–world applications are emerging. To ensure application interoperability, the Moving Picture Experts Group (MPEG) has begun standardizing visual search technologies and is developing the compact descriptors for visual search (CDVS) standard. MPEG seeks to develop a collaborative platform for evaluating existing visual search technologies. Peking University has participated in this standardization since the 94th MPEG meeting, and significant progress has been made with the various proposals. A test model (TM) has been selected to determine the basic pipeline and key components of visual search. However, the first–version TM has high computational complexity and imperfect retrieval and matching. Core experiments have therefore been set up to improve TM. In this article, we summarize key technologies for visual search and report the progress of MPEG CDVS. We discuss Peking University's efforts in CDVS and also discuss unresolved issues.

Keywords

visual search; mobile; visual descriptors; low bit rate; compression

1 Visual Search

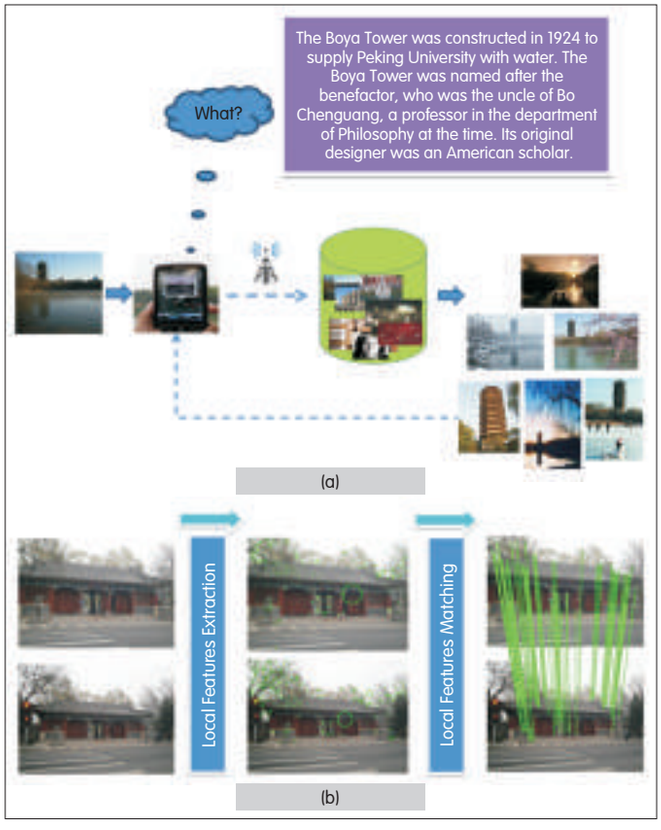
Searching for a specific object in a collection of images has been a long–standing problem with a wide range of applications in location recognition, scene retrieval, product search, and CD and book cover search. Smartphones and tablets have great potential for visual search because they have the integrated functionality of high–resolution embedded cameras, color displays, natural user interfaces, and 3G wireless connections. Most existing mobile visual search systems are deployed in a client–server architecture. The server end has a visual search system in which image representation and inverted index are based on a bag of words (BoW) model. In an online search, a mobile user can take a query photo and send it to a remote server that identifies the query image and its related description (Fig. 1a).

Much research has been done on robust image representation, matching, indexing, retrieval, and geometric re–ranking. Fig. 2 shows a typical flow chart for a visual search. Global features [1], [2] and/or local features [3]–[7] are extracted from database images at a server. The server looks for relevant images based on visual similarity between a query and database images, and inverted indexing speeds up the process (Fig. 2a). Local features are particularly important in a visual search. Even though detecting global features is fast and little memory is required, matching and retrieval based on global features is often less accurate than matching and retrieving based on local features [8]. Recently, Chen et al. showed that a visual search using both global and local features yields better results than using local features only [9].

Using brute force to match query features with database features is infeasible for large–scale databases, and a feature index is needed to improve search efficiency (Fig. 2b) [3], [10]–[12]. Other solutions include approximate nearest–neighbor search, such as *k*–D tree [3], [10], or locality–sensitive hashing [11], [12]. In addition, a feature–space quantization scheme, such as *k*–means, and scalable vocabulary tree [13]–[15] have been widely used for scalable image search, in which two individual features are considered the same word if they fall into the same cluster.

In online search, local features in a query image are

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P60



▲Figure 1. (a) Mobile visual search pipeline. An image database and scalable image index is maintained at the server end. A query photo is transmitted to the remote server to identify reference images and relevant information. (b) Image matching involves feature extraction and feature matching.

extracted and used to search for the local features' nearest neighbors based on a database of reference images. Database images containing nearest neighbors are quickly collected using indexing and are ranked according to a similarity score (Fig. 2c) [13]–[15]. Finally, the top returned images are re–ranked through geometric by taking into account the location of local features (Fig. 2d) [14], [16], [17].

Both academia and industry have made significant progress on visual search, and there is now a growing number of systems or prototypes, including Google Goggles and Nokia Point & Find. To ensure application interoperability, the Moving Picture Experts Group (MPEG) has begun standardizing visual search technologies, in particular, compact descriptors for visual search (CDVS) [18]. CDVS is considered a core technique in augmented reality.

In section 2, we discuss MPEG's progress on CVDS and Peking University's (PKU's) responses to the MPEG CDVS call for proposals (CfP). We also describe the setups of PKU's core experiments. In section 3, we discuss compact descriptors and related issues such as indexing, retrieval, and geometric re–ranking. In section 4, we discuss open problems and challenges. Section 5 concludes the paper.

2 MPEG Compact Descriptors for Visual Search

2.1 Introduction to CDVS

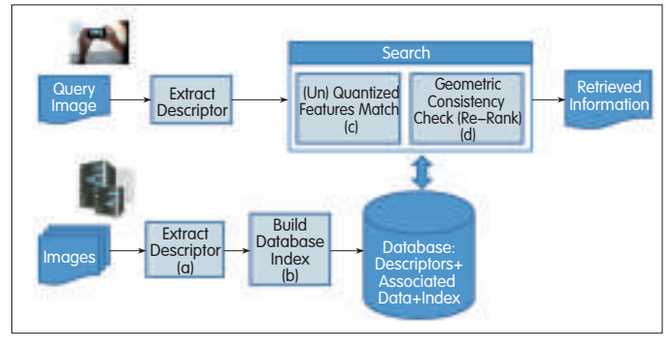
Academia and industry have made progress on key components for visual search; however, several issues still remain [2]–[4], [13]–[15]. It is not clear, for example, how to make visual search applications compatible across a broad range of devices and platforms. This is one of the motivations for MPEG CDVS standardization, which was discussed during the first and second Workshops on Mobile Visual Search hosted by Stanford University [19] and PKU [20], respectively. These workshops led to a formal request being made to MPEG for CDVS standardization and a fleshing out of CDVS requirements.

At the 92nd to 96th MPEG meetings, CDVS experts investigated potential applications, scope of standardization, requirements, and evaluation framework [21]–[24]. At the 97th MPEG meeting, the CfP was issued [25]. To ensure interoperability, the CDVS standard aims to define the format of compact visual descriptors as well as the pipeline for feature extraction and search process [22].

The visual descriptors need to be robust, compact, and easy to compute on a wide range of platforms. High matching accuracy must be achieved at least for images of rigid, textured objects; landmarks; and documents. Matching should also be accurate despite partial occlusions and changes in vantage point, camera parameters, and lighting. To reduce the amount of information transferred from the client end to the server end, and to alleviate query transmission latency, the descriptor length must be minimized. Descriptor extraction must allow adaptation of descriptor length so that the required performance level can be satisfied for different database. Extracting descriptors must not be complex in terms of memory and time so that they are deployable in most common platforms.

2.2 Evaluation Framework

Here, we summarize the CDVS evaluation framework [26],



▲Figure 2. Visual search pipeline in a client–server architecture, and basic components of feature (a) extraction, (b) indexing, (c) matching, and (d) re–ranking.

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P61

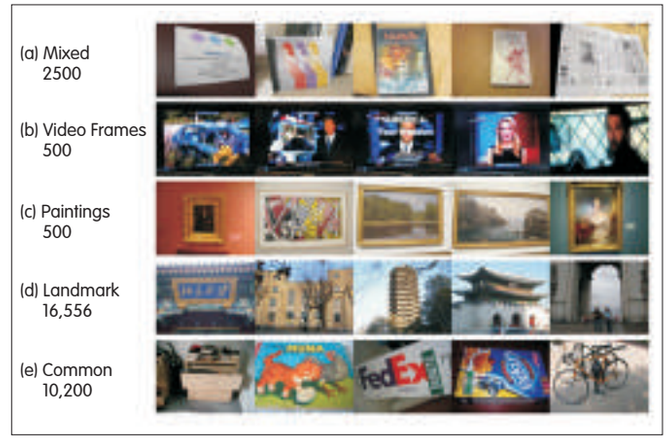
which closely aligns with CDVS requirements [22]. Different proposals is evaluated using two types of experiments: retrieval and pairwise matching. The retrieval experiment is done to determine descriptor performance during image retrieval. Mean average precision (mAP) and success rate for top matches are the evaluation criteria. The pairwise matching experiment is done to determine the performance of descriptors during the matching of image pairs. The localization precision is assessed according to the ground–truth of localization annotations. Performance is measured by the success rate at a given false alarm rate (for example 1%), and localization precision.

Each proposal sent to MPEG needed to show the descriptor's scalability, and this is done by reporting the performances at six operating points of different query sizes: 512 bytes, 1 kB, 2 kB, 4 kB, 8 kB, and 16 kB. Interoperability of the descriptors is also evaluated. A descriptor generated at any of these six operating points should allow matching with descriptors generated at any other operating point. For each proposal, an experiment is conducted to evaluate pairwise matching of 1 kB descriptor versus 4 kB descriptor, and 2 kB descriptor versus 4 kB descriptor. To reduce time–related complexity, feature extraction must not be longer than two seconds, pairwise matching must not be longer than one second, and retrieval must not be longer than five seconds in a single thread the reference platform (Dell Precision workstation 7400–E5440).

Eight datasets are collected in the CDVS evaluation framework. These datasets are ZuBud, UKBench, Stanford, ETRI, PKU, Telecom Italia (TI), Telecom SudParis, and Huawei. There are 30,256 images categorized as mixed text and graphics, paintings, frames captured from video clips, landmarks, and common objects. Fig. 3 shows these categories and the number of reference images. Each dataset provides ground–truth annotation of pairs of matching and non–matching images as well as ground–truth annotation of query images and corresponding reference images. For localization, the mixed text and graphics category provides bounding boxes for each matching pair. In the retrieval experiment, the discriminability of a descriptor is assessed using a distractor image set containing one million images of varying resolutions and content (collected from Flickr).

2.3 Progress on CDVS and Evidence from CDVS Research

Remarkable progress has been made at MPEG CDVS Ad–Hoc Group meetings. At the 94th MPEG meeting, PKU proposed an extremely compact descriptor based on a bag–of–features histogram rather than features. This compact descriptor provides a much higher compression rate without serious loss of discriminability [27]–[30]. To determine the lowest operating point for a promising visual search, geo–tag (a kind of side information) is used to produce very compact descriptors for visual searches of landmarks [29],



▲Figure 3. Image categories and the number of reference images in the corresponding CDVS evaluation dataset.

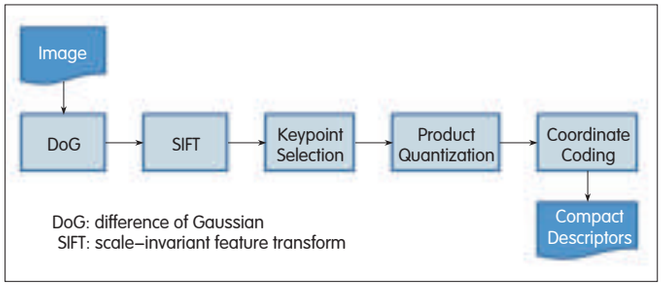
[30]. A photo collection of landmarks is first segmented according to discrete geographical regions using a Gaussian mixture model. Then, ranking–sensitive vocabulary boosting is done to create a very compact codebook within each region. When entering a new geographical region, codebooks in the mobile device are downstream–adapted, which ensures compactness and discriminative power. This technique is called location–discriminative vocabulary coding (LDVC). With only hundreds of bits per query image, LDVC performs descriptor coding over the bag of scale–invariant feature transform (SIFT) features [29]. Referring to such a small number of bits, the CDVS Ad–hoc Group has determined that 512 bytes, is the lowest point (including space for coordinate information).

Beyond landmark search, visual statistics and rich contextual cues at the mobile end are exploited over the reference image database to determine multiple coding channels [28]. A compression function is determined for each channel. A query is first described in a high–dimensional visual signature that is mapped to one or more channels for further compression. The compression function within each channel is determined according to a robust principle component analysis (PCA) scheme, and the retrieval ranking capability of the original signature is maintained. With hundreds of bits, the resulting descriptor has search accuracy that is comparable to that of the original SIFT features [28]. In summary, PKU's research has shown that extremely low bit–rate descriptors (128 or 256 bits per query image[1]) are possible using quantization and by establishing a small but discriminative codebook.

At the 98th MPEG meeting, proponents of CDVS submitted 11 proposals for compact descriptors. These proposals broadly fall into two categories. In the first category, compact descriptors are produced by compressing local descriptors, for example SIFT descriptors, in a data–driven or data–independent way. Raw descriptors are first extracted using existing techniques. Then, raw descriptors are compressed to produce a compact descriptor. Proposals from TI [26], PKU [31], [32], and Stanford & Aptina [33] fall

[1] Extra bits for coding locations are not counted as part of the 256 or 128 bits. In the current evaluation framework [39], each operating point refers to the length of overall bit streams, including visual descriptors, locations, and other information.

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P62



▲Figure 4. Key components in CDVS test model under consideration.

into this category. In the second category, distinct compact descriptors are produced at the first stage of raw–descriptor extraction; there is no additional and separate compression stage. NEC's proposal belongs to this class [34].

Both TI and PKU proposed quantizing raw SIFT descriptors with product quantization in which quantization tables are pre–learned on a training dataset [35], [36]. The difference between the proposals from TI and PKU lies in how the raw segment of a SIFT descriptor is subdivided. PKU divides a SIFT descriptor into more segments, and each segment is quantized with a smaller codebook. In contrast, TI divides a SIFT descriptor into fewer segments, and each segment is quantized by a large codebook. TI's solution has greater matching and retrieval accuracy than PKU's; however, PKU's solution has greater localization accuracy than TI's. Moreover, TI's quantization table may take up 500 MB, which is much larger than PKU's at approximately 10 MB.

Stanford & Aptina [33] use type coding to compress SIFT–like descriptors. Compared with the product quantization of TI and PKU, type coding is data–independent and requires none of pre–learned quantization tables. Data–independent methods usually perform worse than data–driven methods, especially at lower operating points such as 512 bytes and 1 KB.

NEC [34] proposed a binary descriptor created by binarizing the histogram of quantized gradient orientations. The pairwise matching and localization of NEC's binary descriptor outperforms that of TI and PKU. However, retrieval is much worse than in other proposals at almost all operating points and datasets. Significant degeneration in retrieval can be attributed to severe loss of discriminative information when there is a huge number of distractors.

After the CDVS CfP issued at the Torino meeting, a test model under consideration (TMuC) based on product quantization was selected at the 98th MPEG meeting in Geneve [37]. Crucial stages such as local descriptor extraction, feature selection, compression, and location coding have been identified in the TMuC [37]. Fig. 4 shows the modules and dataflow in the test model. The first two blocks extract raw local descriptors, including a multiscale key–point detector based on difference of Gaussian (DoG) and a feature descriptor SIFT based on gradient–orientation histogram. Key–point selection filters in a subset of important key points fulfill the compactness and scalability requirements. Product quantization compresses a SIFT descriptor in a data–driven way, and coordinate coding compresses key point locations. In the TMuC, SIFT is used as local descriptors because it has excellent scale and rotation invariance. More than 1000 key points could be detected in a 640 × 480 VGA image. To create a very compact descriptor, a subset of useful key points is kept. The TMuC has shown that key–point selection is critical to maintain search accuracy at lower operating points because noise points can be filtered out without loss of performance. The descriptor at each key point is product quantized with pre–learnt tables. Descriptor length can be significantly reduced; for example, in PKU's proposal, more than 85% of bits are saved. Feature locations are finally compressed by quantization and context arithmetic coding.

Several issues remain open in the first version of the TMuC. First, quantization tables may consume up to 500 MB of memory, which is infeasible for most mobile phones. Second, the current TMuC is built on DoG–based interest–point detection and raw SIFT descriptor. However, both these techniques are patented [38]. UBC has granted both non–exclusive and exclusive field–of–use licenses to multiple companies. If UBC has already granted exclusive rights to commercialize SIFT in mobiles (where most likely MPEG CDVS would be used), the patents would prevent widespread use of the standard and stymie competition. Fortunately, experts have shown that the current TMuC can be enhanced by replacing individual components with new technologies such as alternative interest–point detector and raw local descriptor.

2.4 Core Experiments

At the 99th MPEG meeting, six core experiments [39] were set up to investigate

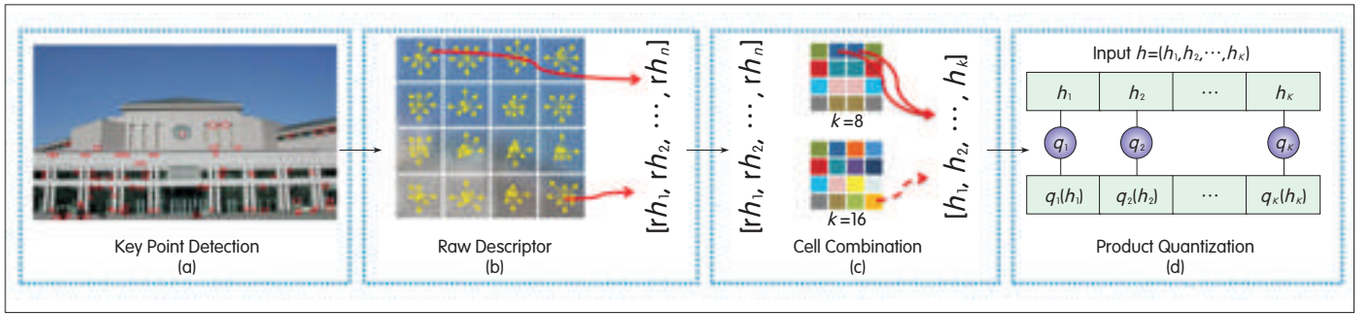
- the effects of combining local descriptors with a global descriptor
- low memory solutions for descriptor compression
- improved feature–location compression efficiency
- key–point detection techniques to bypass intellectual property issues
- better uncompressed (raw) local descriptors
- more efficient retrieval pipeline.

Improvements in memory use, time complexity, search accuracy, and other aspects were verified at the 100th MPEG

▼Table 1. Timeline and milestones of MPEG CDVS standardization

Meeting	Date	Milestone
97th	Jul. 2011	Call for Proposals
98th	Dec. 2011	Proposals Evaluated, Test Model Determined
99th	Feb. 2012	Six Core Experiments Set up
100th	Apr. 2012	Evaluation of Proposals
101st	Jul. 2012	First Working Draft
103rd	Jan. 2013	Committee Draft
105th	Jul. 2013	Draft of International Standard
107th	Feb. 2014	Final Draft of International Standard

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P63



▲Figure 5. Product quantizing raw SIFT descriptors in a data-driven way.

meeting. Table 1 shows the timeline and milestones of MPEG CDVS.

3 Compact Descriptors

To minimize the amount of information to be transferred, compact descriptors are desirable. Descriptor compactness relates to the number of local features and the size of a local feature. Typically, each local feature consists of visual descriptor and its location coordinates. Hence, descriptor size is determined by the compression rates of both visual descriptors and locations. Key-point selection is necessary to figure out a subset of important key points (Fig. 4). Product quantization minimizes the length of local descriptors, and coordinate coding minimizes the length of location codes. In the following subsections, we introduce techniques for local descriptor compression, location coding, and key-point selection. We also discuss PKU's compact descriptors.

3.1 Local Descriptor Compression

Feature extraction starts by selecting key points in an image. State-of-the-art visual search algorithms are built on a DoG detector followed by SIFT description [3]. Such algorithms were used (sometimes in slightly modified form) in the CDVS proposals. To achieve low-bit-rate visual search, SIFT descriptors and other popular descriptors, such as sped-up robust features (SURF) [4] and PCA-SIFT [40], are oversized. The size of hundreds of these local descriptors is often greater than the original image size. The objective of descriptor compression is to significantly reduce the descriptor size without deteriorating visual discriminative power.

In general, there are two groups of algorithms for compressing local descriptors. Algorithms in the first group quantize raw descriptors in a data-independent way. Chandrasekhar et al. proposed a descriptor called compressed histogram of gradients (CHoG), which uses Huffman Tree, Gagie Tree [41], or type coding [42] to compress a local feature into approximate 50 bits. Type coding involves constructing a lattice of distributions (types), given by

$$Q = Q(K_1, K_2, \ldots, K_m), \quad (1)$$

where K_i is the number of points in each lattice of a predefined 2-D gradient partition. Thus, the probability of these lattice distributions is

$$q_i = \frac{k_i}{n},\ k_i,\ n \in Z_+,\ \Sigma_i k_i = n, \quad (2)$$

where n is the number of points within the gradient partition.

The index of the type closest to the original distribution is then selected and transmitted [42]. Prior to type coding, descriptors are L1-normalized so that the descriptor can be dealt with as a probability distribution.

Algorithms in the second group work are data-driven. A codebook to partition the descriptor space is first determined from training data. Quantized descriptors are represented by the nearest code word [13], [26], [28]–[32],[35]. PKU's proposal [31], [32], [35] and TI's proposal [26] fall into this category. Fig. 5 shows PKU's descriptor compression using product quantizing. Generally speaking, a data-driven approach may result in better pairwise matching and retrieval than a data-independent approach, but storing a codebook consumes more memory.

3.1.1 PKU's Proposal for Product Quantizing SIFT Descriptors

In product quantization, an input vector is divided into k segments, and those segments are independently quantized using k subquantizers [31]. Each compressed descriptor is thus represented by k indexes comprising the nearest code words of k segments. A less-complex quantizer q_j is associated with the jth segment of the input vector. Product quantization is used to compress raw descriptors, given by

$$h = (h_1 = (h_1^1, \ldots, h_1^m), \cdots, h_k = (h_k^1, \ldots, h_k^m)) \quad (3)$$

where $h_j = (h_j^1, \ldots, h_j^m)$ is the histogram of the ith subsegment of length m.

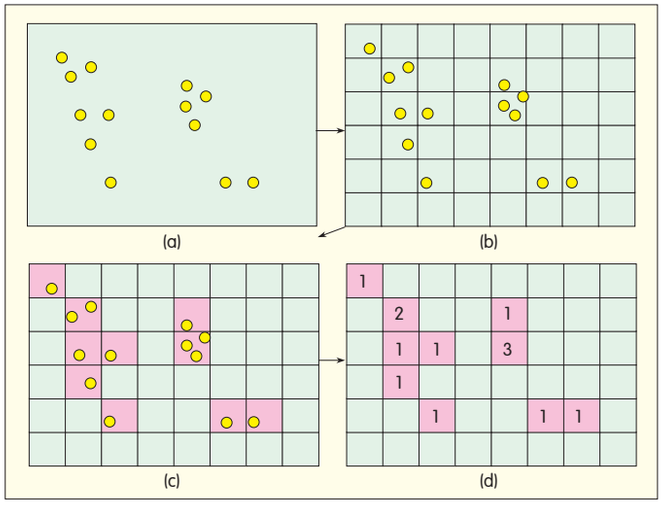
The descriptor dimension is $n = m \times k$. h is structured into k parts, each having m dimensions. The vectors are independently quantized from k parts using k subquantizers. The size z of subquantizer dictionaries is the same. The Cartesian representation space is given by z^k. Each subquantizer q maps an m-dimensional vector h_j to

$$q(h_j) \in C = \{c_i,\ c_i \in R^m,\ 1 \leqslant i \leqslant z\}, \quad (4)$$

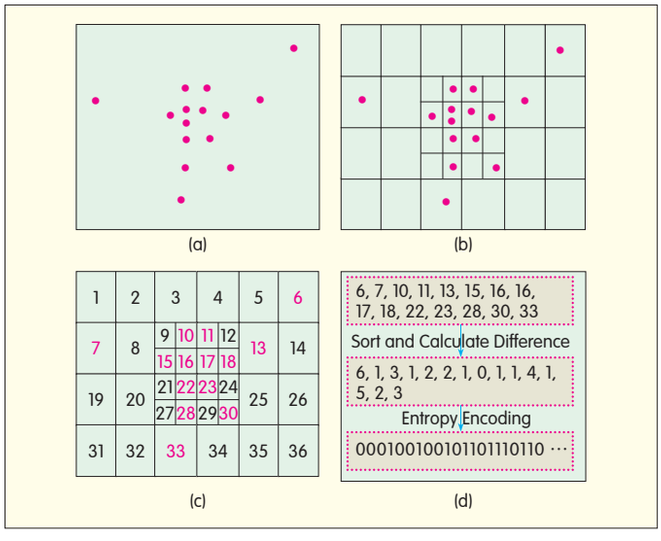
where C is a set of reproduction values c_j to represent original vector h_j with dimension m. The size of the dictionaries is z.

Quantizer quality is measured by the mean squared error

D:\EMAG\2011-02-29/VOL9\CONTETN.VFT——1PPS/P64



▲Figure 6. Coding location coordinates. (a) Spatial distribution of local features, (b) coordinate space uniformly quantized with a 2D grid, (c) histogram map indicating emptiness\non–emptiness of each block, and (d) histogram counting the number of features that fall into each non–empty block.



▲Figure 7. Coding location coordinates according to an importance map. (a) Spatial distribution of local features. (b) The coordinate space is divided into blocks of different sizes, and the divisions are optimized by statistical learning. (c) Each block is assigned a unique number to identify different locations. (d) Block identifiers are sorted and are followed by difference–based entropy encoding.

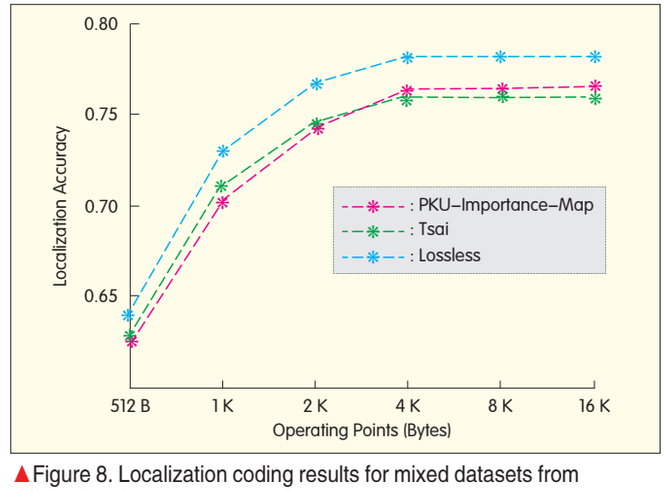
(MSE) between an input vector and its reproduction value. Subquantizers are greedy learned by minimizing MSE using Lloyd's algorithm. Consequently, the compressed descriptor is represented by a short code comprising the indexes in all the subquantizers. However, the descriptor compression in the TMuC is imperfect because product quantization requires large memory to store several tables. The size of quantization tables may reach up to 500 MB, which is unacceptable in mobile platforms. Reducing the size of quantization tables has been set up as a core experiment [39].

3.2 Location Coordinate Coding

Each local feature comprises a visual descriptor and location coordinates. A visual descriptor is used to compute visual similarity, and the location part is for geometric verification when re–ranking returned images. Location compression is important in low bit–rate visual search. Take, for example, the lowest operating point of 512 bytes per query in the CDVS evaluation framework. For a 640 × 480 VGA image, 20 bits are needed to encode a feature's location without compression. For an image comprising 500 local features, it would take about 1250 bytes to code location coordinates, which is more than 512 bytes

Tsai et al. proposed lossy compression of feature locations [43]. Location coordinates are quantized with a uniform partition of the 2D location space prior to coding (Fig. 6b). The quantized location is converted into a location histogram comprising histogram map (Fig. 6c) and histogram count (Fig. 6d). The histogram map comprises empty and non–empty blocks, and the histogram count is the number of features in each non–empty block. Different block sizes lead to different quantization levels. The histogram map and histogram count are coded separately with context–based arithmetic coding. This scheme reduces bits by a factor of 2.8 compared with fixed–point representation. When run length, quad–tree, and context–based algorithms are compared, context–based coding performs the best, and gain decreases as block size becomes smaller. Run–length coding for a larger block size does not provide any gain.

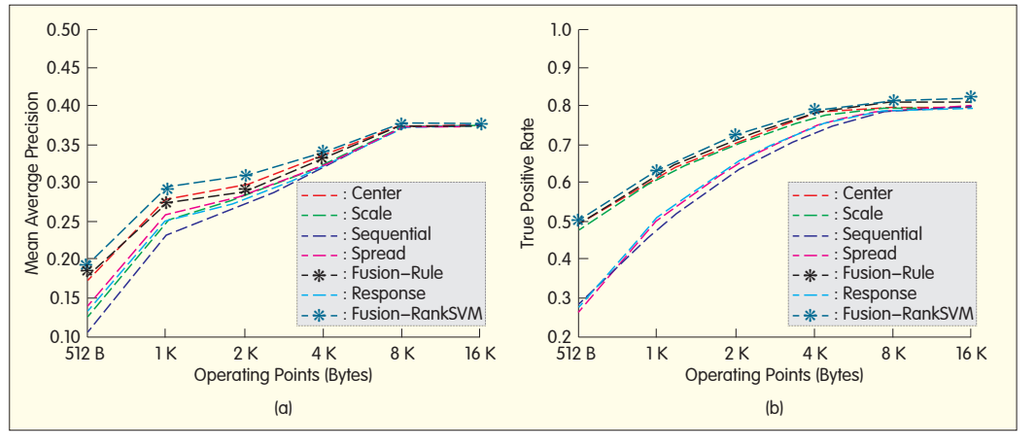
Tsai's location coding algorithm treats all features equally; however, in real–world applications, a subset of features could be more important for robustness and information fidelity. Compression distortion of important features can be reduced at the risk of increasing distortion for unimportant features. Therefore, an importance map can be placed over an image. Important features are quantized to finer levels, and unimportant features are quantized to coarser levels (Fig. 7). The definition of an importance map depends on applications.



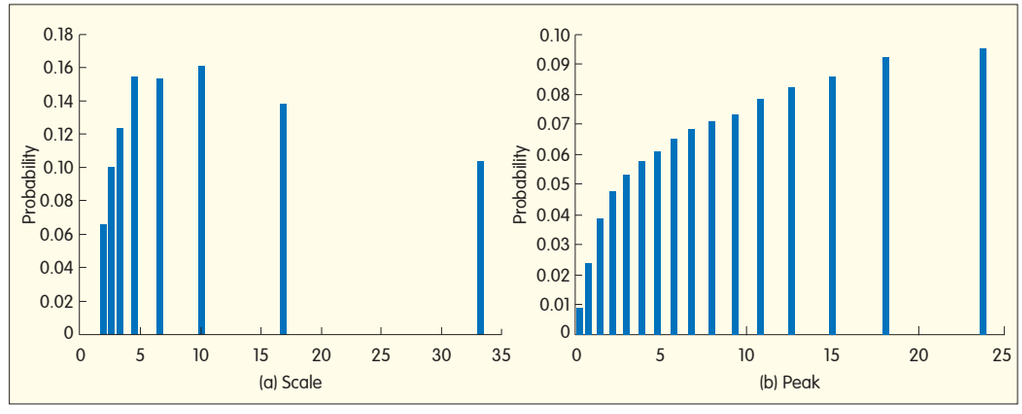
▲Figure 8. Localization coding results for mixed datasets from different methods.

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P65

In our experiments, an importance map is a Gaussian function so that key points near the center of the image are emphasized. This definition is validated in subsequent key–point selection experiments. These experiments show that, compared with Tsai's approach, the importance map approach can reduce bits by 15% without seriously affecting performance (Fig. 8) [44]. Location coding was one of the core experiments at the 99th MPEG meeting.



▲Figure 9. (a) Retrieval and (b) matching results with different key–point selection algorithms.



▲Figure 10. Statistics that are helpful for feature selection: (a) Probability of a correct match as a function of scale. The scale is quantized into eight levels. (b) The probability of a correct match as a function of peak value.

3.3 Key–Point Selection

Selection criteria may include a key point's location, peak value of DoG response, scale, dominant orientation, or descriptors. We conducted a systematic study [45], and Fig. 9 shows some results of feature–point selection using the following approaches:

- Sequential selection. This involves selecting key points continuously from top left to bottom right.
- Spread selection. This involves selecting key points to maximize the spread of the points.
- Center selection. This involves selecting key points near the image center.
- Scale selection or peak selection. This involves selecting key points of larger scales or peak values.
- Fusion selection. This involves using a supervised approach to combine factors for better performance. Key points can be repeatedly detected and matched and are regarded as stable. The stability score of a key point is the number of images containing correctly matched points. A RankSVM[2] ranking model is trained to sort key points according to their stability scores [46].

For operating points from 512 bytes to 16 kB, we set the maximum number of key points to 200, 350, 450, 550, and 1000. If the number of key points in an image exceeds the limit, this number needs to be cut down to meet the limit. Four hundred images, including 100 objects from UKBench dataset, were used to generate features for RankSVM training.

The fusion selection approach performs the best, but center selection is also effective. In matching experiments, center selection and scale selection have about 10–30% gain at 512 bytes and outperform sequential selection. The gain is less significant when the operating point goes up to 4 kB because there is no need to cut down the number of key points. The performance of peak selection is comparable to that of spread selection, with TP improved by about 5–10%. Sequential selection performs the worst. Fusion based on RankSVM brings about slight gain, and similar results have been shown in retrieval experiments. However, scale selection performs much worse than center selection.

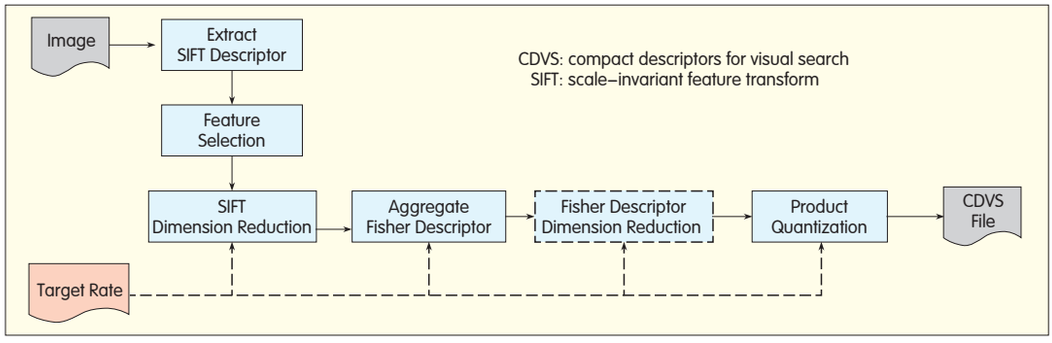
In summary, key–point selection significantly contributes to pairwise matching and retrieval, especially at very low operating points. Center selection basically works well; however, when the query object is far from the center, this approach is poor. A solution is to combine different factors.

In our model, each key point is characterized by scale,

[2] RankSVM addresses a ranking problem by formulating a classification problem. A binary classifier is learnt in order to determine which key point is better. In training, the goal of optimization is to minimize the average number of inversions on training data. With the learnt model, we obtain a relative order for any pair of key points. All key points are sorted according to minimum number of inversions (in terms of the predicted relative orders).

In our scenario, each set of training data contains a sequence of images of the same object, which may be shot at different viewpoints or scales. Pairwise matching is applied, and we count how many images are correctly matched with each feature. We use this number to estimate the feature's importance; the larger the number, the more important the feature. A feature's scale, distance to center, and peak value are concatenated to form a vector to train RankSVM.

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P66



◀Figure 11. Global feature extraction pipeline.

dominant orientation, peak value of DoG, and distance to center. Using the training model, for each individual factor, we estimate the probability that a feature can be matched correctly to features in other images. Fig. 10 shows these estimations based on scale and peak values. The probability of a correct match increases as the scale increases within a certain range, and this is consistent with our hand–crafted rule. The probability decreases as the scale exceeds the limit, and this is not taken into account in our current model.

Other models are fused with a naive Bayesian approach; that is, probabilities for all factors are multiplied to produce a score. Key points of higher scores are selected.

Instead of hand–crafted rules, as in [45], Gianluca et al. [26] exploit the difference statistics for correctly and incorrectly matched key points. The difference statistics are consistent across various large datasets [26]. Training data contains images from the Pasadena buildings dataset, INRIA holidays dataset, and additional images of food labels and Italian buildings. Training data also contains video frames from movies. The training set is organized into a list of pairwise matching images. Each image undergoes key–point detection, matching, and geometric consistency check. The actual (true or false) matched pairs of features are grouped as inliers or outliers. Inliers are key points that have been correctly matched, and outliers are key points that have not been matched or have been incorrectly matched. Each feature is labeled according to whether the feature has been matched correctly (value 1) or not (value 0).

Key–point selection is important; however, more in–depth work is needed on how to make optimal combinations of factors. Key–point selection might be a core experiment in the upcoming 100th MPEG meeting.

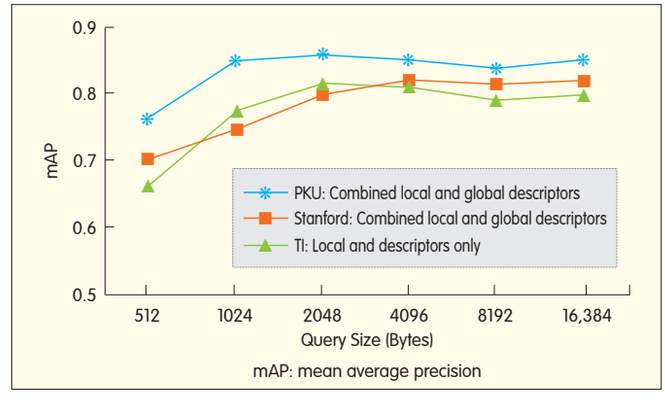
3.4 Global Descriptors

State–of–the–art visual search applications usually use a BoW model based on local features. Chen et al. showed that visual search accuracy can be improved by combining global and local features [9]. A global feature is generated from SIFT local descriptors. Some 190 centroids are generated by applying *k*–means clustering to SIFT descriptors. Given an image, each SIFT descriptor is quantized to its nearest centroid. The residual between the descriptor and the centroid is computed. For each centroid, the mean residual between the centroid and all quantized descriptors falling to this centroid is computed. To reduce the dimensions of residuals, linear discriminative analysis (LDA) is applied, and 32 of the most discriminative LDA eigenvectors are retained. The transformed values are binarized by the sign. Experiments show that incorporating global descriptors may improve performance, especially at lower operating points.

PKU has provided evidence that visual search accuracy can be improved by leveraging global descriptors [47]. A global descriptor is generated by aggregating SIFT descriptors (Fig. 11). For a given image, a subset of SIFT key points is selected, and each 128–D SIFT descriptor is reduced to 64–D eigenvectors by applying PCA. A Gaussian mixture model (GMM) with 256 Gaussians is trained using an independent dataset. To form a global descriptor, an image is represented as a fixed–size Fisher vector generated from the gradient of the probability of the selected 64–D SIFT descriptors over the mean of the learned GMM. For each Fisher vector, PCA dimensions are reduced, and products are quantized to encode the global feature. When a CDVS evaluation framework is used, experiment results show that applying global descriptors to the retrieval step and then applying local descriptors to the re–ranking step greatly improves performance at all operating points (Fig. 12) [47].

4 Other Issues

Feature indexing, retrieval, and geometric re–ranking may not be included in the MEPG CDVS standard. Proper handling of these issues will greatly improve retrieval scalability and



▲Figure 12. Retrieval performance using combined local and global descriptors , and local descriptors only on a mixed dataset with 1 M distracters.

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P67

accuracy. For fast search, database features must be indexed [3], [10]–[12]. There are two kinds of indexing approaches. The first approach involves attempting to search for approximate nearest neighbors, for example, *k*–D tree [3], [10] and locality–sensitive hashing (LSH) [11], [12]. The second approach involves a BoW model [13]–[15], and greater speed is achieved by quantizing feature space. *k*–D tree methods outperform BoW methods in terms of run time and recognition accuracy. However, BoW methods require much less storage space than *k*–D tree and LSH methods because inverted indexing based on BoW does not require feature vectors to be stored. LSH methods generally result in better search accuracy than *k*–D tree methods but are inferior in terms of search time (which increases sharply as the database size increase). In the CDVS evaluation framework, memory use is limited to 16 GB. Neither *k*–D tree nor LSH is able to load the descriptors of all reference images (including one million distractors) into 16 GB memory. The current TMuC uses the BoW model.

In online search, local features in a query image are used to perform nearest–neighbor search. Database images are ranked by a scoring function that represents their visual similarity to nearest neighbors [13]–[15]. Post–processing can be used to verify geometric consistency within a subset of top–ranked images [14], [16], [17], and subset images are re–ranked accordingly. Various re–ranking algorithms have been proposed for trading–off between accuracy and time complexity. In [14], spatial verification is used to estimate a transformation between a query region and each target image. Returned images are re–ranked according to discriminability of spatially verified visual words (inliers). In [16], Jegou et al. propose using angle and scale to adjust the similarity score of an image. The score decreases when a visual word is inconsistent with transformation and increases in the presence of consistently transformed points. Compared with re–ranking in [14], re–ranking in [16] is much faster, but there is a risk of higher false positive rate.

In the TMuC, the fast re–ranking algorithm in [17] is slightly modified. Feature pairs are formed by comparing descriptor classification paths in a scalable vocabulary tree, and a geometric verification score is computed for these pairs. A histogram of logarithmic distance ratios (LDRs) for pairs is used in the TMuC. Distribution of LDRs for pairs of inliers is different from that of LDRs of pairs of outliers, which are used to rate geometric consistency.

5 Conclusion

In this paper, we have discussed developments in visual search technologies. These developments include state–of–the–art, low bit rate visual search pipeline, and important components such as compact descriptors and efficient image–matching and retrieval algorithms. To facilitate interoperability between visual search applications, MPEG has made great efforts to standardize compact descriptors for visual search. We have reported a CDVS evaluation framework, which is a competitive and collaborative platform for evaluating visual search technologies and solutions and is a benchmark for crucial modules.

Despite significant progress on visual search within academia and industry, a few challenges remain. The size of visual queries transmitted from a mobile device to servers needs to be minimized because of the bandwidth constraint of (3G) wireless networks. Under the CDVS umbrella, descriptor quantization [26], [27], [31], [32], [37], [41], [42], location coding [43], [44] and key–point selection [37], [45] have been attempted in order to produce more compact descriptors. Approximately 90% of bits can be saved using compact descriptors as opposed to using uncompressed SIFT descriptors, and search performance is well maintained. In addition, a mobile device is constrained by battery life, so energy saving is crucial for feature extraction and compression. Because mobile phones usually have limited computing capability, operations on a mobile phone must not be too complex. A core experiment that has been set up to reduce quantization table size to less than 5 MB. Moreover, computing DoG is also time–consuming. Aptina [48] provided fast algorithms for key–point detection and descriptor generation, and these algorithms are targeted at system–on–chip (SoC) implementation. However, the robustness of a detector still needs to be evaluated. More importantly, the ongoing MPEG CDVS standardization has attracted much interest from hardware manufacturers such as Aptina, Nvidia, and STMicroelectronics.

Acknowledgments:

We would like to thank the reviewers for their useful comments. This work was supported by National Basic Research ("973") Program of China (2009CB320902), and in part by the Chinese National Nature Science Foundation (60902057).

References

[1] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. Conf. Vision and Pattern Recognitioni (CVPR)*, San Diego, CA, 2005, vol. 1, pp. 886–893.
[2] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," in *Int. J. Comput. Vision*, vol. 42, no. 3, pp. 145–175, 2001.
[3] D.G. Lowe, "Distinctive image features from scale–invariant keypoints," in *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.
[4] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: speeded up robust features," in *J. Comput. Vision Image Understanding*, vol. 110, no. 3, pp. 346–359, Jun. 2006.
[5] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *Proc. 7th European Conf. Comput. Vision (ECCV'02)*, Copenhagen, 2002, pp. 128–142.
[6] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *Proc. British Machine Vision Conf.*, Cardiff, Wales, 2002, pp. 384–393.
[7] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.
[8] M. Douze, H. Jégou, H. Sandhawalia, L. Amsaleg, and C. Schmid, "Evaluation of gist descriptors for web–scale image search," in *Proc. Int. Conf. Image and Video Retrieval (CIVR'09)*, Santorini, Greece, July 2009.
[9] D. Chen, V. Chandrasekhar et al., MPEG, "Improvements to the Test Model Under Consideration with a Global Descriptor", ISO/IEC JTC1/SC29/WG11/ M23578, 2012/02.
[10] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu, "An optimal algorithm for approximate nearest neighbor searching fixed dimensions," in *J. of the ACM*, vol. 45, no. 6, pp. 891–923, Nov. 1998.
[11] A. Andoni and P. Indyk, "Near–optimal hashing algorithms for approximate nearest neighbor in high dimensions," in *Commun. of the ACM*, vol. 51, no. 1, pp.

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P68

117–122, 2008.
[12] Y. Ke, R. Sukthankar, and L. Huston, "An efficient parts–based near–duplicate and sub–image retrieval system," in *Proc. 12th Annu. ACM Int. Conf. Multimedia*, New York, NY, 2004, pp. 869–876.
[13] J. Sivic and A. Zisserman, "Video Google: a text retrieval approach to object matching in videos," in *Proc. 9th IEEE Int. Conf. on Comput, Vision*, Nice, France, 2003, pp. 1470.
[14] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. IEEE Conf. Comput Vision and Pattern Recognition,* Minneapolis, MN, 2007, pp. 1–8.
[15] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recognition*, New York, NY, Jun. 2006, vol. 2, pp. 2161–2168.
[16] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *Proc. 10th European Conf. Comput. Vision: Part I*, Marseille, 2008, pp. 304–317.
[17] S. S. Tsai, D. Chen, G. Takacs, V. Chandrasekhar, R. Vedantham, R. Grzeszczuk, and B. Girod, "Fast geometric re–ranking for image–based retrieval," in *Proc. 17th IEEE Int. Conf. Image Processing*, Hong Kong, 2010, pp. 1029–1032.
[18] MPEG, "Call for Proposals for Compact Descriptors for Visual Search," ISO/IEC JTC1/SC29/WG11 N12201, 2011/07.
[19] Standford Center for Image Systems Engineering. Workshop on mobile visual search (2009) [Online]. Available: http://scien.stanford.edu/pages/conferences/mvs/
[20] Peking University. The 2nd workshop on mobile visual search (2011) [Online]. Available: http://idm.pku.edu.cn/mvs/talksabstract.asp
[21] MPEG, "Compact Descriptors for Visual Search: Evaluation Framework," ISO/IEC JTC1/SC29/WG11/N12202, 2011/07.
[22] MPEG, "Compact Descriptors for Visual Search: Requirements," ISO/IEC JTC1/SC29/WG11/N11531, 2010/07.
[23] MPEG, "Compact Descriptors for Visual Search: Context and Objectives," ISO/IEC JTC1/SC29/WG11/N11530, 2010/07.
[24] MPEG, "Compact Descriptors for Visual Search: Applications and Use Scenarios," ISO/IEC JTC1/SC29/WG11/N11529, 2010/07.
[25] MPEG, "Call for Proposals for Compact Descriptors for Visual Search," ISO/IEC JTC1/SC29/WG11 N12201, 2011/07.
[26] MPEG, "Telecom Italia's response to the MPEG CfP for Compact Descriptors for Visual Search," ISO/IEC JTC1/SC29/WG11/ M22672, 2011/11.
[27] MPEG, "A Case Study for Compact Descriptors for Visual Search – Location Discriminative Mobile Landmark Search," ISO/IEC JTC1/SC29/WG11/ M18542, 2010/10.
[28] R. Ji, L. Y. Duan, J. Chen, H. Yao, Y. Rui, S. F. Chang, and W. Gao, "Towards low bit rate mobile visual search with multiple–channel coding," in *Proc. 19th ACM Int. Conf. on Multimedia*, Scottsdale, AZ, 2011, pp. 573–582.
[29] R. Ji, L. Y. Duan, J. Chen, H. Yao, T. Huang, and W. Gao, "Learning Compact Visual Descriptor for Low Bit Rate Mobile Landmark Search," in *Proc. Int. Joint Conf. on Artificial Intelligence*, Barcelona, 2011, pp. 2456–2463.
[30] R. Ji, L. Y. Duan, J. Chen, H. Yao, J. Yuan, Y. Rui, and W. Gao. "Location Discriminative Vocabulary Coding for Mobile Landmark Search," in *Int. J. Comput. Vision*, vol. 96, no. 3, pp. 290–314, 2012.
[31] MPEG, "Peking Compact Descriptor–PQ–SIFT," ISO/IEC JTC1/SC29/WG11/ M22620, 2011/11.
[32] MPEG, "Peking compact descriptor–PQ–WGLOH," ISO/IEC JTC1/SC29/WG11/ M22619, 2011/11.
[33] MPEG, "CDVS Proposal: Stanford Nokia Aptina Features," ISO/IEC JTC1/SC29/WG11/ M22554, 2011/11.
[34] MPEG, "NEC´s Response to CfP for Compact Descriptor for Visual Search" ISO/IEC JTC1/SC29/WG11/ M22717, 2011/11.
[35] Chunyu Wang, Ling–Yu Duan, Yi–Zhou Wang, and Wen Gao, "PQ–WGLOH: A bit–rate scalable local feature descriptor," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Kyoto, Mar. 2012.
[36] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," in *IEEE Trans. Pattern Analysis & Machine Intelligence*, vol. 33, no. 1, pp. 117–128, Jan. 2011.
[37] MPEG, "Description of Test Model under Consideration for CDVS," ISO/IEC JTC1/SC29/WG11 W12367, 2011/12.
[38] David G. Lowe, "Method and apparatus for identifying scale invariant features in an image and use of same for locating an object in an image," US Patent 6 711 293, March 23, 2004.
[39] MPEG, "Description of Core Experiments on Compact descriptors for Visual Search," ISO/IEC JTC1/SC29/WG11/ N12551, 2012/02.
[40] Y. Ke and R. Sukthankar, "PCA–SIFT: A more distinctive representation for local image descriptors," in *Proc. IEEE Comput. Soc. Conf. Computer Vision and Pattern Recognition*, Washington DC, 2004, vol. 2, pp. 506–513.
[41] V. Chandrasekhar, G. Takacs, D. Chen, S. Tsai, R. Grzeszczuk, and B. Girod, "CHoG: compressed histogram of gradients—a low bit–rate descriptor," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Miami, FL, 2009, pp. 2504–2511.
[42] V. Chandrasekhar, G. Takacs, D. M. Chen, S. S. Tsai, Y. Reznik, R. Grzeszczuk, and B. Girod, "Compressed histogram of gradients: a low bitrate descriptor," in *Int. J. Comput. Vision*, vol. 96, no. 3, pp. 384–399, 2011.
[43] S. S. Tsai, D. Chen, G. Takacs, V. Chandrasekhar, J. P. Singh, and B. Girod, "Location coding for mobile image retrieval systems," in P*roc. 5th Int. ICST Mobile Multimedia Commun. Conf.*, London, Sep. 2009, paper 8.
[44] Chunyu Wang et al., "Scalable location coding towards low–bit rate visual search," Institute of Digital Media, Peking University, Tech. Rep. 2012.
[45] MPEG, "Reference results of key point reduction," ISO/IEC JTC1/SC29/WG11/ M23929, 2012/02.
[46] T. Joachims, "Optimizing search engines using click through data," in *Proc. 8th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, ACM, New York, NY, 2002, pp. 133–142.
[47] MPEG, "Peking University's Response to CDVS Core Experiment 1", JTC1/SC29/WG11/ m24781, 2012/4.
[48] MPEG, "Aptina Interest Point and Descriptor Method Summary," ISO/IEC JTC1/SC29/WG11/ m22775, 2011/11.

Manuscript received: March 8, 2012

Biographies

Ling–Yu Duan (lingyu@pku.edu.cn) received his MSc degree in automation from The University of Science and Technolohy, China, in 1999. He received his MSc degree in computer science from the National University of Singapore in 2002 and his PhD degree in information technology from The University of Newcastle, Australia, in 2007. From 2003 to 2008, he was a research scientist at the Institute for Infocomm Research, Singapore. Since 2008, he has been an associate professor at the School of Electrical Engineering and Computer Science at Peking University. Dr. Duan currently His research interests include visual search and reality augmentation, multimedia content analysis, and mobile media computing. He has authored more than 70 papers in these areas.

Jie Chen (cjie@pku.edu.cn) is a PhD candidate at the School of Electrical Engineering and Computer Science, Peking University. His research interest include mobile visual search, low bit–rate visual descriptors, and vector quantizer. He has published more than 10 journal or conference papers.

Chunyu Wang (wangchunyu@pku.edu.cn) is a PhD candidate at the School of Electrical Engineering and Computer Science, Peking University. His research interests include visual search, object recognition, and activity recognition.

Rongrong Ji (rrji@pku.edu.cn) received his PhD degree in computer science from Harbin Institute of Technology, China. He is currently a postdoctoral research fellow at Columbia University. His research interests include image and video search, content analysis and understanding, mobile visual search and recognition, and interactive human–computer interface. Dr. Ji received the Best Paper Award at ACM Multimedia 2011 and received a Microsoft Fellowship in 2007.

Tiejun Huang (tjhuang@pku.edu.cn) received his BSc and MSC degrees in computer science from Wuhan University of Technology in 1992 and 1995. He received his PhD degree in pattern recognition and image analysis from Huazhong University of Science and Technology, China, in 1998. He is currently a professor in the School of Electrical Engineering and Computer Science, Peking University. He is also vice director of the National Engineering Laboratory for Video Technology of China. His research interests include video coding, image understanding, digital rights management (DRM), and digital library. He has published more than sixty peer–reviewed papers and has authored or co–authored three books. He is a member of the board of directors for the Digital Media Project; he is on the advisory board for IEEE Computing Now; he is on the editorial board of the Journal on 3D Research; and he is on the board of the Chinese Institute of Electronics.

Wen Gao (wgao@pku.edu.cn) received his MSc degree in computer science from Harbin Institute of Technology, China, in 1985. He received his PhD degree in electronics engineering from the University of Tokyo in 1991. He is a professor in the School of Electronics Engineering and Computer Science, Peking University. He has led research efforts in video coding, face recognition, sign language recognition and synthesis, and multimedia retrieval. Professor Gao was admitted as an Academedian of the China Engineering Academy in 2011 and became an IEEE Fellow in 2010 for his contribution to video coding technology. He has been on the editorial boards of IEEE Trans. on Multimedia, IEEE Trans. Circuits Syst. For Video Tech., and several other top international academic journals. He was the chair of IEEE Int. Conf. Multimedia & Expo (ICME) 2007, and ACM Int. Conf. Multimedia (ACM–MM) 2009. He has authored four books and published more than 500 research papers on video coding, signal processing, computer vision, and pattern recognition.

D:\EMAG\2011–02–29/VOL9\CONTETN.VFT——1PPS/P69