TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY
Julkaisu 796 • Publication 796

Miska M. Hannuksela

Error-Resilient Communication Using the H.264/AVC Video Coding Standard

Tampere 2009

Tampereen teknillinen yliopisto. Julkaisu 796
Tampere University of Technology. Publication 796

Miska M. Hannuksela

Error-Resilient Communication Using the H.264/AVC Video Coding Standard

Thesis for the degree of Doctor of Technology to be presented with due permission for public examination and criticism in Tietotalo Building, Auditorium TB111, at Tampere University of Technology, on the 23rd of March 2009, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2009

ISBN 978-952-15-2115-7 (printed)
ISBN 978-952-15-2132-4 (PDF)
ISSN 1459-2045

Tampereen Yliopistopaino Oy, 2009

Abstract

The Advanced Video Coding standard (H.264/AVC) has become a widely deployed coding technique used in numerous products and services, such as Blu-ray Disc, Adobe Flash, video conferencing, and mobile television. H.264/AVC utilizes predictive coding to achieve high compression ratio. However, predictive coding also makes H.264/AVC bitstreams vulnerable to transmission errors, as prediction incurs temporal and spatial propagation of the degradations caused by transmission errors. Due to the delay constraints of real-time video communication applications, transmission errors cannot usually be tackled by reliable communication protocols. Yet, most networks are susceptible to transmission errors. Consequently, error resilience techniques are needed to combat transmission errors in real-time H.264/AVC-based video communication. The aim of the thesis is to improve the error robustness of H.264/AVC in real-time video communication applications.

Error resilience techniques applicable for H.264/AVC-based real-time video communication are reviewed in the thesis. Error resilience techniques are commonly classified into interactive error control, forward error correction and concealment, and error concealment by post-processing. Interactive error control methods try to avoid the emergence of transmission errors proactively or compensate the transmission errors reactively by cooperation between the transmitter and the receiver. Forward error correction and concealment refer to those techniques in which the transmitter adds redundancy to the transmitted data enabling the receiver to recover or estimate the contents of the transmitted data even if there were transmission errors. Both interactive error control and forward error correction and concealment can be applied equally to all parts of a transmitted bitstream or unequally, e.g., being biased by the impact of the respective protected part on the reconstructed video quality. Error concealment by post-processing refers to the estimation of the correct representation of erroneously received data. The thesis also discusses the choice of the most useful error resilience techniques, which depends on the application and network in use.

The thesis presents methods to improve error resilience from the level achievable by earlier methods. The presented methods can be grouped into three topics: isolated regions, sub-sequences and interleaved transmission, and encoder-assisted error concealment. The isolated regions technique falls into the category of forward error concealment methods and it can also be used as a tool for region-of-interest partitioning for unequal error protection. The sub-sequence technique provides means for hierarchical temporal adaptation of the coded bitstream. In other words, parts of the bitstream can be decoded to obtain a subsampled picture

rate. It is shown that the sub-sequence technique improves compression efficiency compared to non-hierarchical temporal scalability and non-scalable bitstreams. Furthermore, two error resilience schemes utilizing the sub-sequence technique are presented: an unequal error protection scheme in which interleaved transmission is required and a forward error concealment scheme called intra picture postponement. In the final part of the thesis, two encoder-assisted error concealment methods are presented. These are shown to improve the handling of transmission errors in certain situations.

A part of the research work presented in this thesis was targeted at the H.264/AVC standard. Specifically, isolated regions, sub-sequences, and the presented encoder-assisted error concealment methods were adopted into H.264/AVC, and the interleaved transmission feature was included in the specification for real-time carriage of H.264/AVC bitstreams over the Internet Protocol.

Preface

The research presented in this thesis has been carried out during the years 2000-2007 at Nokia, Tampere, and at the Department of Signal Processing of Tampere University of Technology. During the preparation of the thesis, the author worked with several Nokia units, including Mobile Phones, Mobile Software, and Research Center. One of the papers included in the thesis was supported by Radio- ja Televisiotekniikan Tutkimus Oy.

First and the foremost, I wish to express my deepest gratitude to my supervisor Prof. Moncef Gabbouj for encouragement and scientific guidance throughout the years as well as careful review of the thesis. I would also like to thank Prof. Gabbouj for fruitful collaboration between Nokia and the Department of Signal Processing of Tampere University of Technology.

I would like to thank the reviewers of the thesis, Prof. Olli Silvén and Dr. Nikolaus Färber, for their valuable and constructive comments.

Most papers included in this thesis were prepared in a collaboration project between Nokia and Prof. Gabbouj's team at Tampere University of Technology. I owe many thanks to my former superior Janne Juhola for setting up the collaboration project in Nokia side.

Many of the first papers included in this thesis were prepared in collaboration with Dr. Ye-Kui Wang and Dr. Dong Tian. I would like to thank them for their countless hours spent for this work. I would also like to thank all the other co-authors of the papers for providing essential contribution for the thesis: Dr. Thomas Stockhammer, Prof. Thomas Wiegand, Kerem Caglar, Vinod Kumar Malamal Vadakital, Dr. Stephan Wenger, Dr. Mehdi Rezaei, and Satu Jumisko-Pyykkö. Furthermore, I am grateful to Ye-Kui, Dong, and Vinod for letting me reuse a few figures that they created originally.

During the years I have had the pleasure of working with great colleagues at Nokia and Tampere University of Technology. I would especially like to thank Dr. Petri Haavisto and Dr. Roberto Castagno for reviewing my first publications carefully and helping me get started with my researcher career.

Last but not least, I wish to express my warmest thanks to my parents, Matti and Aila Hannuksela, who have very much encouraged me to complete the thesis.

Tampere, February 2009 Miska Hannuksela

Contents

List of Publications

This thesis is written on the basis of the following publications.

[P1] M. M. Hannuksela, "Simple packet loss recovery method for video streaming," *Proceedings of the 11th International Packet Video Workshop*, pp. 138-143, Apr. 2001.

[P2] D. Tian, M. M. Hannuksela, Y.-K. Wang, and M. Gabbouj, "Error resilient video coding techniques using spare pictures," *Proceedings of the International Packet Video Workshop*, Apr. 2003.

[P3] T. Stockhammer, M. M. Hannuksela, and T. Wiegand, "H.264/AVC in wireless environments," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 657-673, Jul. 2003.

[P4] Y.-K. Wang, M. M. Hannuksela, K. Caglar, and M. Gabbouj, "Improved error concealment using scene information," *Proceedings of the International Workshop VLBV03*, published as *Lecture Notes in Computer Science*, vol. 2849/2003, pp. 283-289, Springer, Sep. 2003.

[P5] M. M. Hannuksela, Y.-K. Wang, and M. Gabbouj, "Isolated regions in video coding," *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 259-267, Apr. 2004.

[P6] D. Tian, M. M. Hannuksela, and M. Gabbouj, "Sub-sequence video coding for improved temporal scalability," *Proceedings of IEEE International Symposium on Circuits and Systems*, vol. 6, pp. 6074-6077, May 2005.

[P7] D. Tian, V. K. Malamal Vadakital, M. M. Hannuksela, S. Wenger, and M. Gabbouj, "Improved H.264/AVC video broadcast/multicast," *Proceedings of Visual Communications and Image Processing 2005*, published as *Proceedings of SPIE*, vol. 5960, pp. 71-82, Jul. 2005.

[P8] T. Stockhammer and M. M. Hannuksela, "H.264/AVC video for wireless transmission," *IEEE Wireless Communications*, vol. 12, no. 4, pp. 6-13, Aug. 2005.

[P9] V. K. Malamal Vadakital, M. M. Hannuksela, M. Rezaei, and M. Gabbouj, "Method for unequal error protection in DVB-H for mobile television," *Proceedings of IEEE In-*

ternational Symposium on Personal, Indoor and Mobile Radio Communications, Sep. 2006.

[P10] M. M. Hannuksela, V. K. Malamal Vadakital, and S. Jumisko-Pyykkö, "Comparison of error protection methods for audio-video broadcast over DVB-H," *EURASIP Journal on Advances in Signal Processing*, doi:10.1155/2007/71801, 2007.

List of Supplementary Publications

The following supplementary publications support the novel techniques and results of the thesis but have not undergone a thorough academic review process or are not as essential for the thesis as the publications listed earlier.

[S1] M. M. Hannuksela, “New Annex W functions for error resilience,” ITU-T Video Coding Experts Group document Q15-J-55, May 2000 <http://ftp3.itu.ch/av-arch/video-site/0005_Osa/q15j55.doc>.

[S2] M. M. Hannuksela, “Enhanced concept of GOP,” Joint Video Team document JVT-B042, Jan. 2002 <http://ftp3.itu.ch/av-arch/jvt-site/2002_01_Geneva/JVT-B042r1.doc>.

[S3] Y.-K. Wang and M. M. Hannuksela, “Error-robust video coding using isolated regions,” Joint Video Team document JVT-C073, May 2002 <http://ftp3.itu.ch/av-arch/jvt-site/2002_05_Fairfax/JVT-C073.doc>.

[S4] Y.-K. Wang and M. M. Hannuksela, “Gradual decoder refresh using isolated regions,” Joint Video Team document JVT-C074, May 2002 <http://ftp3.itu.ch/av-arch/jvt-site/2002_05_Fairfax/JVT-C074.doc>.

[S5] M. M. Hannuksela, “Signaling of enhanced GOP information,” Joint Video Team document JVT-C080, May 2002 <http://ftp3.itu.ch/av-arch/jvt-site/2002_05_Fairfax/JVT-C080.doc>.

[S6] M. M. Hannuksela, “Signaling of enhanced GOPs,” Joint Video Team document JVT-D098, Jul. 2002 <http://ftp3.itu.ch/av-arch/jvt-site/2002_07_Klagenfurt/JVT-D098.doc>.

[S7] Y.-K. Wang and M. M. Hannuksela, “Signaling of shot changes,” Joint Video Team document JVT-D099, Jul. 2002 <http://ftp3.itu.ch/av-arch/jvt-site/2002_07_Klagenfurt/JVT-D099.doc>.

[S8] D. Tian, Y.-K. Wang, and M. M. Hannuksela, “Spare pictures,” Joint Video Team document JVT-D100, Jul. 2002 <http://ftp3.itu.ch/av-arch/jvt-site/2002_07_Klagenfurt/JVT-D100.doc>.

[S9] M. M. Hannuksela, Y.-K. Wang, and M. Gabbouj, "Sub-picture video coding for un-equal error protection," *Proceedings of European Signal Processing Conference*, vol. 2, pp. 526-529, Sep. 2002.

[S10] Y.-K. Wang and M. M. Hannuksela, "Motion-constrained slice group indicator," Joint Video Team document JVT-E129, Oct. 2002 <http://ftp3.itu.ch/av-arch/jvt-site/2002_10_Geneva/JVT-E129.doc>.

[S11] M. M. Hannuksela and D. Tian, "Video simulations for MBMS streaming," 3GPP TSG-SA4 document S4-040671, Nov. 2004 <http://www.3gpp.org/ftp/tsg_sa/WG4_CODEC/TSGS4_33/Docs/S4-040671.zip>.

[S12] S. Wenger, M. M. Hannuksela, T. Stockhammer, M. Westerlund, and D. Singer, "RTP Payload Format for H.264 Video," IETF Request for Comments 3984, Feb. 2005 <http://www.ietf.org/rfc/rfc3984.txt>.

List of Acronyms

3GPP	Third Generation Partnership Project
AC coefficient	All other transform coefficients of a block than the DC coefficient
ACK	Positive acknowledgement
ADT	Application Data Table
ARQ	Automatic Repeat reQuest
ASO	Arbitrary Slice Ordering
B	Bi-predicted (picture, slice, or macroblock)
CIF	Common Intermediate Format (352x288 luma samples)
CPB	Coded Picture Buffer
CRC	Cyclic Redundancy Check
DC	Direct Current, represents the mean value of a waveform
DCT	Discrete Cosine Transform
DPB	Decoded Picture Buffer
DVB	Digital Video Broadcasting project
DVB-H	Digital Video Broadcasting – Handheld standard
DVB-T	Digital Video Broadcasting – Terrestrial standard
ETSI	European Telecommunications Standards Institute
EURASIP	European Association for Signal and Image Processing
FEC	Forward Error Correction
FMO	Flexible Macroblock Ordering
fps	frames per second
GDR	Gradual Decoding Refresh
GOP	Group of Pictures
GPRS	General Packet Radio Services
H.264/AVC	Advanced Video Coding standard
HRD	Hypothetical Reference Decoder
HTTP	Hypertext Transfer Protocol

I	Intra-coded (picture, slice, or macroblock)
IDR	Instantaneous Decoding Refresh
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
ISO	International Standardisation Organisation
ITU	International Telecommunication Union
ITU-T	Telecommunications Standardisation Sector of ITU
JM	Joint Model, the reference software of H.264/AVC
JVT	Joint Video Team
kbps	kilobits per second
LA-RDO	Loss-Aware Rate-Distortion-Optimized (macroblock mode selection)
MAC	Medium Access Control
MBAFF	Macroblock-Adaptive Frame-Field
MDC	Multiple Description Coding
MBMS	3GPP Multimedia Broadcast/Multicast Service
MCU	Multipoint Control Unit
MPE	Multi-Protocol Encapsulation
MPE-FEC	Multi-Protocol Encapsulation Forward Error Correction
MPEG	Moving Picture Experts Group
MSVC	Multiple State Video Coding
MVC	Multiview Video Coding, the multiview extension of H.264/AVC
NACK	Negative acknowledgement
NAL	Network Abstraction Layer
P	Predicted (picture, slice, or macroblock)
PDU	Protocol Data Unit
PSNR	Peak Signal-to-Noise Ratio
PSS	3GPP Packet-switched Streaming Service
QCIF	Quarter Common Intermediate Format (176x144 luma samples)
QoS	Quality of Service
RFC	Request for Comments
RLC	Radio Link Control
RS	Reed-Solomon (FEC coding)

RTCP	Real-time Transport Control Protocol
RTP	Real-time Transport Protocol
RTP/AVP	RTP profile for audio and video conferences with minimal control
RTP/AVPF	Audio-visual RTP profile with feedback
RTSP	Real-Time Streaming Protocol
SD	Standard Definition
SDP	Session Description Protocol
SDU	Service Data Unit
SEI	Supplemental Enhancement Information
SPIE	Society of Photo-Optical Instrumentation Engineers
SVC	Scalable Video Coding standard, the scalable extension of H.264/AVC
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UEP	Unequal Error Protection
UMTS	Universal Mobile Telecommunications System
UTRAN	UMTS Terrestrial Radio Access Network
VCEG	Video Coding Experts Group
VCL	Video Coding Layer
W3C	World Wide Web Consortium
XOR	Exclusive or operation

List of Tables

List of Figures

Chapter 1

Introduction

Digital video communication systems, such as digital television and video streaming over the Internet, belong to the every-day life of many people. A simplified block diagram of a general video communication system is presented in Figure 1 [143]. Due to the fact that uncompressed video requires a huge bandwidth, the input video is compressed by the source coder to a desired bitrate. The source coder can be divided into two components, namely the waveform coder and the entropy coder. The waveform coder performs lossy video signal compression, whereas the entropy coder losslessly converts the output of the waveform coder into a bitstream. The transport coder encapsulates the compressed video according to the communication protocols in use. Then, the data is transmitted to the receiver side via a transmission channel. The receiver performs inverse operations to obtain a reconstructed video signal for display.

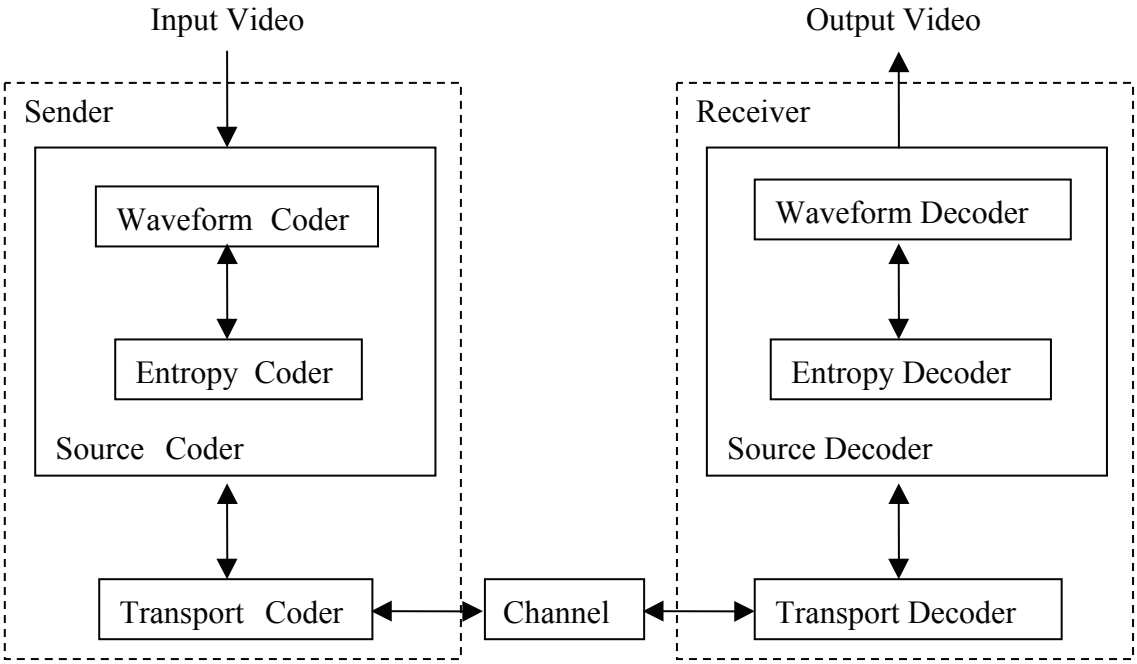


Figure 1. Functional block diagram for a video communications system.

Predictive coding is utilized in the waveform coder to achieve high compression efficiency. There are two basic types of prediction: intra and inter. Intra prediction refers to the estimation of a pixel block in a picture from other areas of the same picture. In inter prediction, a pixel block is estimated based on previous pictures, usually by indicating the location of a similar pixel block in a previous picture as a motion vector.

As most real-world channels are susceptible to transmission errors, certain measures need to be taken in order to protect the video data from such errors. While error-free communication can be achieved by retransmitting data packets until they are correctly received, real-time video communication cannot rely solely on retransmission due to delay constraints arising from user expectations and requirements. For example, the end-to-end delay in video telephony is expected to be such low that natural conversation is not disturbed – usually end-to-end delay less than 100 ms is considered desirable [9]. Moreover, retransmission is unfeasible in broadcast applications due to the unidirectional channel.

Predictive coding makes video vulnerable to transmission errors. Not only are the regions that correspond to the lost or corrupted transmission packets visibly damaged, but the damaged regions are also propagated spatially and temporally. When a degraded region is used as a source for intra prediction or inter prediction, the damaged area becomes larger or spans in time, respectively. Figure 2 presents three consecutive coded pictures illustrating how a degraded region is propagated temporally and spatially after a transmission error occurring in a previous picture.

Due to the inability to use reliable transmission for real-time video communication and the vulnerability of coded video, transmission errors have to be handled carefully in video communication systems. In general, transmission errors should be first detected and then corrected or concealed. Error correction refers to the capability to recover erroneous data perfectly as if no errors were ever present in the received bitstream. Error concealment refers to the capability to conceal degradations caused by transmission errors so that they become hardly visible in the reconstructed video. Some amount of redundancy is typically added into the transmitted data stream in source or transport coding in order to help in error detection, correction and concealment. [143]

Figure 2. Example of temporal error propagation.

Error resilience techniques can be roughly classified into three categories as suggested in [143]: interactive error control, forward error correction and concealment, and error concealment by post-processing. Interactive error control can be further split into two classes. First, congestion control methods aim at avoiding losses proactively by reacting to channel and receiver state feedback [42]. Second, in methods falling into the category of interactive error correction and concealment, the transmitter and the receiver co-operate in order to minimize the degradations caused by transmission errors. Forward error correction (FEC) refers to those techniques in which the transmitter adds redundancy, often known as parity or repair symbols, to the transmitted data, enabling the receiver to recover the transmitted data even if there were transmission errors. In systematic FEC codes, the original video bitstream appears as such in encoded symbols, while encoding with non-systematic codes does not recreate the original video bitstream as output. Methods in which additional redundancy provides means for approximating the lost content are classified as forward error concealment techniques. Error concealment by post-processing refers to the estimation of the correct representation of erroneously received data.

The channel, as referred to in Figure 1, usually consists of one or more networks including network elements and links connecting those network elements. Data communication over the channel is typically considered to comply with a stack of communication protocols usually organized in layers, including a physical layer, a link layer, a network layer, a transport layer, and an application layer [174]. The physical layer provides physical means for connections between network elements, whereas the link layer manages data links between network elements. The network layer provides addressing of end-points and performs routing of transmitted data through the network. The transport layer provides a connection-oriented or connectionless end-to-end message transfer functionality between end-points. The application layer is the top-most layer in the protocol stack and serves directly the end-user. Source coding is considered to be included in the application layer. Error correction and concealment techniques can operate in any layer of the protocol stack [36].

Equal error protection refers to error resilience techniques applied identically to all parts of video bitstreams. However, a transmission error may have a very different impact on visual quality depending on which part of a video bitstream it hits. Therefore, certain parts of a video bitstream may need better protection than others in order to improve visual quality of the reconstructed video when the bitstream is conveyed over an error-prone channel. This approach is exploited in unequal error protection (UEP) methods. Partitioning of a bitstream to parts of different priorities is a prerequisite for UEP. Interactive error control methods or forward error correction and concealment methods can then be used to provide error resilience strength according to a derived priority. [143]

Standardization aims at creating specifications that enable development of interoperable implementations. Standards therefore have an essential role in open communication systems. The Advanced Video Coding standard [68][69][70], referred to as H.264/AVC, is one of the most recently specified video compression standards. As most other video coding standards, it specifies the bitstream format and the decoding process for compliant bitstreams. H.264/AVC

improves compression efficiency substantially compared to previous standards [159], such as MPEG-2 Video [66] and H.263 [67], and provides flexibility for a wide variety of applications and networks [P3][P8]. H.264/AVC is deployed extensively in products and services such as Blu-ray Disc, Adobe Flash, video conferencing, and mobile television.

1.1. OUTLINE AND OBJECTIVES OF THE THESIS

The thesis presents methods for reducing the quality degradation caused by transmission errors in video communication systems using H.264/AVC. Other factors affecting the end-user satisfaction, such as compression efficiency and end-to-end latency, are omitted or regarded as constraints when optimizing error resilience. Particular emphasis is given to video communication applications that are relevant for mobile handheld devices. The goal of the research is to improve the error resilience of H.264/AVC video transmission compared to earlier standards.

This thesis primarily focuses on error resilience techniques that operate in the application layer and involve H.264/AVC encoders and/or decoders. It is evident, however, that proper error resilience design in a video communication system requires interoperation of several protocol stack layers [113]. Thus, the thesis also pays attention to relevant error resilience features of layers below the application layer and considers cross-layer optimization of error resilience.

The research work presented in this thesis can be categorized into three areas. First, the isolated regions technique provides means for forward error concealment as well as priority partitioning. Second, the sub-sequence technique and interleaved transmission together can be applied to congestion control, unequal forward error correction, and forward error concealment. Third, two methods for encoder-assisted error concealment are presented.

The thesis is organized as follows: The H.264/AVC coding standard is reviewed in Chapter 2, the focus being on those features that are relevant for the thesis. Chapter 3 provides an overview of the most relevant video communication applications and systems for the thesis as well as the communication protocols used in these systems. Chapter 4 contains a literature review of error resilience techniques applicable to video communication systems using H.264/AVC. Chapters 5, 6, and 7 summarize the main contributions provided in this thesis, i.e., isolated regions, sub-sequences and interleaved transmission, and encoder-assisted error concealment methods, respectively. Finally, conclusions are drawn in Chapter 8.

1.2. PUBLICATIONS AND AUTHOR'S CONTRIBUTIONS

The author was strongly involved in the development of H.264/AVC. Particularly, the author proposed or was involved in the development of many error resilience features of H.264/AVC. Publications [P3] and [P8] review the use of H.264/AVC in wireless transmission environments and embody the major contribution of the author in this domain. These publications were jointly prepared by all their authors. Chapters 2, 3, and 4 describe the relevant features of H.264/AVC, discuss multimedia services for wireless networks, and provide

an exhaustive review of error resilience techniques applicable for H.264/AVC-based video communication. These chapters extend the reviews provided in [P3] and [P8].

The isolated regions technique, presented in [P5], falls into the category of forward error concealment methods and it can also be used as a tool for priority partitioning for unequal error protection. The author was one of the two inventors for the isolated regions technique. He was also responsible of authoring [P5] and supervising the related implementation and simulation work. The isolated regions method is summarized in Chapter 5.

Sub-sequences provide a mechanism for temporal adaptation of video bitstreams. Publications [P1], [P6], [P7], [P9], and [P10] relate to sub-sequences and their applications for compression efficiency and error resilience. A summary of these publications is provided in Chapter 6.

An idea to encode a chain of predicted pictures in reverse output order in addition to conventionally predicted pictures was presented in [P1]. This method belongs to the category of forward error concealment methods, as the additional prediction chain limits temporal error propagation compared to conventional bitstreams. The author was responsible for all the research for [P1].

Sub-sequences can be used for hierarchical temporal scalability, which was shown to improve compression efficiency compared to non-scalable bitstreams and non-hierarchical scalable bitstreams in [P6]. Hierarchical temporal scalability can be used for priority partitioning for unequal error protection, as shown in the subsequent publications (see below). The author designed the sub-sequence feature in H.264/AVC and proposed the use of sub-sequences for hierarchical temporal scalability [S2]. The author also supervised the implementation and simulation work for [P6].

An uneven level of forward error correction can be provided for different layers of scalable bitstreams, when an FEC code is separately calculated for each layer. When a block of data packets of a layer is transmitted subsequently, FEC decoding operation is similar to that for the equal error protection. Consequently, the transmission order differs from the decoding order of data, and hence the transmission mechanism has to provide means to recover the decoding order in the receiver. This unequal error protection method was studied with temporal scalability in mobile cellular network environment [P7] and in television broadcast network environment [P9][P10]. The author designed the support for real-time H.264/AVC data transmission out of decoding order over the Internet Protocol for the respective standard [S12]. The author was also the originator of the method for unequal error protection in [P7], [P9], and [P10], and he supervised the work for these papers.

Two methods for improving error concealment by post-processing using additional information provided by the encoder were proposed in [P2] and [P4]. The original idea of the spare picture method [P2] was proposed by the author, while the detailed design for H.264/AVC was done jointly by the research team and the respective implementation and simulations were supervised by the author. The author was one of the two original inventors for the scene information method [P4]. The research team expanded the original design for

H.264/AVC jointly, and the author supervised the implementation and simulation work. Chapter 7 contains a summary of publications [P2] and [P4].

Chapter 2

The H.264/AVC Video Coding Standard

The H.264/AVC standard was developed by the Joint Video Team (JVT) of the Video Coding Experts Group (VCEG) of the Telecommunications Standardisation Sector of International Telecommunication Union (ITU-T) and the Moving Picture Experts Group (MPEG) of International Standardisation Organisation (ISO) / International Electrotechnical Commission (IEC). The H.264/AVC standard is published by both parent standardization organizations, and it is referred to as ITU-T Recommendation H.264 and ISO/IEC International Standard 14496-10, also known as MPEG-4 Part 10 Advanced Video Coding (AVC). By the time of the publication of this thesis, there have been eight versions of the H.264/AVC standard, each integrating new features to the specification. Some of the most important versions include the following. Version 1 [68] refers to the first (2003) approved version of the standard. Version 4 [69] refers to the integrated text containing the "Fidelity range extensions" amendment. Version 8 [70] refers to the standard including the Scalable Video Coding (SVC) amendment. The reference software for H.264/AVC, known as the Joint Model (JM), is also published by both ITU-T [71] and ISO/IEC [63], but the JVT constantly updates the latest version [131]. The JVT has also finalized the Multiview Video Coding (MVC) extension for H.264/AVC [138], and a new version of the H.264/AVC standard including the MVC extension was in the approval process at the time of writing this thesis.

Similarly to earlier video coding standards, the bitstream syntax and semantics as well as the decoding process for error-free bitstreams are specified in H.264/AVC. The encoding process is not specified, but encoders must generate conforming bitstreams. Bitstream and decoder conformance can be verified with the Hypothetical Reference Decoder (HRD), which is specified in Annex C of H.264/AVC. The standard contains coding tools that help in coping with transmission errors and losses, but the use of the tools in encoding is optional and no decoding process has been specified for erroneous bitstreams.

The elementary unit for the input to an H.264/AVC encoder and the output of an H.264/AVC decoder is a picture. A picture may either be a frame or a field. A frame com-

prises a matrix of luma samples and corresponding chroma samples. A field is a set of alternate sample rows of a frame and may be used as encoder input, when the source signal is interlaced. A macroblock is a 16x16 block of luma samples and the corresponding blocks of chroma samples. A picture is partitioned to one or more slice groups, and a slice group contains one or more slices. A slice consists of an integer number of macroblocks ordered consecutively in the raster scan within a particular slice group.

The elementary unit for the output of an H.264/AVC encoder and the input of an H.264/AVC decoder is a Network Abstraction Layer (NAL) unit. Decoding of partial or corrupted NAL units is typically remarkably difficult. For transport over packet-oriented networks or storage into structured files, NAL units are typically encapsulated into packets or similar structures. A bytestream format has been specified in H.264/AVC for transmission or storage environments that do not provide framing structures. The bytestream format separates NAL units from each other by attaching a start code in front of each NAL unit. To avoid false detection of NAL unit boundaries, encoders must run a byte-oriented start code emulation prevention algorithm, which adds an emulation prevention byte to the NAL unit payload if a start code would have occurred otherwise. In order to enable straightforward gateway operation between packet- and stream-oriented systems, start code emulation prevention is performed always regardless of whether the bytestream format is in use or not.

The intent of this chapter is to review those features of H.264/AVC that are essential for the scope of the thesis. Section 2.1 reviews certain profiles and levels specified for H.264/AVC. The types of predictive coding applied in H.264/AVC are overviewed in Section 2.2, as propagation of transmission errors can be limited by constraining predictive coding. Slices and slice groups, introduced in Section 2.3, are the basic units for picture partitioning and coded data encapsulation into transmission packets. Section 2.4 reviews how multiple reference pictures for inter prediction are managed in the decoding process, while Section 2.5 presents how reference pictures for inter prediction and pictures to be ordered in correct output order are managed in the decoded picture buffer (DPB). Section 2.6 describes the bitstream structure of H.264/AVC streams. Finally, Section 2.7 discusses picture output order and timing.

2.1. PROFILES AND LEVELS

A number of profiles and levels are specified in H.264/AVC. A profile consists of a subset of the algorithmic features or coding tools of the standard and a set of constraints on those features. A profile is typically targeted for a family of applications sharing similar trade-offs between memory, processing, latency, and error resilience requirements. A level corresponds to a set of limits mainly on memory requirements and computational performance. Decoders conforming to a profile must support all the features of a profile, whereas encoders have the freedom to select which features of the profile are used to produce compliant bitstreams. A decoder conforming to a level must be capable of decoding any bitstream that conforms to the level. In other words, levels give minimum requirements for decoders and constraints for bit-

streams and encoders. The specified profiles and levels quantize the numerous operation points of H.264/AVC codecs to a manageable number and hence help in facilitating interoperability between codec implementations and applications. The pair of profile and level is used to indicate the characteristics of a bitstream in a session announcement. For example, a streaming server can indicate the characteristics of an offered stream by its profile and level. In video conferencing applications, the pair of profile and level can indicate the capability of a decoder and hence be used to negotiate a common operation point during the session setup.

There are a number of profiles specified in H.264/AVC, out of which the Baseline and High profiles are briefly reviewed below. These two profiles are required or recommended in multimedia service standards that are relevant for this thesis.

The Baseline profile of H.264/AVC suits low-latency applications, such as video conferencing, in which error resilience in source coding is required. It includes all the fundamental features of the H.264/AVC standard, thus providing a very good compression performance. In addition, it contains arbitrary slice ordering (ASO, see Section 2.3), flexible macroblock ordering (FMO, see Section 2.3) and redundant slices (see Section 2.6.2) error resilience features. The Baseline profile is meant for progressive scan content only, i.e., no field coding tools are included in the Baseline profile.

The High profile allows the use of bi-predictive slices and weighted prediction, which improve the compression efficiency especially in applications with relaxed latency requirements at the expense of increased computational requirements. Furthermore, the High profile includes coding tools for interlaced content and context-based adaptive binary arithmetic coding (CABAC) for more efficient entropy coding. It excludes the error resilience tools mentioned above, and therefore it suits playback from local mass storage and broadcast applications for which more latency can be allowed and more efficient decoder implementations can be afforded compared to the conferencing applications and the Baseline profile computational requirements, respectively.

The intersection of the Baseline and High profiles is referred to as the Constrained Baseline in this thesis. The Constrained Baseline is not a profile specified in H.264/AVC. However, H.264/AVC enables the indication of Baseline bitstreams that are also compliant with the High profile, or vice versa; therefore, in practice indicating that the bitstreams conform to the Constrained Baseline. Furthermore, many multimedia service standards enable indication of the Constrained Baseline in the codec capability exchange procedure. Therefore, it can be treated analogously to profiles in most applications. The Constrained Baseline suits applications that do not require the error resilience features mentioned above and cannot afford the computational complexity that is inherent in those High profile tools that are excluded from the Constrained Baseline. For example, the Constrained Baseline is recommended in the Packet-switched Streaming Service (PSS) [2] and the Multimedia Broadcast/Multicast Service (MBMS) [3] for mobile networks.

The specified levels of H.264/AVC correspond to such memory requirements that range from picture sizes such as Quarter Common Intermediate Format (QCIF, corresponding to 176x144 luma samples) to picture extents of thousands of samples. The addressed bitrates

range similarly from tens of kilobits per second to several hundred megabits per second. Hence, the specified levels suit a large variety of applications and devices.

2.2. PREDICTIVE CODING IN H.264/AVC

Video coding is typically a two-stage process: First, a prediction of the video signal is generated based on previous coded data. Second, the residual between the predicted signal and the source signal is coded. Prediction enables efficient compression, but it causes some complications in error-prone environments, in random access, and in parallel decoding. In the following, the types of prediction in H.264/AVC are categorized.

Inter prediction, which is also referred to as temporal prediction and motion compensation, removes redundancy between subsequent pictures. H.264/AVC, as other current video compression standards, divides a picture into a mesh of rectangles, for each of which a similar block in one of the reference pictures is indicated. The location of the prediction block is coded as motion vector that indicates the position of the prediction block compared to the block being coded. The inter prediction process can be characterized using the following factors:

- The accuracy of motion vector representation. It has been shown that sub-pixel accuracy in motion vectors improves compression efficiency [132]. In H.264/AVC, motion vectors are of quarter-pixel accuracy, and sample values in fractional-pixel positions are obtained using a finite impulse response (FIR) filter. Motion vector values are differentially coded relative to the neighboring motion vectors, while differential coding is disabled across slice boundaries.
- Block partitioning for inter prediction. A basic unit for inter prediction in current coding standards is a macroblock, corresponding to a 16x16 block of luma samples and corresponding chroma samples. In H.264/AVC, a macroblock can be further divided to 16x8, 8x16, or 8x8 macroblock partitions, and the 8x8 partition can be further divided to 4x4, 4x8, or 8x4 sub-macroblock partitions, and a motion vector is coded for each partition.
- Number of reference pictures for inter prediction. The sources of inter prediction are previously decoded pictures. In early video coding standards, such as H.261 [65] and MPEG-2 Video [66], only the previous decoded picture is available as a reference for inter prediction. H.264/AVC enables storage of multiple reference pictures for inter prediction and selection of the used reference picture on macroblock or macroblock partition basis. Section 2.4 reviews the management of multiple reference pictures in H.264/AVC.
- Multi-hypothesis motion-compensated prediction. A theoretical analysis of multi-hypothesis video coding is provided in [48]. H.264/AVC enables linear combination of two motion-compensated prediction blocks for bi-predictive slices, which are also referred to as B slices. In contrast to earlier coding standards, in H.264/AVC the reference pictures for a bi-predictive picture are not

limited to be the subsequent picture and the previous picture in output order, but rather any reference pictures can be used.

- Weighted prediction. Whereas earlier coding standards used a prediction weight of 1 for prediction blocks of inter (P) pictures and 0.5 for each prediction block of a B picture (resulting into averaging), H.264/AVC allows weighted prediction for both P and B slices. In implicit weighted prediction, the weights are proportional to picture order counts (see Section 2.7). Alternatively, prediction weights can be explicitly indicated.

Intra prediction, which is also referred to as spatial prediction, utilizes the fact that adjacent pixels within the same picture are likely to be correlated. Generally speaking, intra prediction can be performed in spatial or transform domain, i.e., either sample values or transform coefficients can be predicted. Intra prediction in H.264/AVC is performed in the spatial domain, by referring to neighboring samples of previously decoded blocks that are to the left and/or above the block to be predicted. In order to avoid spatio-temporal error propagation, which can result when inter prediction has been used for neighboring macroblocks, a constrained intra coding mode can alternatively be selected. In the constrained intra coding mode, intra prediction is performed only from intra-coded neighboring macroblocks.

Three primary types of intra coding are supported in H.264/AVC: intra 4x4, intra 8x8, and intra 16x16 prediction, all applicable to luma blocks. Intra 8x8 prediction modes is available in the High profile but not in the Baseline profile. In intra 4x4 and 8x8 modes, the encoder can select one of the eight directional sample value prediction schemes or use the DC prediction, in which a single value is used to predict the entire block. Intra 4x4 and 8x8 modes are suitable for predicting textures with details. Intra 16x16 prediction includes four modes: vertical, horizontal, DC, and plane. The three first ones are similar to the modes of the 4x4 and 8x8 prediction, whereas the plane prediction mode models the predicted block as a plane and uses position-specific linear functions to obtain sample values. Intra 16x16 prediction is suitable for smooth textures. The chroma samples in intra macroblocks are predicted similarly to 16x16 intra prediction for luma.

One outcome of the coding procedure is a set of coding parameters, such as motion vectors and quantized transform coefficients. Many parameters can be entropy-coded more efficiently if they are predicted first from spatially or temporally neighboring parameters. For example, a motion vector is typically predicted from spatially adjacent motion vectors. Prediction of coding parameters and intra prediction are collectively referred to as in-picture prediction in this thesis.

2.3. SLICES AND SLICE GROUPS

H.264/AVC, as many other video coding standards, allows splitting of a coded picture into slices. In-picture prediction is disabled across slice boundaries. Thus, slices can be regarded as a way to split a coded picture into independently decodable pieces, and slices are therefore elementary units for transmission.

The Baseline profile of H.264/AVC enables the use of up to eight slice groups per coded picture. When more than one slice group is in use, the picture is partitioned into slice group map units, which are equal to two vertically consecutive macroblocks when the macroblock-adaptive frame-field (MBAFF) coding is in use and equal to a macroblock otherwise. The picture parameter set (see Section 2.6.1) contains data based on which each slice group map unit of a picture is associated with a particular slice group. A slice group can contain any slice group map units, including non-adjacent map units. When more than one slice group is specified for a picture, the flexible macroblock ordering (FMO) feature of the standard is used. Some applications for flexible macroblock ordering are presented in Section 4.5.2 and Chapter 5.

In H.264/AVC, a slice consists of one or more consecutive macroblocks (or macroblock pairs, when MBAFF is in use) within a particular slice group in raster scan order. If only one slice group is in use, H.264/AVC slices contain consecutive macroblocks in raster scan order and are therefore similar to the slices in many previous coding standards. When the Baseline profile is in use, slices of a coded picture may appear in any order relative to each other in the bitstream, which is referred to as the arbitrary slice ordering (ASO) feature. Otherwise, slices must be in raster scan order in the bitstream.

2.4. MANAGEMENT OF MULTIPLE REFERENCE PICTURES

Multiple reference pictures for inter prediction have been enabled in modern video coding standards, such as H.263 [67], MPEG-4 Visual [62], and H.264/AVC, to improve error resilience and compression efficiency. The reference picture selection mode (Annex N) of H.263 and the NEWPRED mode of MPEG-4 Visual enable selection of the reference picture for motion compensation per each picture segment, e.g., per each slice in H.263, and are used primarily for error resilience (see Section 4.3). H.264/AVC and the Enhanced Reference Picture Selection mode of H.263 enable selection of the reference picture for each macroblock separately and can be used both for improved compression efficiency and error resilience. This section reviews the features of H.264/AVC related to the management of multiple reference pictures for inter prediction.

The bitstream syntax of video coding standards provides means for detecting coded pictures that can be removed without affecting the decoding of any other pictures. In many video coding standards, such as MPEG-2 Video [66], MPEG-4 Visual [62], and H.263 [67], bi-predictive (B) pictures are not used as prediction references for inter prediction. Consequently, they provide a way to achieve temporal scalability, i.e., B pictures in the named video coding standards can be removed and hence a lower picture rate can be obtained compared to the picture rate of the original bitstream. The bitstream syntax of H.264/AVC indicates whether or not a particular picture is a reference picture for inter prediction of any other picture. Consequently, a picture not used for prediction (a non-reference picture) can be safely disposed. Pictures of any coding type (I, P, B) can non-reference pictures in H.264/AVC.

H.264/AVC specifies the process for decoded reference picture marking in order to control the memory consumption in the decoder. The maximum number of reference pictures used for inter prediction, referred to as *M,* is determined in the sequence parameter set (see Section 2.6.1). When a reference picture is decoded, it is marked as "used for reference". If the decoding of the reference picture caused more than *M* pictures marked as "used for reference", at least one picture must be marked as "unused for reference". There are two types of operation for decoded reference picture marking: adaptive memory control and sliding window. The operation mode for decoded reference picture marking is selected on picture basis. The adaptive memory control enables explicit signaling which pictures are marked as "unused for reference" and may also assign long-term indices to short-term reference pictures. The adaptive memory control requires the presence of memory management control operation (MMCO) parameters in the bitstream. If the sliding window operation mode is in use and there are *M* pictures marked as "used for reference", the short-term reference picture that was the first decoded picture among those short-term reference pictures that are marked as "used for reference" is marked as "unused for reference". In other words, the sliding window operation mode results into first-in-first-out buffering operation among short-term reference pictures.

One of the memory management control operations in H.264/AVC causes all reference pictures except for the current picture to be marked as "unused for reference". An instantaneous decoding refresh (IDR) picture contains only intra-coded slices and causes a similar "reset" of reference pictures. In addition, the reference picture marking process of H.264/AVC facilitates hierarchical temporal scalability, which is discussed in Section 6.1.3.

The reference picture for inter prediction is indicated with an index to a reference picture list. The index is coded with variable length coding, i.e., the smaller the index is, the shorter the corresponding syntax element becomes. Two reference picture lists are generated for each bi-predictive slice of H.264/AVC, and one reference picture list is formed for each inter-coded slice of H.264/AVC. A reference picture list is constructed in two steps: first, an initial reference picture list is generated, and then the initial reference picture list may be reordered by reference picture list reordering (RPLR) commands contained in slice headers. The RPLR commands indicate the pictures that are ordered to the beginning of the respective reference picture list.

The use of multiple reference pictures for improved compression efficiency was originally proposed by Wiegand et al. and they also provided results for the H.263 codec indicating up to about 20% bitrate savings compared to the use of one reference frame [157]. Puri et al. tested the compression improvement of multiple reference frames in H.264/AVC and discovered up to about 10% bitrate savings [101]. The bitrate savings achievable with hierarchical temporal scalability are discussed in Section 6.1.3.

2.5. DECODED PICTURE BUFFERING

The hypothetical reference decoder (HRD), specified in Annex C of H.264/AVC, is used to check bitstream and decoder conformance. The HRD contains a coded picture buffer (CPB),

an instantaneous decoding process, a decoded picture buffer (DPB), and an output picture cropping block. The CPB and the instantaneous decoding process are specified similarly to any other video coding standard, and the output picture cropping block simply crops those samples from the decoded picture that are outside the signaled output picture extents. The DPB was introduced in H.264/AVC in order to control the required memory resources for decoding of conformant bitstreams. There are two reasons to buffer decoded pictures, for references in inter prediction and for reordering decoded pictures into output order. As H.264/AVC provides a great deal of flexibility for both reference picture marking and output reordering, separate buffers for reference picture buffering and output picture buffering could have been a waste of memory resources. Hence, the DPB includes a unified decoded picture buffering process for reference pictures and output reordering. A decoded picture is removed from the DPB when it is no longer used as reference and needed for output. The maximum size of the DPB that bitstreams are allowed to use is specified in the Level definitions (Annex A) of H.264/AVC.

There are two types of conformance for decoders: output timing conformance and output order conformance. For output timing conformance, a decoder must output pictures at identical times compared to the HRD. For output order conformance, only the correct order of output picture is taken into account. The output order DPB is assumed to contain a maximum allowed number of frame buffers. A frame is removed from the DPB when it is no longer used as reference and needed for output. When the DPB becomes full, the earliest frame in output order is output until at least one frame buffer becomes unoccupied.

2.6. STRUCTURE OF H.264/AVC BITSTREAMS

As explained in the introduction of this chapter, H.264/AVC bitstreams contain Network Abstraction Layer (NAL) units in decoding order either in the bytestream format or being externally framed. NAL units consist of a header and payload. The NAL unit header indicates the type of the NAL unit and whether a coded slice contained in the NAL unit is a part of a reference picture or a non-reference picture. The header for SVC NAL units additionally contains various indications related to the scalability hierarchy. NAL unit types and their categorization are presented in Section 2.6.1. NAL units can be clustered into logical entities, such as coded pictures, access units, and coded video sequences, which are reviewed in Section 2.6.2.

2.6.1. Categorization of NAL Units

NAL units can be categorized into Video Coding Layer (VCL) NAL units and non-VCL NAL units. VCL NAL units are either coded slice NAL units, coded slice data partition NAL units, or VCL prefix NAL units. Coded slice NAL units contain syntax elements representing one or more coded macroblocks, each of which corresponds to a block of samples in the uncompressed picture. There are four types of coded slice NAL units: coded slice in an Instantaneous Decoding Refresh (IDR) picture, coded slice in a non-IDR picture, coded slice of an auxiliary coded picture (such as an alpha plane) and coded slice in scalable extension (SVC). A

set of three coded slice data partition NAL units contains the same syntax elements as a coded slice. Coded slice data partition A comprises macroblock headers and motion vectors of a slice, while coded slice data partition B and C include the coded residual data for intra macroblocks and inter macroblocks, respectively. It is noted that the support for slice data partitions is not included in the Baseline or High profile of H.264/AVC. A VCL prefix NAL unit precedes a coded slice of the base layer in SVC bitstreams and contains indications of the scalability hierarchy of the associated coded slice.

A non-VCL NAL unit may be of one of the following types: a sequence parameter set, a picture parameter set, a supplemental enhancement information (SEI) NAL unit, an access unit delimiter, an end of sequence NAL unit, an end of stream NAL unit, or a filler data NAL unit. Parameter sets are essential for the reconstruction of decoded pictures, whereas the other non-VCL NAL units are not necessary for the reconstruction of decoded sample values and serve other purposes presented below. Parameter sets and the SEI NAL unit are reviewed in depth in the following paragraphs. The other non-VCL NAL units are not essential for the scope of the thesis and therefore not described.

Many of the conventional video codecs contain sequence and picture headers embedded in the bitstream. A loss or a corruption of a header typically prevents the correct decoding of the respective part of the bitstream. Thus, to avoid the drastic impact of a header loss, different kinds of header repetition mechanisms are provided both in the source coding specification and with the packetization mechanism. For example, MPEG-4 Visual [62] contains a header extension mechanism for picture header repetition in the slice headers, picture header repetition is enabled through the supplemental enhancement information mechanism of H.263 [51], and the Real-time Transport Protocol (RTP) payload format of H.263 [13] also allows repetition of picture headers.

In order to improve the transmission robustness of infrequently changing coding parameters compared to conventional header repetition, Wenger and Stockhammer [152] proposed the parameter set mechanism. Hannuksela and Wang later proposed parameter sets to be divided to sequence and picture parameter sets [52], which then became the design adopted to H.264/AVC. Parameters that remain unchanged through a coded video sequence are included in a sequence parameter set. In addition to the parameters that are essential to the decoding process, the sequence parameter set may optionally contain video usability information (VUI), which includes parameters that are important for buffering, picture output timing, rendering, and resource reservation. A picture parameter set contains such parameters that are likely to be unchanged in several coded pictures. No picture header is present in H.264/AVC bitstreams but the frequently changing picture-level data is repeated in each slice header and picture parameter sets carry the remaining picture-level parameters. H.264/AVC syntax allows many instances of sequence and picture parameter sets, and each instance is identified with a unique identifier. Each slice header includes the identifier of the picture parameter set that is active for the decoding of the picture that contains the slice, and each picture parameter set contains the identifier of the active sequence parameter set. Consequently, the transmission of picture and sequence parameter sets does not have to be accurately synchronized with

the transmission of slices. Instead, it is sufficient that the active sequence and picture parameter sets are received at any moment before they are referenced, which allows transmission of parameter sets using a more reliable transmission mechanism compared to the protocols used for the slice data. For example, parameter sets can be included as a parameter in the session description for H.264/AVC RTP sessions [S12]. It is recommended to use an out-of-band reliable transmission mechanism whenever it is possible in the application in use. If parameter sets are transmitted in-band, they can be repeated to improve error robustness.

An SEI NAL unit contains one or more SEI messages, which are not required for the decoding of output pictures but assist in related processes, such as picture output timing, rendering, error detection, error concealment, and resource reservation. Several SEI messages are specified in H.264/AVC, and the user data SEI messages enable organizations and companies to specify SEI messages for their own use. H.264/AVC contains the syntax and semantics for the specified SEI messages but no process for handling the messages in the recipient is defined. Consequently, encoders are required to follow the H.264/AVC standard when they create SEI messages, and decoders conforming to the H.264/AVC standard are not required to process SEI messages for output order conformance. One of the reasons to include the syntax and semantics of SEI messages in H.264/AVC is to allow different system specifications to interpret the supplemental information identically and hence interoperate. It is intended that system specifications can require the use of particular SEI messages both in the encoding end and in the decoding end, and additionally the process for handling particular SEI messages in the recipient can be specified.

2.6.2. Grouping of NAL Units into Logical Entities

A coded picture consists of the VCL NAL units that are required for the decoding of the picture. A coded picture can be a primary coded picture or a redundant coded picture. A primary coded picture is used in the decoding process of valid bitstreams, whereas a redundant coded picture is a redundant representation that should only be decoded when the primary coded picture cannot be successfully decoded. More details about redundant coded pictures are provided in Section 4.5.5.

An access unit consists of a primary coded picture and those NAL units that are associated with it. The appearance order of NAL units within an access unit is constrained as follows. An optional access unit delimiter NAL unit may indicate the start of an access unit. It is followed by zero or more SEI NAL units. The coded slices or slice data partitions of the primary coded picture appear next, followed by coded slices for zero or more redundant coded pictures.

A coded video sequence is defined to be a sequence of consecutive access units in decoding order from an IDR access unit, inclusive, to the next IDR access unit, exclusive, or to the end of the bitstream, whichever appears earlier. A group of pictures (GOP) can be decoded regardless of whether any previous pictures were decoded. An open GOP is such a group of pictures in which pictures preceding the initial intra picture in output order may not be correctly decodable. An H.264/AVC decoder can recognize an intra picture starting an

open GOP from the recovery point SEI message in an H.264/AVC bitstream. A closed GOP is such a group of pictures in which all pictures can be correctly decoded. In H.264/AVC, a closed GOP starts from an IDR access unit.

2.7. PICTURE OUTPUT ORDER AND TIMING

One of the design decisions of H.264/AVC was to have output timestamps optionally present in the bitstream syntax to avoid conflicts between the timestamps carried by the transport protocol or in the file storage format. A conflict may arise from concatenation of coded bitstreams or playing at a faster pace than the original decoding speed, for example.

Picture output timing information may be included in the Picture Timing SEI message for systems that do not provide timestamps in the transport or file level. Picture Timing SEI messages indicate the decoding time and output time relative to the operation of the HRD. They may also contain rendering instructions for frame and field duplication in systems that are oriented for fixed picture rate rendering. When H.264/AVC streams are conveyed over RTP, the use of picture timing SEI messages is strongly discouraged and RTP timestamps override any picture timing SEI messages in picture output timing.

Even though picture output timing is not included in the integral part of the bitstream, information on output order was found useful. Hence, a value of picture order count (POC) is derived for each picture and is non-decreasing with increasing picture position in output order relative to the previous IDR picture or a picture containing a memory management control operation marking all pictures as “unused for reference”. POC therefore indicates the output order of pictures. It is also used in the decoding process for implicit scaling of motion vectors in the temporal direct mode of bi-predictive slices, for implicitly derived weights in weighted prediction, and for reference picture list initialization of B slices. Furthermore, POC is used in the verification of output order conformance.

Chapter 3

Video Communication Systems

In this thesis, streaming delivery refers to transmission, reception, decoding, and rendering of multimedia data in real time, where reception, decoding, and rendering happen simultaneously. An unreliable transport protocol is typically used in streaming delivery to achieve low end-to-end delay within a certain limit. Consequently, it may not be possible to reproduce the multimedia data perfectly in the receiver due to transmission errors. The thesis concentrates on handling of transmission errors in streaming video delivery.

Streaming is a collective term for streaming delivery and streaming applications. Streaming delivery is also used in applications beyond streaming, such as video telephone. Streaming applications are often characterized by being essentially half-duplex and utilizing initial buffering before the start of media rendering.

Progressive downloading refers to transmission of a multimedia file over a reliable communication protocol, such as Hypertext Transfer Protocol (HTTP) [41], and decoding and rendering the file while it is being received. While progressive downloading guarantees perfect reproduction of the transmitted file, the receiver may have to halt the playback if the channel throughput falls under the data rate of the file. Even though progressive downloading may be used similarly to streaming delivery in various applications, the technical challenges related to progressive downloading are partly different and not covered in this thesis.

Data delivery methods can also be grouped according to the used routing scheme into unicasting, multicasting, and broadcasting depending on the number of the recipients and their subscriber relationship. Unicasting refers to the transmission of a single message or a content stream to one subscriber. Multicasting is defined as the transmission of a single message or a content stream to a known group of subscribers. Broadcasting means transmission of a single message or a content stream to an unknown and possibly indiscriminated group of subscribers.

This chapter reviews the video communication systems that are relevant for the thesis. First, the types of transmission errors occurring in communications systems are described in Section 3.1. The protocol basis of all real-time multimedia services in the scope of the thesis is the Real-time Transport Protocol (RTP), and therefore the RTP protocol stack is introduced

in Section 3.2. The Internet Protocol (IP) data casting service over Digital Video Broadcasting – Handheld (DVB-H) is overviewed in Section 3.3. Protocols and applications for streaming media delivery over mobile networks are discussed in Section 3.4.

3.1. TYPES OF TRANSMISSION ERRORS

Most real-world channels are susceptible to transmission errors, which can be roughly classified into two categories: bit errors and erasure errors. Bit errors are discussed in this paragraph, whereas erasure errors are reviewed in the next paragraph. Bit errors are caused by physical events occurring in the transmission channel, such as noise and interference. Protocol stacks for real-time media transport typically provide means, such as cyclic redundancy check (CRC) codes, for detection of bit errors. It is a common practice to discard erroneous protocol payloads in the transport decoder, although some attempts have been made to utilize awareness of bit errors across protocol layers [59][73][135]. The challenges in decoding of erroneous video data lie in the likelihood of bursty bit errors [81], the exact detection of the position of the error, and variable length coding (VLC) used by the entropy coder. Due to the burstiness of bit errors, it is likely that a large portion of an erroneous protocol payload would be non-decodable and therefore discarding the entire protocol payload does not cause very much unnecessary data exclusion. The error detection means provided by the communication protocols are typically able to yield a binary conclusion: either the packet is corrupted or correct. It is therefore up to source coding layer means to conclude the exact location of errors. Even though there are methods based on syntactic and semantic violations and unnatural texture disruptions, false detection of bit errors may lead to subjectively annoying video [59]. Due to variable length coding, a single bit error is likely to change the interpretation of the codeword in which it occurs and cause a loss of synchronization of subsequent codewords. Even if codeword synchronization were re-established, it might not be possible to determine the spatial or temporal location of decoded data. Thus, handling of bit errors in the source decoder is not considered in this thesis, but rather it is assumed that the transport decoder passes only correct protocol data units for decoding.

There are two main sources of erasure errors in packet-oriented networks [11]: First, the transport decoder typically processes bit errors by removing the entire packets in which the bit errors occurred. Second, queue overflows in congested network elements, such as routers, cause packet losses. Packets may also undergo a different amount of end-to-end delay, which may cause some packets to miss their decoding or playback time and hence be effectively considered lost. The behavior of the Internet, including packet loss, delay, and reordering statistics, has been studied in various experiments, such as [98] and [102].

3.2. RTP-BASED MEDIA TRANSMISSION

RTP [117] is used for transmitting continuous media data, such as coded audio and video streams in IP-based networks. The Real-time Transport Control Protocol (RTCP) [117] is a companion of RTP, i.e., RTCP should be used to complement RTP, when the network and

application infrastructure allow its use. RTP and RTCP are usually conveyed over the User Datagram Protocol (UDP) [100], which, in turn, is conveyed over the Internet Protocol (IP). There are two versions of IP, IPv4 [99] and IPv6 [28], differing by the number of addressable end-points among other things. RTCP is used to monitor the quality of service provided by the network and to convey information about the participants in an ongoing session. RTP and RTCP are designed for sessions that range from one-to-one communication to large multicast groups of thousands of end-points. In order to control the total bitrate caused by RTCP packets in a multiparty session, the transmission interval of RTCP packets transmitted by a single end-point is proportional to the number of participants in the session. Each media coding format has a specific RTP payload format, which specifies how media data is structured in the payload of an RTP packet. Figure 3 illustrates a simplified protocol stack for RTP-based media transmission.

A number of profiles have been specified for RTP, each of which specifies extensions or modifications to RTP that are specific to a particular family of applications. One of the most popular profiles is called RTP profile for audio and video conferences with minimal control [118] and abbreviated RTP/AVP. The specification provides the semantics of the generic fields in RTP header for the use in audio and video conferences. It also specifies the RTP payload format for some audio and video codecs. An RTP profile known as the audio-visual profile with feedback [97] is abbreviated RTP/AVPF. The audio-visual profile with feedback allows terminals to send feedback faster than RTCP originally allowed and can therefore be used to convey messages for interactive error correction. The codec control messages specified in [156] extend RTP/AVPF with messages that are primarily useful in centralized multipoint conferences.

A media type is specified together with an RTP payload format usually in the same specification. A media type can be used with various protocols, such as HTTP and RTP, to identify the content carried within the protocols. The name of a media type consists of a content type and sub-type separated with a slash, e.g., “video/H264”. Any number of required and optional parameters can be specified for each media type to indicate the characteristics of the media in a more detailed level. Details of media types and their registration are available in [43].

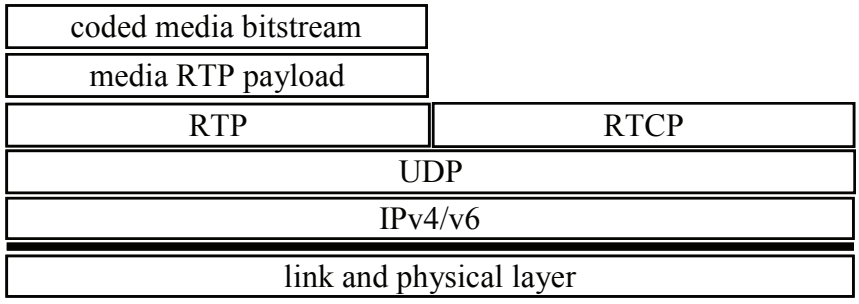


Figure 3. Simplified protocol stack for RTP-based media transmission.

In unicast, multicast, and broadcast streaming applications the available streams are announced and their coding formats are characterized to enable each receiver to conclude if it can decode and render the content successfully. Sometimes a number of different format options for the same content are provided, from which each receiver can choose the most suitable one for its capabilities and/or end-user wishes. The available media streams are described with the corresponding media type and its parameters that are included in a session description formatted according to the Session Description Protocol (SDP) [49]. In unicast streaming applications the session description is usually carried by the Real-Time Streaming Protocol (RTSP) [116], which is used to set up and control the streaming session. In broadcast and multicast streaming applications, the session description may be carried as part of the electronic service guide (ESG) for the service.

In video conferencing applications, the used codecs and their modes are negotiated during the session setup, e.g., using the Session Initiation Protocol (SIP) [110]. Among other things SIP conveys messages according to the SDP offer/answer model [111].

3.3. IP DATA CASTING OVER DVB-H

DVB-H is based on and compatible with DVB-Terrestrial (DVB-T) [37]. The extensions in DVB-H relative to DVB-T make it possible to receive broadcast services in handheld devices. DVB-H is discussed in depth in [40] and reviewed here only when it comes to the most essential parts for the thesis.

The protocol stack for DVB-H is presented in Figure 4. IP packets are encapsulated to Multi-Protocol Encapsulation (MPE) sections for transmission over the Medium Access (MAC) sub-layer. Each MPE section consists of a header, the IP datagram as a payload, and a 32-byte cyclic redundancy check (CRC) for the verification of payload integrity. The MPE section header contains addressing data among other things. The MPE sections can be logically arranged to application data tables in the Logical Link Control (LLC) sub-layer, over which Reed-Solomon (RS) FEC codes [105] are calculated and MPE-FEC sections are formed. The process for MPE-FEC construction is explained in more detail below. The MPE and MPE-FEC sections are mapped onto MPEG-2 Transport Stream (TS) packets.

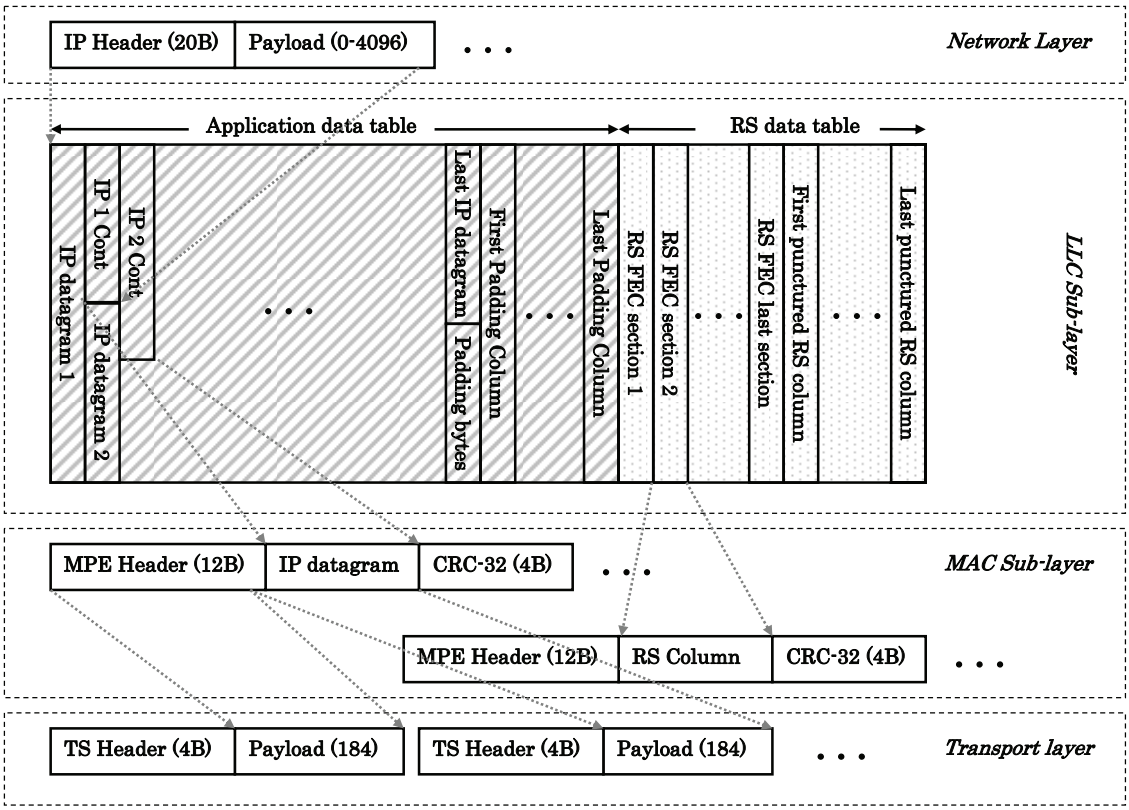


Figure 4. Subset of the protocol stack of DVB-H.

MPE-FEC was included in DVB-H to combat long burst errors that cannot be efficiently corrected in the physical layer. As Reed-Solomon code is a systematic code, i.e., the source data remains unchanged in the FEC encoding, MPE-FEC decoding is optional for DVB-H terminals. MPE-FEC repair data is computed over IP packets and encapsulated into MPE-FEC sections, which are transmitted such a way that an MPE-FEC ignorant receiver could just receive just the unprotected data while ignoring the repair data that follows.

To compute MPE-FEC repair data, IP packets are filled column-wise into an N x 191 matrix where each cell of the matrix hosts one byte and N denotes the number of rows in the matrix. The standard defines the value of N to be one of 256, 512, 768 or 1024. RS codes are computed for each row and concatenated such that the final size of the matrix is of size N x 255. The N x 191 part of the matrix is called the Application data table (ADT) and the next N x 64 part of the matrix is called the RS data table (RSDT). The ADT need not be completely filled, which must be used to avoid IP packet fragmentation between two MPE-FEC frames and may also be exploited to control bitrate and error protection strength. The unfilled part of the ADT is called padding. To control the strength of the FEC protection, all 64 columns of RSDT need not be transmitted, i.e., the RSDT may be punctured. The structure of an MPE-FEC frame is shown in Figure 5 and further detailed information on the MPE-FEC matrix construction can be obtained from [38].

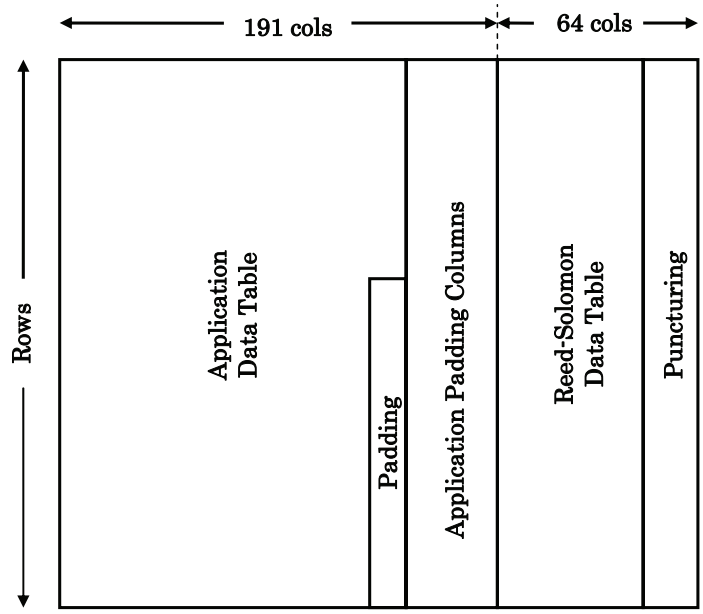


Figure 5. MPE-FEC frame structure.

Mobile devices have a limited source of power. The power consumed in receiving, decoding and demodulating a standard full-bandwidth DVB-T signal would use a substantial amount of battery life in a short time. Time slicing of the MPE-FEC frames is used to solve this problem [4]. The data is received in bursts so that the receiver, utilizing control signals, remains inactive when no bursts are to be received. A burst is sent at a significantly higher bitrate compared to bitrate of the media streams carried in the burst.

3.4. PACKET-ORIENTED REAL-TIME MEDIA TRANSPORT OVER MOBILE NETWORKS

A mobile network consists of three interacting domains, the core network (CN), the radio access network (RAN), and the user equipment (UE). The Third Generation Partnership Project (3GPP) is a joint effort of a number of standardization organizations. 3GPP develops the core network as an evolution of the Global System for Mobile Communications (GSM) core network, specifies radio access networks and user equipment, and defines systems and services for the 2.5G and 3G mobile systems (and beyond). 2.5G is an unofficial term used for the evolutions of the GSM technology, such as General Packet Radio Services (GPRS), providing packet-switched technology. The third-generation (3G) mobile technologies include the Universal Mobile Telecommunications System (UMTS) specified by 3GPP. In general, 3GPP multimedia services operate on both 2.5G and 3G mobile networks.

3GPP has developed two families of radio access technologies: GSM/EDGE radio access network (GERAN) based on time-division multiplexing and UMTS terrestrial radio access network (UTRAN) based on wideband code division multiple access (WCDMA). EDGE stands for enhanced data rates for GSM evolution. UMTS mobile stations can operate both in

circuit-switched (CS) operation primarily for conventional speech calls and packet-switched (PS) operation for IP-based services. For IP services, UMTS offers quality of service (QoS) classes for four types of traffic: conversational (voice and multimedia telephony), streaming, interactive (e.g., web browsing), and background (e.g., email and short message service). The QoS classes are characterized by and differ in maximum bitrate, guaranteed bitrate, maximum service data unit (SDU) size, residual error ratios, and transfer delay among other things.

This section is organized as follows: As UTRAN is one of the most important network environments for mobile multimedia and provides a good example of the operation of mobile radio networks, the characteristics of protocol layers 1 and 2 of UTRAN are introduced in Section 3.4.1. The 3GPP services that are most relevant for this thesis are packet-switched streaming service (PSS) [2] and multimedia broadcast/multicast service (MBMS) [3], which are reviewed in Sections 3.4.2 and 3.4.3, respectively.

3.4.1. UMTS Terrestrial Radio Access

The protocol stack for UTRAN includes the physical layer (also known as layer 1) and the link layer (also known as layer 2), which is further divided into three sub-layers, the medium access control (MAC) layer, the radio link control (RLC) layer, and the packet data convergence protocol (PDCP) layer. The operation of UTRAN is briefly described next using a simplified protocol stack illustrated in Figure 6 [P8]. In UMTS, each RTP/UDP/IP packet is treated as a service data unit (SDU) for PDCP. Thus, an RTP/UDP/IP packet and the PDCP header form a PDCP protocol data unit (PDU) and serve as an SDU for the radio link control (RLC) layer. The PDCP layer hides the differences of protocols in higher layers, such as IPv4 and IPv6, from the lower layers and provides services such as protocol header compression. An RLC-SDU is segmented into RLC protocol data units (RLC-PDUs), which are the elementary units to be transmitted over the physical layer. The length of RLC-PDUs varies typically in the range of 20 to 100 bytes depending on the selected radio bearer. The RLC layer provides logical links over the radio interface, i.e., it is capable of fragmentation and reassembly of higher layer packets (RLC-SDUs). The fragmentation and reassembly functionality is realized through sequence numbers (SNs) assigned for each RLC-PDU. There are three transmission modes in the RLC layer: transparent, unacknowledged, and acknowledged. The transparent mode provides only the fragmentation and reassembly function and is used mainly for the circuit-switched operation. The unacknowledged mode is capable of error detection in the receiving end, and erroneous RLC-SDUs are not passed to higher layers. This mode can be used for delay-constrained packet-switched applications, such as video telephony. The acknowledged mode provides link-layer retransmission of erroneous RLC-PDUs and it can be used for applications, such as streaming, that can tolerate a moderate amount of end-to-end delay. The MAC layer controls the access for the radio channel. The physical layer generally adds FEC parity bits to RLC-PDUs and protects their integrity with a CRC code. The transmission time interval (TTI) between two consecutive RLC-PDUs and the used modulation scheme determine the system delay and the bearer bitrate. More details on the UMTS radio access are provided in [36] and [129]. [P3][P8]

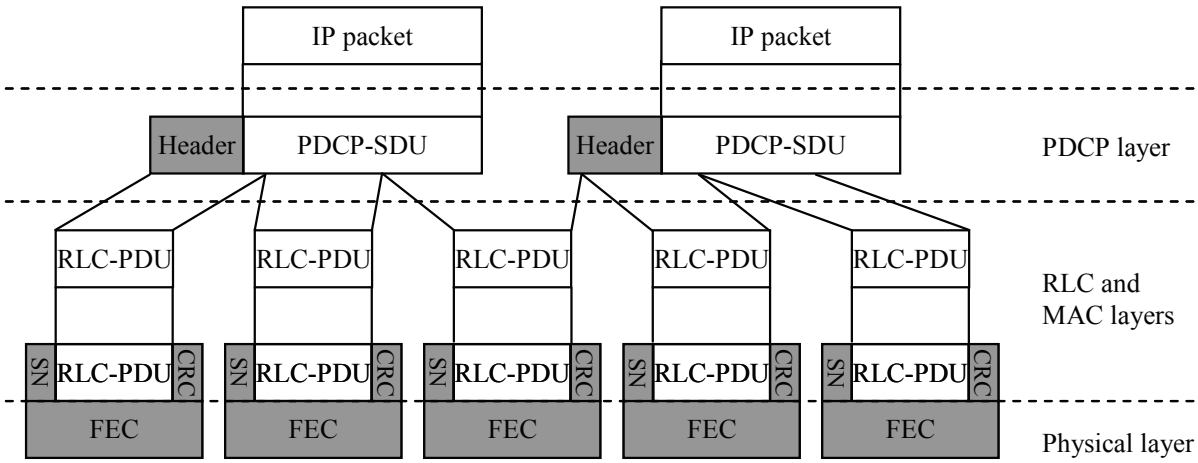


Figure 6. Simplified UTRAN protocol stack. Elements added by a protocol stack layer are indicated by gray background.

3.4.2. 3GPP Packet-Switched Streaming Service (PSS)

The PSS standard [2] specifies the framework, formats and protocols that can be used in the service. The PSS focuses on streaming of timed media, such as audio and video, but it also provides means for a rich multimedia presentation using synchronized multimedia integration language (SMIL) [139] and/or scalable vector graphics (SVG) [140]. An RTP-based protocol stack, very similar to the one presented in Section 3.2, is used in PSS for real-time media streaming. RTP/AVP is required in PSS, and RTP/AVPF is recommended. Additional recommended error resilience features of PSS include client buffer feedback signaling, used for bitrate adaptation and reviewed in Section 4.2, and RTP-layer retransmissions, reviewed in Section 4.4. Any radio access, such as UTRAN presented in Section 3.4.1, can be used with PSS, and usually the acknowledged mode is preferred in the radio link layer. In addition to the RTP-based transport, PSS also specifies the coding formats that are allowed in the service. At the time of writing this thesis, H.264/AVC Constrained Baseline is a recommended video format. A review of PSS is available in [44].

3.4.3. 3GPP Multimedia Broadcast/Multicast Service (MBMS)

MBMS can be functionally split into the bearer service [1] and the user service [3]. The MBMS bearer service specifies the transmission procedures below the IP layer, whereas the MBMS user service specifies the protocols and procedures above the IP layer. The MBMS user service includes two delivery methods: download and streaming. This section provides a brief overview of the MBMS streaming delivery method. A thorough overview of MBMS is available in [56].

The streaming delivery method of MBMS uses a protocol stack similar to the stack used for real-time media streaming in PSS. Due to the broadcast/multicast nature of the service, interactive error control features, such as retransmissions, are not used. Instead, MBMS

includes an application-layer FEC scheme for streamed media. The scheme is based on an FEC RTP payload format that has two packet types, FEC source packets and FEC repair packets. FEC source packets contain media data according to the media RTP payload format followed by the source FEC payload ID field. FEC repair packets contain the repair FEC payload ID and FEC encoding symbols, i.e., repair data. The FEC payload IDs indicate which FEC source block the payload is associated with and the position of the header and the payload of the packet in the FEC source block. FEC source blocks contain entries, each of which has a one-byte flow identifier, two-byte length of the following UDP payload, and an UDP payload, i.e., RTP packet including the RTP header but excluding any underlying packet headers. The flow identifier, which is unique for each pair of destination UDP port number and destination IP address, enables the protection of multiple RTP streams with the same FEC coding. This enables larger FEC source blocks compared to FEC source blocks composed of single RTP stream under the same period of time and hence may improve error robustness. However, a receiver must receive all the bundled flows (i.e., RTP streams), even if only a subset of the flows belongs to the same multimedia service.

The processing in the sender can be outlined as follows: An original media RTP packet, generated by the media encoder and encapsulator, is modified to indicate RTP payload type of the FEC payload and appended with the source FEC payload ID. The modified RTP packet is sent using the normal RTP mechanisms. The original media RTP packet is also copied into the FEC source block. Once the FEC source block is filled up with RTP packets, the FEC encoding algorithm is applied to calculate a number of FEC repair packets that are also sent using the normal RTP mechanisms. Systematic Raptor codes [91] are used as the FEC encoding algorithm of MBMS.

At the receiver, all FEC source packets and FEC repair packets associated with the same FEC source block are collected and the FEC source block is reconstructed. If there are missing FEC source packets, FEC decoding can be applied based the FEC repair packets and the FEC source block. FEC decoding leads to the reconstruction of any missing FEC source packets, when the recovery capability of the received FEC repair packet is sufficient. The media packets that were received or recovered are then handled normally by the media payload decapsulator and decoder.

Chapter 4

Error Resilience in H.264/AVC Video Communication

This chapter reviews the error resilience techniques for video communication applications using H.264/AVC. Communication protocols for real-time media transport typically attach a sequence number that is incremented by one for each transmitted packet, and therefore packet losses can be detected from a gap in the sequence number values of consecutively received packets. Error detection is therefore not discussed to a large extent. The chapter starts with a review of priority partitioning methods for unequal error protection in Section 4.1. The presented priority partitioning methods can be used with interactive and forward error resilience methods. Then, interactive error control methods, i.e., congestion control, interactive error concealment, and interactive error correction, are reviewed in Sections 4.2, 4.3, and 4.4, respectively. Forward error correction and concealment methods are discussed in Section 4.5. Section 4.6 provides an overview of methods for error concealment by post-processing. Lastly, Section 4.7 summarizes this chapter by giving recommendations which error resilience methods suit particular applications.

4.1. PRIORITY PARTITIONING FOR UNEQUAL ERROR PROTECTION

Video bitstreams can be partitioned to segments of different priorities according to their impact on subjective quality. Priority ranking can be used for different methods for unequal protection of coded data. For example, priority segments can be protected with unequal amount of FEC repair data (see Section 4.5.8), scheduled for transmission such that the most important pieces can be retransmitted if needed (see Section 4.2.2), or network congestion and packet losses can be avoided by omitting the transmission of the least important data (see Section 4.2.3). Priority partitioning methods can be roughly categorized into temporal segmentation, spatial and quality layering, data partitioning, and region-of-interest prioritization, which are briefly reviewed next.

4.1.1. Temporal Segmentation

Temporal segmentation according to the importance of pictures can be done based on the picture type (I, P, B) [87], or, more generally, based on classification into reference and non-reference pictures. The importance of a reference picture can be scored according to the number of subsequent pictures in decoding order that have direct or indirect prediction dependencies on the reference picture. In other words, pictures at the tail of a GOP are less important than pictures at the head of the GOP [20]. Non-reference pictures do not affect decoding of any other picture and are therefore least important subjectively.

The categorization into reference and non-reference pictures results into two priority classes, whose bitrates are governed by the used encoding settings. In other words, the number of priority classes is fixed to two and the respective bitrate shares of the two priority classes are inflexible. The usage of the picture distance from the start of a GOP as a priority partitioning criterion provides a greater number of priority classes but typically results into discontinuous video under error-prone transmission conditions. Hierarchical temporal scalability structures enabled by the sub-sequence coding technique presented in Chapter 6 enable a flexible number of priority classes and increase the likelihood of continuous video, when errors occur in transmission.

4.1.2. Spatial and Quality Layering

Some video coding schemes offer quality and spatial scalability [96]. Scalable video is typically ordered into hierarchical layers including a base layer, which contains an independent representation of a video sequence, and one or more enhancement layers, which contain quality or spatial refinement data. Quality scalability, also known as Signal-to-Noise Ratio (SNR) scalability, is based on the use of a finer quantization step in an enhancement layer compared to the base layer. There are three types of quality scalability, coarse granular scalability (CGS), medium grain scalability (MGS), and fine granular scalability (FGS) [120], out of which CGS and MGS are supported in SVC. In CGS and MGS, the order of coded parameters in an enhancement layer picture resembles that of the base layer picture, i.e., a macroblock is an atomic unit in the bitstream and macroblocks are arranged in raster scan order within coded slices. In FGS, the order of bits in a coded slice is generally a decreasing function of importance with respect to the subjective quality. In other words, coded data of macroblocks are interleaved, and an FGS slice can be truncated at any byte location. An MGS or FGS enhancement layer picture can be typically omitted with manageable drift in the decoded signal, while disposal of a CGS enhancement layer picture may cause severe degradation of quality. In spatial scalability, pictures in enhancement layers have a higher resolution compared to the base layer pictures. The layer hierarchy forms a natural basis for priority ordering of data. However, especially when a bitstream includes different types of scalability, priority ordering becomes a multi-dimensional rate-distortion optimization problem, as presented in [6].

4.1.3. Data Partitioning

There are two levels of data partitioning typically present in video coding schemes. First, sequence and picture level parameters that are required for decoding of the slice and macroblock level data are typically organized in their own structures. For example, in H.264/AVC, sequence and picture parameter sets are necessary for decoding of any picture data and are carried in separate NAL units from the slice data. Second, coded slice data is typically organized in macroblock scanning order, i.e., all data for a macroblock resides consecutively in the bitstream. However, many video coding schemes enable slice data partitioning, i.e., forming of bitstream structures that contain certain syntax elements across all macroblocks of a slice. For example, slice data partition A of H.264/AVC comprises macroblock headers and motion vectors of a slice, whereas slice data partitions B and C include the coded residual data for intra macroblocks and inter macroblocks, respectively. Encoders may generate slice data partitions instead of regular slices into the bitstream.

4.1.4. Region-of-Interest Prioritization

A picture can be partitioned into regions or objects of different subjective importance. Several methods have been proposed for region of interest determination in the literature, some of which are briefly reviewed next.

An example of the widely studied research topic of automatic face detection was provided by Eleftheriadis and Jacquin, who proposed face detection based on head outline and eye-nose-mouth detection from a binary-thresholded edge image [34]. Doulamis et al. proposed a neural network classifier, which uses the AC coefficients of Discrete Cosine Transform (DCT) blocks as feature vectors [31]. The classifier is first trained by ground-truth sequences that are hand-segmented to foreground and background.

Foveation-based image and video processing is based on the fact that the sampling density of the human visual system is higher at and close to the focus of the human eye, also known as the fixation point, compared to the rest of the vision. An eye tracker can be used to trace the gazing direction of the end-user and the gazing direction can then be utilized in video processing. Lee et al. [86] proposed to use foveation information for unequal error protection as follows. Each macroblock is mapped to “the foveated layer” or “the background layer”, which are then coded as slices using the independent segment decoding mode (see Section 4.5.1) and data partitioning. Automatic repeat-requests (see Section 4.4) and reference picture selection (see Section 4.3.2) make it possible to use only correctly decoded picture areas as references for inter prediction. Unequal error protection is obtained by inserting more resynchronization markers in the foveated layer compared to the background layer, i.e., having a smaller packet size in the foveated layer.

Im and Pearmain [60] proposed picture partitioning into two slice groups based on error concealment distortion. For each picture, the number of coded bits (the rate, R) and distortion (D_{coded}) are estimated first. Initial values of distortion for error-concealed macroblocks ($D_{concealed}$) are calculated by assuming that no data for the picture is received. Then, macrob-

locks with the highest difference between D_{coded} and $D_{concealed}$ relative to the rate (R) are iteratively selected to the first slice group. Values of $D_{concealed}$ are also updated iteratively assuming that the first slice group is correctly received. The iteration process is continued until the first slice group is of desired size in terms of number of bits. Dhondt et al. [30] used a similar scheme as Im and Pearmain but also considered the error concealment distortion in subsequent pictures.

Regions of different priority should reside in different coded data units in order to enable unequal error protection. Slices or slice groups can be used to enclose a particular region of interest. It is desirable that a region of interest enclosing a particular object would not be influenced by potential errors hitting other regions of interest or background. This desired characteristic is enabled by the isolated regions technique presented in Chapter 5. An isolated region is independently decodable and can enclose a region of interest that may change location and shape.

4.2. CONGESTION CONTROL IN UNICAST APPLICATIONS

Initial buffering in receivers smoothes away small fluctuations in throughput and transmission delay. When the data rate throughput change is such long in duration or big in bitrate that receiver buffers are in danger of under- or overflowing, senders should adjust the transmission bitrate to the available throughput. Sources of throughput changes and detection methods for them are reviewed in Section 4.2.1.

Abrupt and substantial throughput drops, such as those caused by cell handovers in mobile reception, pose a challenge to continuous media rendering. Even if senders react to detected throughput drops immediately by adjusting the transmission bitrate, client buffer may drain in the meanwhile and cause discontinuous playback. In order to counter throughput drops proactively, robust packet scheduling methods can be used, as discussed in Section 4.2.2.

When congestion in fixed packet-switched network is detected, senders are often expected to behave similarly to the congestion control algorithm of the Transmission Control Protocol (TCP), which adjusts the sending rate by an Additive Increase Multiplicative Decrease (AIMD) rate control algorithm [18]. Additive increase refers to the fact that the transmission rate is raised incrementally if no packet losses are detected, and multiplicative decrease refers to halving the transmission rate once every round-trip time as a response to a packet loss. It is obvious that when a sender encodes video in real-time and there is only one receiver, the bitrate control algorithm of the encoder can be used for data rate adjustment. Otherwise, methods manipulating coded bitstreams, such as stream thinning and switching, should be used. A brief review of stream thinning and switching methods is given in Section 4.2.3.

4.2.1. Sources and Detection of Throughput Changes

There are three main sources for throughput drops in streaming over heterogeneous networks: congestion in a best-effort network, decrease of a service class in networks providing guaranteed quality of service (QoS), and cell handovers in mobile networks. Each type of a throughput drop requires specific handling, even though the underlying processing techniques remain the same for all.

RTCP receiver reports provide some means for detecting throughput changes. When congestion is the only source of packet losses, the packet loss counters included in RTCP receiver reports give an indication of congestion. RTCP receiver reports also include the interarrival jitter field, which represents the statistical variance of packet interarrival times relative to RTP timestamps. Increased interarrival jitter can be of a sign of congestion and enables the sender to react before packet losses happen. [117]

Many of the currently used packet-switched mobile networks, such as GPRS, offer a best-effort service. The use of RTCP receiver reports for throughput change detection is unreliable in best-effort mobile networks, because the wireless link is an additional source of packet losses. Thus, senders cannot conclude based on RTCP receiver reports whether an increased packet loss rate is due to congestion or more severe radio conditions. Moreover, the use of the acknowledged mode in the link layer may cause substantial end-to-end delay variation, which makes methods based on interarrival jitter inoperable. [36]

In order to provide means to differentiate between congestion and harsh radio conditions, an RTCP extended report with receiver buffer status indications, also known as RTCP APP packet with client buffer feedback (NADU APP packet), has been specified in the 3GPP packet-switched streaming service [2]. The signaling enables the sender to reconstruct the exact status of the receiver buffer and derive the delay prevailing in the network. Reviews of the bitrate adaptation feature of 3GPP packet-switched streaming are provided in [26] and [27].

An example of signaling for a throughput change when guaranteed QoS is in use can also be found in the 3GPP packet-switched streaming service [2], which specifies the 3GPP-Link-Char header to be used with certain RTSP methods for signaling of the guaranteed radio link bandwidth.

Cell handovers may cause data rate throughput drops that are typical for mobile networks. When it comes to the detection of a handover in the application layer of real-time multimedia applications, Kampman and Baldo proposed to use the NADU APP packet for handover detection [77]. While the proposed technique detects handovers accurately, the detection can only operate after the handover is over, as no RTCP extended report is conveyed during the handover. Therefore, Bouazizi et al. proposed handover detection based on the expected reception interval of RTCP receiver reports [14].

4.2.2. Robust Packet Scheduling

Robust packet scheduling techniques are applicable when recipients buffer data initially before the start of media playback. The techniques can be applied for minimizing the impact of

abrupt network throughput changes, such as cell handovers, and for selecting an optimal truncation path for stream thinning. The idea of robust packet scheduling is as follows: Coded slices and coded slice data partitions that are subjectively the most important are sent earlier than their decoding order indicates, whereas coded slices and coded slice data partitions that are subjectively the least important are sent later than their natural decoding order indicates. Consequently, any temporary decrease in the network throughput might cause the least important data to arrive too late for decoding while sufficient amount of the most important data would be readily buffered in the receiver to compensate for the throughput drop. Moreover, if a piece of the most important data is lost during transmission, it is more likely that the piece could be retransmitted and received before its scheduled decoding or playback time compared to the least important data. [54]

The algorithms for robust packet scheduling can be categorized into two classes: heuristic and rate-distortion-optimized algorithms. Two examples of the heuristic algorithms are given in this paragraph, whereas the rate-distortion-optimized algorithms are introduced in the next paragraph. Miao and Ortega proposed an algorithm called Expected Runtime Distortion Based Scheduling (ERDBS) [93][94], which derives the importance and scheduling of each packet in run-time based on the scalability layer which the packet contains data for, the fact whether any data dependent on the packet has already been transmitted and received correctly, and the probability of the packet to reach the destination before the decoding and playback time of the packet. Kang and Zakhor [78] proposed an algorithm that assigns an importance factor to packets and pictures carried in the packets as an increasing function of the picture decoding order relative to the previous intra picture starting a GOP. Packets are sent in increasing order of the importance factor within a certain time window relative to the current playback time in the recipient. The algorithm is integrated with feedback from the recipient indicating which packets have been correctly received. Moreover, the algorithm can be applied together with data partitioning by weighting motion and texture packets differently. In their later work [79], Kang and Zakhor developed a rate-distortion-optimized version of the heuristic algorithm [78]. The algorithm uses statistical video traffic models and considers channel bitrate fluctuations.

Chou and Miao developed rate-distortion-optimized algorithms for robust packet scheduling in streaming applications in their technical report [19], later published as a journal paper [21]. The algorithm optimizes the use of time and bandwidth resources by minimizing a Lagrangian rate-distortion cost function. The paper concludes that rate-distortion-optimized streaming of an entire presentation can be solved by focusing on the error-cost optimized transmission of a single data unit in isolation. The presented algorithm can be applied to various streaming scenarios, including sender-driven and receiver-driven, with and without feedback, with and without retransmission and forward error correction, as well as best-effort and guaranteed quality of service. The work of Chou and Miao has been used as the basis for several enhancements and extensions to new applications, such as considerations for multiple playback deadlines and accelerated retroactive decoding [76], multi-path transmission [17], and rate-congestion optimization considering congestion avoidance on bottleneck links [121].

In order to facilitate robust packet scheduling, the transmission system has to provide means for sending data out of its decoding order and recovering the data decoding order in receivers. The RTP sequence number is required to be incremented by one for each sent RTP packet, therefore providing the capability of recovering the transmission order of packets in receivers. However, RTP includes no mechanism for recovery of a correct decoding order if data is not transmitted in decoding order. A mechanism for the RTP payload format of H.264/AVC allowing any data transmission order and recovery of the correct decoding order in receivers is presented in Section 6.2.2. The presented mechanism includes signaling enabling receivers to allocate a data reordering buffer having a sufficient size and applying a correct amount of initial data buffering. Section 6.3.1 discusses how robust packet scheduling can be applied for H.264/AVC.

4.2.3. Stream Thinning and Switching

Stream thinning refers to omission of certain coded data units, such as non-reference pictures and the least important scalability layers, from the transmitted stream. Even non-scalable bitstreams can be thinned as explained next. A known method in current streaming systems to cope with drastically dropped channel throughput is to transmit intra-coded pictures only. When the network throughput is restored, inter-coded pictures can be transmitted again from the beginning of the next GOP. Generally, any chain of inter-coded pictures can be safely disposed, if no other picture is predicted from them. Consequently, inter-coded pictures at the tail of a GOP can be removed without affecting the decoding of any previous or subsequent picture. In general, priority partitioning methods, reviewed in Section 4.1, can be used to select the order according to which parts of the bitstream are omitted from the transmitted stream when the channel throughput is not sufficient. The sub-sequence technique, presented in Chapter 6, can be used to increase the number of priority classes and hence provide finer bitrate adaptation steps compared to what can be achieved with conventional non-hierarchical temporal scalability.

If stream thinning does not provide a big enough dynamic range for bitrate adjustment, the server should switch to a different version of the same content coded for a bitrate that is close to the network throughput. Switching to a different bitstream can naturally be done at any random access point. In order to respond to a need for adjusting bitrate faster and avoid the compression penalty of frequent intra pictures, there have been studies how stream switching could be done starting from non-intra pictures. Färber and Girod proposed S frames that are inter-coded frames used only when switching from a first stream to a second stream [39]. S frames are encoded with a small quantization step and make the decoded S frame close but typically not identical to the corresponding decoded picture of the second stream. H.264/AVC includes the feature known as SI/SP pictures [80], which can be used similarly to S frames but provide identical decoded picture after switching compared to decoding of the stream from the beginning. Identical decoded pictures are obtained with the cost of additional transform and quantization steps in the decoding process for SI/SP pictures both in the primary streams

and SI/SP pictures used for switching only. However, the SI/SP feature is not included in the Baseline or High profile and therefore not commonly used.

4.3. INTERACTIVE ERROR CONCEALMENT

In conversational video communications systems, such as video telephony, there is usually a feedback channel from the receiver to the sender. The feedback channel can be utilized for recovery from transmission errors among other things. Interactive error concealment messages from the receiver to the sender can be categorized into intra update requests, loss indications, and positive acknowledgements of correctly received and decoded data. The encoder can respond to the messages by intra coding or encoding using only those reference pictures that are correct in content. The encoder can also further improve compression efficiency and completeness of error correction, if it tracks the propagation of the indicated errors, recovers also those areas that are damaged by error propagation, and uses only undamaged areas as references for inter prediction.

This section reviews the literature and the standards related to interactive error control for low-latency video communication. The section is organized as follows: Section 4.3.1 discusses intra update requests and picture loss indications. Section 4.3.2 summarizes the essentials regarding interactive reference picture selection. The error tracking methods for encoders are briefly outlined in Section 4.3.3.

4.3.1. Intra Update Requests

A simple method for recovery from transmission errors is to request the far-end encoder to encode the erroneous areas in intra coding mode. In addition to recovery from transmission errors, the fast update picture command can be issued by the multipoint control unit (MCU), when there is a need to switch from one video originator to another in centralized multipoint conferencing or a new end-point joins a conference. Because the response of an encoder to an indication of lost data is not necessarily the same as the response to a request for a random access point, the codec control messages extension [156] of RTP/AVPF includes a picture loss indication and a full intra request command separately. A full intra request command is also included in RTP/AVPF [97]. Encoders may react to a picture loss indication by transmitting an IDR picture or by intra coding the picture area gradually, i.e., in a number of consecutive pictures [64]. When gradual recovery is used, a recovery point SEI message of H.264/AVC is sent to indicate when the entire picture area is correct in content. The gradual recovery procedure may be used in error-prone transmission environments in which an IDR picture would be likely to be hit by transmission errors due to its large size relative to a typical inter picture.

When a gradual recovery process is applied, the encoder must not use any non-refreshed area as an inter prediction reference for those areas that are being refreshed in the picture being encoded. Methods for this kind of spatio-temporal limitation of inter prediction

are utilized in the isolated regions technique presented in Chapter 5, and Section 5.3 summarizes the use of isolated regions in gradual decoding refresh.

4.3.2. Interactive Reference Picture Selection

Intra coding resulting from picture loss indications reduces compression efficiency compared to inter coding. In order to improve the compression efficiency, an encoder can choose such a reference picture for inter prediction that is known to be correct and available based on the feedback from the far-end decoder. This technique was first proposed in [46] and is often referred to as NEWPRED. The technique requires that the video coding scheme allows the use of multiple reference pictures, and hence H.263 Annex N, H.263 Annex U, and H.264/AVC can be used, for example. There are two types of feedback messages: negative acknowledgements (NACKs) indicating that a certain packet or a certain picture or certain areas of a particular picture were not received correctly and positive acknowledgements (ACKs) indicating which pictures or parts of pictures were correct. When negative acknowledgements are in use, the encoder typically uses any available reference picture for inter prediction except for those that are known to be erroneous based on the received NACK messages. Due to the fact that the end-to-end delay may be greater than the interval between two encoded pictures, the encoder may not know that some of the recently encoded reference pictures are not received correctly at the time of encoding a new picture. Thus, the NACK mode of NEWPRED stops error propagation in about one round-trip time similarly to the fast update requests. When positive acknowledgements are in use, the encoder typically uses only those reference pictures for inter prediction that are known to be correct based on the received ACK messages.

ITU-T Recommendation H.271 [72] specifies generic back-channel message syntax for use with any video codec. Six messages are specified: an indication that one or more pictures are decoded without detected errors, an indication that one or more pictures are entirely or partially lost, an indication that all or certain data partitions of a set of coding blocks of one picture are lost, a cyclic redundancy check (CRC) value for one parameter set, a CRC value for all parameter sets of a certain type, and a reset request indicating that the far-end encoder should completely refresh the transmitted bitstream as if no prior video data had been received. The semantics to identify a picture, the size of the coding block in terms of samples, and the definition of parameter sets are specific to the coding format, and H.271 specifies the semantics of the generic message syntax for H.261, H.263, and H.264/AVC. The codec control messages extension of RTP/AVPF includes a video back-channel message that carries messages according to H.271.

4.3.3. Error Tracking

Error tracking refers to encoder's reconstruction of temporal and spatial propagation of errors in the far-end decoder. For example, if frame n is damaged and the corresponding back-channel feedback message arrives in the encoder when it is time to encode frame $n+d$, the encoder reconstructs the location of damaged areas in frames n to $n+d-1$ in the decoder based on

the motion vectors in frames *n+1* to *n+d-1*. The encoder can then avoid using any of the damaged areas in frames *n* to *n+d-1* for inter prediction. Examples of an error tracking algorithm are provided in [126] and H.263 Appendix I [67].

Error tracking can be further refined if the feedback messages contain information which error concealment method the decoder used or the error concealment method has been pre-determined in the system. As a response to receiving a feedback message concerning frame *n*, the encoder must reconstruct the decoding process for frames *n* to *n+d-1* exactly so that the reference pictures at the encoder match the reference pictures in the decoder accurately. Joint error and error concealment tracking was first proposed by Wada [141] and studied in the context of H.263 by Girod and Färber [47].

4.4. INTERACTIVE ERROR CORRECTION

Interactive error correction, also referred to as retransmission, is known to be a powerful technique for error recovery when the buffering latency in the receiver is greater than one retransmission interval. Retransmission can be made universally to any lost or corrupted data or selectively based on playback deadlines, for example. ARQ (Automatic Repeat reQuest) refers to an error control method for data transmission in which the receiver detects transmission losses or errors and automatically requests a retransmission from the sender. As a response to an ARQ message, the sender retransmits the requested piece of data until it is either correctly received or the error persists beyond a predetermined number of retransmissions. Several ARQ protocols have been proposed differing in allowed window size for transmission ahead of received acknowledgements and sender reaction to missing acknowledgements among other things. ARQ protocols can be implemented in different layers in the protocol stack. For example, the UMTS provides an acknowledged data transfer service in which link layer packets (RLC-PDUs) are automatically retransmitted, and TCP ensures correct reception of data by an ARQ procedure. However, automatic retransmission falls outside of the scope of this thesis and therefore only selective retransmission is considered in this section. In the following, a few selective retransmission protocols and methods which are relevant to the thesis are considered.

An RTP payload format for retransmission of any media data conveyed over RTP is specified in [106]. The payload format requires the use of the audio-visual RTP profile with feedback (RTP/AVPF, see Section 3.2). The generic NACK message of RTP/AVPF serves as a retransmission request. The retransmitted packets are delivered in a separate RTP session compared to the original media data. Alternatively, both retransmitted packets and original media packets are transmitted in the same RTP session but the synchronization source (SSRC) field of the RTP header is used to distinguish the type of packets. The values of the most relevant RTP header parameters of the retransmitted packets are identical to the original packet except for the sequence number, and the payload header of the retransmitted packets consists of the original sequence number of the retransmitted packet.

Retransmission can be used for conversational video applications, such as video telephony, when the receiver does not wait for the lost packets, pictures or slices to arrive after sending a retransmission request for them. Instead, the receiver conceals the losses and continues decoding. When the retransmitted data arrives, the receiver reverts to the decoder state which was active when the retransmitted data was supposed to be decoded the first time. Then, the decoder decodes the retransmitted data and any dependent subsequent data again to obtain correct reference pictures for inter prediction of subsequent pictures. This decoding process must happen faster than real-time in order to avoid any latency and hence the technique can be referred to as retroactive accelerated decoding. Error tracking can be used to redecode only those slices that are affected by the retransmitted data. The technique was first proposed by Zhu in [173] and summarized in [143].

Rhee and Joshi proposed a hybrid retransmission and forward error correction scheme called RESCU (recovery from error spread using continuous updates) for conversational video applications [74][107][108]. The RESCU algorithm is based on a reference picture selection scheme in which periodic frames are coded in pre-determined intervals, each periodic frame is predicted only from the previous periodic frame, and other frames are predicted from any available reference frames that do not precede the previous periodic frame in decoding order. The periodic frames can be protected with retransmissions or FEC or both. Retransmission is only requested if a periodic frame is lost because it is likely that a retransmitted periodic frame is received before the next periodic frame is decoded. Similarly, it is likely that a sufficient amount of FEC repair data for a periodic frame is received before the next periodic frame should be decoded. As with Zhu's technique [173], RESCU requires restoring of the decoder state and accelerated decoding when retransmitted or repaired periodic frame is decoded.

4.5. FORWARD ERROR CORRECTION AND CONCEALMENT

Forward error correction and concealment techniques applicable to H.264/AVC include limitation of in-picture prediction, intra coding, limitation of inter prediction, redundant coded pictures, multiple description coding, and assisted error concealment, which are reviewed in Sections 4.5.1 and 4.5.3 to 4.5.7, respectively. Efficient use of methods limiting in-picture prediction requires knowledge from other protocol stack layers, and hence cross-layer optimization of in-picture prediction limitation is discussed separately in Section 4.5.2. Error-robust entropy coding methods also fall into the category of forward error correction and concealment but are not considered, as they apply to bit-error-prone environments, which are not considered in this thesis for reasons mentioned in Section 3.1. Forward error correction and concealment may be applied equally over the entire stream. Alternatively, the stream may be divided into parts of different importance to subjective quality, as reviewed in Section 4.1, and those parts can be protected unevenly. Some unequal error protection schemes for forward error correction and concealment are presented in Section 4.5.8.

4.5.1. Constrained In-Picture Prediction

Slices and slice groups are the elementary coding structures for limiting in-picture prediction, as already reviewed in Section 2.3. This section provides some more details about their use for error resilience.

A coded slice is the basic mechanism for limiting in-picture prediction. No prediction of coding parameters happens across a slice boundary. Consequently, a slice can be decoded even if a spatially neighboring slice is not received or decoded. In H.264/AVC, deblocking loop filtering can be applied across slice boundaries, which could potentially cause a leak of errors from an incorrectly decoded or concealed slice to a correctly decoded slice. However, in practice, such a leak is often imperceptible and encoders can also turn the deblocking loop filter off at slice boundaries.

The slice group mechanism of H.264/AVC provides a flexible means for limitation of in-picture prediction. Wenger and Horowitz proposed scattered slice group ordering such that all adjacent macroblocks reside in different slice groups [154]. If a slice is lost, it is probable that the slices containing the adjacent macroblocks for the lost macroblocks are received, and hence error concealment is expected to perform more satisfactorily compared to the error concealment for slices containing macroblocks in raster scan order. However, scattered slice groups also drop compression efficiency due to the fact that motion vector and intra prediction are not done across macroblock boundaries.

A similar, although more restricted, mechanism to slice groups, known as the independent segment decoding mode, was included in H.263 (H.263 Annex R). In this optional mode, all slice boundaries are treated as picture boundaries, and therefore no spatio-temporal error propagation over slice boundaries occurs. When the mode is in use, all slices have to be rectangular and the locations of slice boundaries have to remain unchanged within a GOP, which limits the applicability of the independent segment decoding mode. A rectangular slice may be higher than one macroblock row and narrower than the entire picture width. Due to restricted motion prediction, compression efficiency drops compared to normal slice-based operation. Furthermore, because the number of macroblocks in a slice is constant within a GOP, the slice size cannot be adjusted according to an optimal packet size for the prevailing network conditions. The shortcomings of rectangular slices and the independent segment decoding mode can be overcome with rectangular-oriented and evolving slice groups used together with the isolated regions technique presented in Chapter 5.

In addition to the use of slices and slice groups, the constrained intra coding mode of H.264/AVC should be used in error-prone environments. In the constrained intra coding mode, intra prediction is allowed only from intra-coded neighboring macroblocks and hence there is no error propagation from incorrectly decoded inter macroblocks to intra macroblocks. More information on intra prediction in H.264/AVC is provided in Section 2.2.

4.5.2. Cross-Layer Optimization for In-Picture Prediction Limitation

Encoders should adjust the use of the tools for in-picture prediction such a way that sufficient level of error resilience is obtained and compression efficiency is not decreased unnecessarily. Such a sophisticated encoder operation typically requires cross-layer optimization and information about the prevailing channel conditions. Two aspects of cross-layer-optimized limitation of in-picture prediction are reviewed in this section: the size of slices in terms of bytes and the transmission order of slices. The size of slices can be selected according to the physical layer packet size or expected packet loss ratio. More details of slice size selection and transmission ordering of slices are given next.

Some studies, such as [122] and [162], have suggested matching application-layer packets exactly to physical layer packets. The aim of these studies is to set the application-layer packet size to such that it allocates an integer number of physical layer packets. Consequently, a corruption of one physical layer packet never causes damage to more than one application-layer packet. If the transmission scheme uses time-division multiplexing or time-slicing, exact matching of application-layer packet to physical layer packets may also reduce end-to-end delay. Challenges for the exact slice size matching include temporal and spatial quality variation due to exact bitrate control, impact of varying size headers such as robust header compression of RTP/UDP/IP, and complicated signaling to indicate the physical layer constraints of remote links. Given the challenges in exact slice size matching, it is often sufficient to match the packet size to a certain range, which may, for example, avoid fragmentation to multiple link-layer packets, give a reasonably low packet header overhead, and suit the FEC matrix size. An example of approximate slice and application-layer packet size optimization was provided for DVB-H environment in [92]. Some studies, such as [128], have also questioned whether the use of slices is useful especially in relatively low bitrates and transport environments with long-interleave FEC or link layer retransmissions. Stockhammer analyzed different options for slice sizes for UMTS radio bearers extensively in [129].

It is intuitive that the higher the packet loss rate is, the smaller the packet and slice size should be in order to limit the impact of an individual packet loss and provide better chances for error concealment to succeed. However, the correlation of successive packet losses and the packet header overhead also affect the choice for an optimal packet size. Selection of a packet and slice size in bytes according to expected packet loss rate has been proposed in [45].

Some studies suggest that transmission order should be carefully selected for the most efficient use of slices and slice groups. When arbitrary slice ordering (ASO) is in use, decoders are required to accept slices of a picture in any order, and encoders and transmitters can send slices of a picture in any order. For example, ASO can be used to encapsulate one macroblock line to one slice and interleave macroblock lines in transmission order. When a slice is lost during transmission, it is likely that slices above and below it are received correctly and can be used for error concealment. However, when ASO was tested in a fixed IP network environment, no significant improvement was discovered compared to transmission in raster scan order [155].

Varsa and Karczewicz extended the idea of arbitrary slice ordering to construct packets containing slices from multiple pictures in an interleaved manner [137]. In their scheme, a packet contains no spatially adjacent slices of the same picture or co-located slices of temporally adjacent pictures, hence further improving the likelihood of successful error concealment from the spatial and temporal neighbors of lost or corrupted data. Ndili and Ogunfunmi essentially combined the use of scattered slice groups and multi-picture slice interleaving [95]. In their scheme, a packet contains no spatially adjacent macroblocks of the same picture or co-located macroblocks of temporally adjacent pictures. When slices from multiple pictures are transmitted in an interleaved manner, a correct slice decoding order has to be recovered in receivers. As RTP does not provide such a deinterleaving mechanism, the RTP payload format of H.264/AVC was designed to include an interleaved packetization mode, which is presented in Section 6.2.2.

4.5.3. Intra Coding

Incorrectly decoded picture data is propagated to subsequent pictures due to inter prediction. It is therefore obvious that intra coding can be used to stop temporal error propagation. In addition to intra picture coding, error-robust macroblock mode selection (a.k.a. adaptive intra macroblock refresh) algorithms can be used. They aim at refreshing the most error-prone areas as intra-coded macroblocks to avoid drastic visible errors and can be categorized into non-adaptive and adaptive algorithms. Adaptive methods can be further classified into simple cost function algorithms and rate-distortion-optimized methods.

Non-adaptive intra refresh algorithms typically use a mapping between the packet loss rate and the refresh frequency but apply intra coding uniformly across the picture area. One example of a non-adaptive intra refresh method is the periodical intra refresh algorithm that codes a certain number of intra macroblocks per picture in a pre-defined scan order. Another example of a non-adaptive algorithm is to code a certain number of macroblocks in intra mode at randomly selected macroblock locations [24].

Adaptive macroblock mode decision methods select the intra-coded macroblock locations in a way that the content of the pictures is taken into account. For example, a static background area needs not be refreshed in intra mode as often as moving objects. Simple cost-function-based methods, such as [90] and the adaptive intra refresh method proposed in Annex E of MPEG-4 Visual [62], calculate a cost for each macroblock with a certain function that may take into account the amount of prediction error data after motion compensation, for example. Intra coding is used for a certain number of macroblocks having the highest cost.

Rate-distortion-optimized macroblock mode selection algorithms estimate the end-to-end distortion, including both the distortion resulting from waveform coding and the distortion caused by transmission errors. Typically, a Lagrangian cost function that linearly combines “rate” and “distortion” is used, and the mode selection of each macroblock is such that the cost is minimized. The rate-distortion-optimized mode selection algorithms can be categorized into two categories: optimal per-pixel estimation and model-based methods, which are reviewed in more details below. The computational complexity of rate-distortion-optimized

macroblock mode selection algorithms is typically multifold compared to non-adaptive and simple cost-function-based algorithms.

Optimal per-pixel distortion estimation methods aim at computing the expected distortion at pixel level. One of the most well-known algorithms in this category is the recursive optimal per-pixel estimate (ROPE) algorithm [166], which computes the expected mean-squared error (MSE) by recursively calculating the first and second moment of each pixel. The original ROPE algorithm operates at full-pixel precision, and therefore it has been extended in [88] and [163] to address cross-correlation between pixels for more accurate distortion estimation of sub-pixel-accurate inter prediction. Moreover, the algorithm proposed in [88] extended the ROPE algorithm to consider one long-term reference picture. Other extensions of the ROPE algorithm include refinement of the distortion estimation based on feedback information from the recipient [165] and look-ahead to subsequent pictures in encoding order for non-real-time encoding [167].

Model-based macroblock mode selection algorithms use approximations of the end-to-end distortion. A straightforward approach for calculating the average expected distortion is to run several decoders, each for a different packet loss pattern, at the encoder and to average the resulting distortions [127]. This straightforward algorithm, herein referred to as the loss-aware rate-distortion-optimized (LA-RDO) macroblock mode selection algorithm, was also accepted to the Joint Model reference implementation of the H.264/AVC codec [131]. Although the LA-RDO mode selection algorithm estimates of the expected distortion reasonably accurately when the number of simulations is high enough, the drawback is that the computational complexity and storage requirements are impractical for many software and hardware platforms. Another model-based method was reported in [168]. In this method, a potential error propagation distortion is estimated without running multiple decoders – thus, the computational complexity is lower compared to the LA-RDO algorithm.

While most loss-aware macroblock mode selection algorithms try to minimize the expected distortion in the receiver, the Variance-Aware Per-Pixel Optimal Resource Allocation (VAPOR) algorithm [33] also accounts for the variance of the distortion, therefore increasing the likelihood that the decoded picture quality resembles the mean end-to-end distortion calculated at the transmitter.

It is noted that the interpolation filtering used to obtain sample values at sub-pixel locations in the inter prediction process should be taken into account in encoders when selecting motion vectors in an error-robust manner. For example, even if an inter prediction block is within an intra-coded macroblock, the sample values of an adjacent macroblock may affect the prediction block that is located at a sub-pixel location. Consequently, if the adjacent macroblock were erroneous, the error would propagate to the prediction block. This characteristic feature of the inter prediction process is unfavorable for the use of distinct intra-coded macroblocks for error robustness. In the isolated regions technique presented in Chapter 5, a number of adjacent macroblocks are selected to be intra-coded in each picture and motion vectors are constrained to avoid spatio-temporal error propagation. Furthermore, it is shown in Sec-

tion 5.4 that the isolated regions technique can be combined with the LA-RDO method further improving the performance of both methods applied individually.

4.5.4. Constrained Inter Prediction

Inter prediction can be limited in terms of the number of inter-dependent pictures and the depth of inter prediction dependencies of individual blocks. It is clear that the GOP structure and temporal scalability hierarchy affects the length of inter picture prediction chains. Loss-aware macroblock mode selection algorithms, reviewed in Section 4.5.3, try to optimize the coding mode on macroblock basis, thus limiting the inter prediction. There have also been studies, such as [15][55][158], which propose loss-aware selection of reference pictures in a similar manner compared to the coding mode selection. The use of bi-prediction for error-robust inter prediction was studied in [82]. Moreover, some of the video error resilience methods that are reviewed in other sections actually make use of constrained inter prediction chains. For example, many feedback-driven encoding algorithms select the reference pictures for inter prediction based on the received feedback as presented in Section 4.3.2. Division of coded data to different importance classes for unequal error protection can be based on temporal scalability layers as described in Section 4.1. Video redundancy coding, in which a sequence of pictures is divided into two or more independently coded inter-prediction threads, is an example of multiple description coding and therefore reviewed Section 4.5.6. Video redundancy coding and other methods using more than one inter-prediction thread typically suffer from a decreased compression efficiency compared to conventional non-hierarchical coding structures and hierarchical temporal scalability presented in Section 6.1.3. However, the intra picture postponement method presented in Section 6.3.3 uses two inter-prediction chains without a penalty in compression efficiency.

4.5.5. Redundant Coded Pictures

As discussed in Section 2.6.2, zero or more redundant coded pictures can be included in an access unit of H.264/AVC. The syntax and semantics of a redundant coded picture are identical to those of a primary coded picture. However, a redundant coded picture may contain a subset of the macroblocks of an entire picture, and different values of coding parameters, such as macroblock modes and reference pictures, can be used when compared to the primary coded picture. Decoders should not decode redundant coded pictures when the corresponding primary coded picture is correctly received and can be correctly decoded. However, when a primary coded picture is lost or cannot be correctly decoded, a redundant picture can be utilized to improve the decoded video quality.

Thanks to the flexibility of encoding redundant coded pictures adaptively and with any encoding parameters, a number of encoding methods for redundant coded pictures have been proposed. A method for unequal error protection based on redundant coded pictures was proposed in [147]. In this method, the encoder creates “key” pictures periodically, such as once every second. A “key” picture is either intra-coded or predicted from the previous “key” pic-

ture. Each "key" picture is protected by coding a respective redundant coded picture as an exact copy of the "key" picture. A method for coding redundant coded pictures using earlier reference pictures than those of the respective primary coded pictures was proposed in [170]. Additionally, the paper included a scheme for hierarchical placement of redundant coded pictures and their reference pictures. The allocation of redundant coded pictures was developed further in [171], which proposed an adaptive rate-distortion-optimized algorithm for coding of redundant coded pictures. A comprehensive study including all the methods of [170] and [171] was provided in [172].

4.5.6. Multiple Description Coding

A multiple description coder produces many independent streams, known as descriptions, from one original signal. Each description typically has similar importance, any one of the descriptions is sufficient to reproduce a decoded signal of basic quality, and the reproduction quality improves as a function of received descriptions. It is therefore evident that descriptions are correlated and Multiple Description Coding (MDC) has a penalty in compression efficiency compared to single description coding. The correlation may also enable the decoder to conceal missing descriptions. A number of algorithms have been proposed for multiple description coding, utilizing spatial, frequency, or temporal domain division. As only temporal division to descriptions is applicable to H.264/AVC as such, other types of multiple description coding are not reviewed. For a comprehensive review of all types of MDC algorithms, readers are advised to refer to [145].

Temporal-domain multiple description coding was introduced by Wenger in his method known as Video Redundancy Coding [149], but similar work by Apostolopoulos, known as multiple state video coding (MSVC) [7], may be more well-known in the MDC literature. Temporal-domain decomposition is based on encoding several independent and temporally interleaved picture prediction threads from the original signal in a round-robin manner. For example, two prediction threads, illustrated in Figure 7, can be formed by always selecting the picture preceding the previous picture as reference for inter prediction.

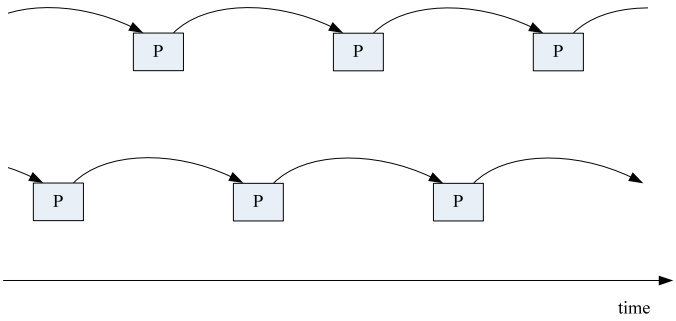


Figure 7. Video redundancy coding (VRC) or multiple state video coding (MSVC) with two prediction threads.

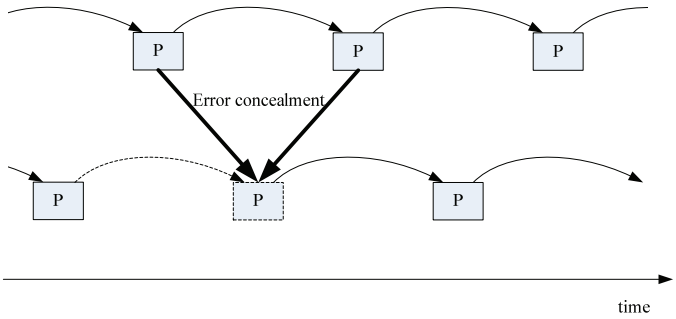


Figure 8. Error concealment using neighboring pictures from the received description in MSVC.

If one prediction thread becomes corrupted by transmission errors, the remaining prediction threads can still be decoded correctly. To stop potential temporal error within a thread, an original picture can be coded multiple times, each time from a different thread, although coding of such synchronization pictures is disadvantageous in terms of compression efficiency. Temporal-domain MDC enables error concealment of corrupted pictures from correctly decoded picture threads. An acceptable error concealment quality may be obtained when a damaged picture is concealed in a bi-directional manner from one or many undamaged threads similarly to the direct mode of bi-predictive pictures [7][53]. This is illustrated in Figure 8, where the picture enclosed within a dash line is lost in transmission and bi-directionally concealed from the surrounding pictures.

In the MSVC-RP algorithm [103], redundant coded pictures are provided for improving the error concealment of damaged pictures. An example with two prediction threads, A and B, is illustrated in Figure 9 to describe the algorithm. A redundant coded picture (RP) of thread A is temporally aligned with a primary coded picture of thread B but uses only the previous picture of thread A as the reference for inter prediction. Hence, if a primary coded picture of thread B is corrupted, a respective redundant coded picture can be decoded based on correctly received thread A. While redundant pictures are usually coded with a greater quantization step size than respective primary pictures, their use in error concealment is beneficial due to the fact that typical artifacts for error concealment algorithms can be avoided.

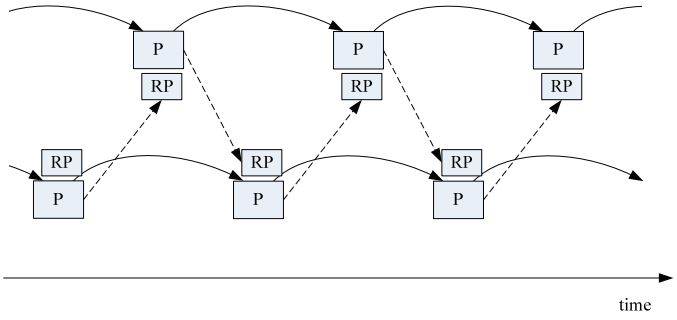


Figure 9. Illustration of MSVC-RP containing redundant coded pictures (RP) and two prediction threads.

4.5.7. Assisted Error Concealment

Techniques that the source coder or the source and transport coder jointly can use to improve the result of error concealment by the receiver are reviewed in this section. There are two types of methods falling into this category. First, there are methods ensuring the reception of the vital high-level information required for the decoding of coded picture data. Second, a class of methods adds auxiliary information to assist in error concealment. These methods are reviewed next.

It is essential to provide the sequence and picture level parameters reliably to the receiver. As reviewed in Section 2.6.1, parameter sets of H.264/AVC can be transmitted reliably by out-of-band means, such as within the SDP description of the stream. If in-band transmission is necessary, parameter sets should be protected strongly or repeated to guarantee their successful reception.

One way to enable encoders to make the impact of transmission errors to picture quality smaller is to attach error concealment side information into the bitstream. Video coding standards support supplemental enhancement information or user data for associating a piece of data to certain coded pictures. Hence, the use of data hiding techniques for the carriage of error concealment data seems unnecessary for packet-oriented transport, even though it is proposed in some papers, such as [164]. The purposes for error concealment side information can be roughly classified into bit error detection, resynchronization of decoding after bit errors, motion vector recovery, texture recovery, and concealment algorithm selection. Bit error detection and resynchronization of decoding after bit errors are not in the scope of this thesis and therefore not reviewed, whereas methods for the other types of side information are briefly outlined below.

For motion vector recovery, two examples are given in the following. First, Song and Liu proposed that a parity vector is calculated over the motion vectors of a picture and embedded in the motion vectors of the following picture [125]. Second, MPEG-2 Video [66] provides a mechanism to transmit redundant motion vectors for macroblocks in intra pictures. In the case of transmission errors, these motion vectors can be used for motion-compensated error concealment. The MPEG-2 mechanism can be useful for periodic intra pictures, for which motion-compensated error concealment typically performs better than spatial error concealment.

There are two common approaches for texture recovery. First, edge direction side information has been reported to improve error concealment considerably [164]. Second, the intensity and texture of a block can be derived from spatially neighboring blocks better if there is side information to indicate how the interpolation should be performed. For example, Hemami and Gray proposed to add vector-quantized weights as auxiliary information for linear interpolation from adjacent blocks [57].

A few approaches have been proposed to signal the optimal selection of the error concealment algorithm as side information. Wenger et al. proposed signaling of the error concealment algorithm on picture basis in the supplemental enhancement information of H.263 [151]. Cen and Cosman proposed a decision tree for the optimal error concealment algorithm

[16]. The decision tree is built based on coherence measures for global motion and parameters of vertically adjacent macroblocks such as macroblock mode, motion vector amplitude and angle difference, and DCT coefficient statistics. The leaves of the decision tree correspond to particular error concealment algorithms. The threshold values for each decision node, corresponding to the mentioned parameters, are adjusted on picture basis to optimize the result of error concealment. The decision tree is coded and included as auxiliary information into the bitstream. Decoders select the error concealment algorithm for a corrupted macroblock by entering the calculated parameters mentioned above to the received decision tree.

The methods in the literature do not indicate a proper error concealment strategy for entirely lost pictures. In order to select a good error concealment method for a completely lost picture, it is essential to conclude whether or not the missing picture represented a scene cut. The scene information SEI message presented in Section 7.1 assists the selection of the error concealment method for an entirely lost picture, and it also speeds up the selection of the error concealment method for partially lost or corrupted pictures.

Even with the assistance of the methods presented in this section, the error-concealed areas can be clearly perceivable and annoying. The methods in the literature do not help decoders conclude when the error-concealed decoding result would be subjectively satisfactory for displaying and when it would be better to display the latest correct or satisfying picture instead. The spare picture SEI message, introduced in Section 7.2, addresses this challenge by indicating which areas can be satisfactorily error-concealed by copying the co-located area in a particular previous picture.

4.5.8. Unequal Error Protection

This section discusses some UEP schemes applicable to forward error correction and concealment. As with equal error protection, UEP can be realized in any protocol stack layer. For example, Wei proposed a modulation scheme with non-uniformly spaced signal points to achieve unequal protection in the physical layer [148]. Most UEP schemes are anyhow applied in the application layer and are therefore reviewed in this section, which is organized as follows. Priority Encoding Transmission (PET), introduced in the paragraph below, is often regarded as the initial work for unequal forward error correction. The work by Horn et al. [58] and the UEP method presented in this thesis are briefly discussed with relation to PET. The activities in IETF are reviewed after that. Finally, a media-aware approach for unequal error protection, known as systematic lossy error protection (SLEP), is presented. It is also noted that many of the source coding methods presented earlier, such as redundant coded pictures, can be applied unequally to different parts of streams.

PET [5] established the work towards unequal error protection in packet-oriented systems. The data to be transmitted is partitioned to messages, which are protected one at a time. The messages are then classified into priority segments according to known characteristics of the source signal. Leicher applied PET to MPEG-1 [61] bitstreams, for which a message was defined as a GOP and priority segments were assigned according to the picture type (I, P, B) [87]. FEC repair data is then generated for each priority segment, and the resulting code

stream is divided into a certain amount of packets, each containing a fixed-length block of data from the resulting code stream. The amount of FEC repair data is a function of the priority class. The PET scheme results into packets which contain data from each priority segment, and the number of packets required to reconstruct a priority segment can be tuned with the amount of FEC repair data per that priority segment. Horn et al. developed a similar scheme [58] compared to PET. They provided details on the practical implementation and application with a spatially scalable video codec. The work in this thesis, summarized in Section 6.3.2, presents an UEP scheme similar to PET and the method by Horn et al. [58]. It is shown how the presented method is applied to H.264/AVC and integrated to RTP-based transmission systems, MBMS and DVB-H taken as practical examples. The data transmission order of the presented method is selected to minimize the expected tune-in delay between the start of the reception of a broadcast and the start of the rendering.

IETF RFC 2733 [109] specifies an RTP payload format for XOR-based FEC protection. The payload header of FEC packets contains a bit mask identifying the packet payloads over which the bit-wise exclusive or (XOR) operation is calculated. One XOR FEC packet enables recovery of one lost source packet. IETF RFC 5109 [89] replaced IETF RFC 2733 recently with a similar RTP payload format for XOR-based FEC protection also including the capability of uneven levels of protection. The payloads of the protected source packets are split into consecutive byte ranges starting from the beginning of the payload. The first byte range starting from the beginning of the packet corresponds to the strongest level of protection and the protection level decreases as a function of byte range order.

Rane et al. proposed an unequal error protection method known as systematic lossy error protection (SLEP) [104], which is based on the Wyner-Ziv coding theorem [160][161]. According to SLEP, the transform coefficients of a set of encoded pictures are requantized with a coarser quantization step. Other coding parameters, such as motion vectors, remain unchanged compared to the original encoded pictures. The resulting coarsely quantized redundant coded pictures are placed in an FEC source matrix and FEC repair data, such as Reed-Solomon parity bits, are calculated over the source matrix. Primary coded pictures and the FEC repair data are transmitted, while the redundant coded pictures are not sent. It is noted that primary coded pictures may be further partitioned to coded slices, each of which may be encapsulated in a separate transmission packet. Certain pieces of side information, such as the quantization step size for the requantization step in the process of forming the redundant coded pictures and the location of the redundant version of each primary coded slice in the FEC source matrix, are additionally transmitted. In the receiving end, an FEC source matrix is filled in by re-quantizing the received primary coded slices. If any primary coded slice is lost or corrupted, the missing data in the FEC source matrix can be recovered using the received FEC repair data provided that the correction capability of the received FEC repair data is sufficient. The redundant coded pictures contained in the recovered FEC source matrix can then be decoded and used to replace corrupted primary pictures entirely or partly. Baccichet et al. applied SLEP to regions of interest coded as rectangular-oriented slice groups of H.264/AVC

[8]. Consequently, the redundant pictures over which the FEC code is calculated contain only the slice groups covering the identified regions of interest.

4.6. ERROR CONCEALMENT BY POST-PROCESSING

Error concealment algorithms can be categorized into spatial and temporal methods. In spatial error concealment, only the information from the current coded picture or decoded picture is used. Temporal error concealment utilizes the reconstructed information from previously decoded pictures. A brief review of both spatial and temporal error concealment algorithms is provided below.

Spatial error concealment can operate either in the frequency domain or in the sample domain. In frequency-domain concealment, the transform coefficients of missing blocks are derived from the transform coefficients of adjacent blocks. For example, an average of the DC coefficients of the adjacent blocks can be used as a concealed coded block. In another approach known as maximally smooth recovery [142], a limited number of DCT coefficients are estimated to provide the smoothest connection with the boundary pixels of the adjacent blocks. In general, frequency-domain algorithms usually interpolate only the low-frequency transform coefficients. In sample-domain concealment, the sample values of a missing block are derived from the sample values of the neighboring blocks. For example, Salama et al. proposed weighted pixel averaging [112], in which each pixel value in a macroblock to be concealed is formed as a weighted sum of the closest boundary pixels of the selected adjacent macroblocks. The weight associated with each boundary pixel is relative to the inverse distance between the pixel to be concealed and the boundary pixel. Weighted pixel averaging was also selected into the JM reference software of H.264/AVC [P3][131][146]. In edge-preserving algorithms, such as [130], [134], and [169], the main edges in the adjacent blocks are detected and the missing pixels are interpolated along the found edge directions. Algorithms based on texture analysis and synthesis have also been developed [10].

The basic idea of temporal error concealment is to estimate the motion vector of a lost block. A simple strategy is to use a zero motion vector or a median of the motion vectors in the neighboring blocks. Lam et al. proposed a boundary matching algorithm, in which the motion vector resulting into the smallest absolute or squared difference between the boundary pixels of an adjacent correctly decoded block and the concealed block is selected [85]. The boundary matching algorithm applied to multiple reference pictures was selected into the JM reference software of H.264/AVC [P3][131][146]. An example of an alternative approach is provided in [124], where the motion vectors of neighboring blocks are used to obtain candidate blocks for the missing block. A weighted average of the candidate blocks is calculated, where the weights are selected to minimize the squared difference of the boundary pixels.

A source decoder should have at least one spatial and one temporal error concealment algorithm. A simple rule is to use the spatial and temporal concealment algorithms for intra and inter pictures, respectively. However, such a simple rule is often not optimal. For example, temporal error concealment usually works better for periodic intra pictures and sometimes

spatial concealment is preferable in inter pictures. One method to select between spatial and temporal concealment is to perform scene cut detection for the correctly received part of a picture. If the picture is concluded to be a scene-cut picture, spatial concealment is performed. Any scene cut detection algorithm is applicable for the selection of the error concealment algorithm. For example, an algorithm considering the variance of motion vectors and the number of intra-coded blocks was proposed in [25]. The decision between spatial and temporal error concealment can also be done on block basis [32]. Error concealment algorithms of the same type can also be switched based on local statistics of the picture that is concealed. For example, it was proposed in [136] that either maximally smooth recovery [142] or weighted pixel averaging [112] is adaptively selected for each macroblock based on the spatial activity of the neighboring areas. It is noted that source encoders can assist in the selection of a proper error concealment algorithm by including auxiliary information into the coded bitstream. Such techniques are reviewed in Section 4.5.7 and Chapter 7.

4.7. SUMMARY AND DISCUSSION

The choice of the most suitable video error resilience methods depends on the applications, networks, and their inherent characteristics. In this section, the usefulness of the presented error resilience methods for different applications is discussed. Furthermore, the necessary enabling factors for the use of each error resilience method are summarized.

The considered applications include unicast streaming, multicast/broadcast streaming, point-to-point video telephony, and multipoint video conferencing. 3GPP Packet-Switched Streaming Service was provided as an example of unicast streaming in Section 3.4.2. Examples of multicast/broadcast streaming include IP Data Casting over DVB-H and 3GPP Multimedia Broadcast/Multicast Service, which were reviewed in Sections 3.3 and 3.4.3, respectively. Point-to-point video telephony refers to full-duplex live video communication between two end-points. Multipoint video conferencing refers to a system with multiple end-points and a multipoint control unit (MCU), which mixes the input streams from the end-points and forwards the mixed stream to all participants. A sophisticated MCU may produce a different stream for each end-point according to the capabilities of the end-point, the network connection of the end-point, and the wishes of the end-user [35].

The following characteristic factors of real-time video communication systems and applications jointly affect which error control methods are the most applicable: availability and types of feedback, quality of service guarantees of the channel, latency requirements of the application and end-to-end latency of the communication system, and the presence of live encoding [P8][144]. These characteristics are discussed in the Sections 4.7.1 to 4.7.4, respectively, after which a summary of the most suitable error resilience methods for the mentioned applications is provided in Section 4.7.5.

4.7.1. Availability and Types of Feedback

The availability of feedback messages depends on the routing scheme used in the communication system. Unicast delivery is usually accompanied by a feedback channel. In multicast delivery, the amount and frequency of allowed feedback is typically a decreasing function of the number of subscribers in order to avoid an explosion of feedback traffic in the channel. For the same reason or due to the lack of a return channel, there is no or little application-layer feedback in broadcast delivery.

Types of feedback can be categorized according to the protocol stack layer used to convey feedback messages. Transport and lower layer feedback relates often to the use of an ARQ protocol (see Section 4.4) to guarantee reliable delivery. The acknowledged mode of UTRAN is an example of link-layer ARQ feedback (see Section 3.4.1), whereas TCP includes a transport-layer ARQ mechanism (see Section 4.4). Connectionless protocols may provide receiver state feedback, such as RTCP receiver reports (see Section 3.2), based on which the sender may also conclude the channel state (see Section 4.2.1). The receiver state feedback may also address parts of the application layer, such as the client buffer feedback of 3GPP packet-switched streaming (see Section 4.2.1) and quality of experience measures [2], up to the decoder state feedback exemplified by reference picture selection messages (see Section 4.3.2). [P8]

4.7.2. Quality of Service Guarantees

In general, communication systems can operate on best-effort basis or provide a certain level of quality of service. In a best-effort system, no guarantee on correct delivery of packets is given. If reliable delivery is desirable, transport or application layer protocols have to be used to achieve robustness in delivery. Guaranteed quality of service can be characterized in terms of guaranteed and maximum bitrate throughput, maximum bit or packet error rate, maximum end-to-end latency, and maximum end-to-end latency variation. In circuit-switched systems, the channel bitrate is typically constant, which can also be considered as a quality of service guarantee. If the channel bitrate is varying, senders have to adapt the transmitted bitrate to the expected channel throughput bitrate. This is known as bitrate adaptation, which was reviewed in Section 4.2. [27][83]

4.7.3. Latency

Many types of latencies are inherent in multimedia communication systems, including end-to-end latency and startup latency. End-to-end latency refers to the time from the capturing a media sample to the rendering of the sample in the far end, although sometimes end-to-end latency is discussed only in the context of transport and then it refers to the time from the transmission of a packet to its reception. Communication systems may suffer from end-to-end latency variation, which can be smoothed with de-jitter buffering in the receiver.

Startup latency refers to the time from the user's initiation of a multimedia session to the moment when media rendering starts. Startup latency may be further divided into the con-

nection establishment time and the initial media buffering time. The connection establishment time lasts from the user's initiation of a multimedia session to the reception of the first media bit, whereas initial media buffering lasts from the reception of the first media bit to the start of the media rendering. Clients of unicast streaming applications typically have a receiver buffer that is capable of storing a relatively large amount of data. Initially, when a streaming session is established, a client does not start playing the stream back immediately, but rather it typically buffers the incoming data for a period of time even up to a few seconds. This buffering helps in maintaining continuous playback, because the client can decode and play buffered data in case of occasional increased transmission delays or network throughput drops. Otherwise, without initial buffering, the client would have to freeze the display, stop decoding, and wait for incoming data. [23]

4.7.4. Live Encoding or Pre-Encoded Content

In some applications, such as video telephony, encoding is naturally done live, i.e., the encoded content is transmitted right away. Other applications, such as streaming, allow encoding to be done live or in advance. When encoding is done before transmission, the encoding delay may not be an issue, which gives the opportunity to use complex or high-latency encoding algorithms, such as two-pass rate control, for achieving improved compression efficiency compared to real-time encoding. However, adaptation of pre-encoded content according to the prevailing transmission conditions is often more difficult compared to live encoding, where source-coding-level error robustness tools can be controlled according to the received feedback. [P8]

4.7.5. Applicable Types of Error Resilience Methods

Table I summarizes the availability and usefulness of error resilience techniques for different applications. The following paragraphs describe how to read the table, while several details of the selected classification are provided below the table.

The leftmost column ("Error resilience technique") includes the name of the referred technology and the section number where it was discussed. The categorization of the error resilience methods follows that given in Chapter 1. It is noted that in wireless communication systems forward error correction is always applied in the physical layer and therefore it is not mentioned separately in Table I. Long-interleave forward error correction refers to the techniques where the FEC code is calculated over several transport layer packets, hence inheriting a considerable FEC coding and decoding delay.

The second column from the left ("Enabling factors") includes the characteristics of the application or the delivery method that enable the use of a particular type of an error resilience technique. Most of these characteristics were listed above.

The four rightmost columns stand for the applications and services introduced at the beginning of Section 4.7. Each application column indicates whether a particular type of error resilience technique is available and useful for the application.

Table I. Availability and usefulness of error resilience techniques for different applications.

		Application			
		Unicast streaming	**Multicast / broadcast streaming**	**Point-to-point video telephony**	**Multipoint video conference**
Error resilience technique	**Enabling factors**	**Error resilience technique available? / useful?**			
Robust packet scheduling (Section 4.2.2)	Startup latency, unicast delivery, receiver state feedback	Yes / Yes	Limited / Partly [1]	No / No	No / No
Stream thinning and switching (Section 4.2.3)	Scalable or multiple bitstreams, unicast delivery, receiver state feedback	Yes / Yes	No / No	No / No [2]	Yes / Yes [3]
Interactive error concealment (Section 4.3)	Live encoding, decoder state feedback	Limited / No [4]	No / No	Yes / Yes	Yes / Yes [5]
Interactive error correction (Section 4.4)	Startup latency, receiver state feedback	Yes / Yes	No / No	Limited / Partly [6]	Limited / Partly [6]
Forward error correction and concealment at source coding layer (Sections 4.5.1 to 4.5.7)	Always possible. (Channel state feedback[7])	Yes / Partly [8]	Yes / Partly [9]	Yes / Yes	Yes / Yes
Long-interleave forward error correction (Examples provided in Sections 3.3 and 3.4.3)	Support in the protocol stack	Yes / Partly [8]	Yes / Yes	Limited / Partly [6]	Limited / Partly [6]
Error concealment by post-processing (Section 4.6)	Always possible	Yes / Yes	Yes / Yes	Yes / Yes	Yes / Yes

[1] Robust packet scheduling could be used in a limited fashion for multicast delivery depending on the number of receivers.
[2] Bitrate control algorithm of the encoder is used instead.
[3] The MCU can perform stream thinning [35].
[4] Interactive error concealment could be applied in live unicast streaming, but interactive error correction is usually more efficient and applied instead [129].
[5] If the number of participants in the multipoint conference is high, interactive error concealment may not be feasible any longer.
[6] This requires retroactive accelerated decoding in order to meet the strict end-to-end delay requirement. Retroactive accelerated decoding may be achieved by re-decoding only a subset of the pictures. See Section 4.4.
[7] The availability of channel state feedback improves compression efficiency, because the amount of redundancy can then be adjusted according to the prevailing channel conditions.
[8] Interactive error correction is often more efficient and used instead [129].
[9] Long-interleave FEC is often more efficient and used instead [129]. Intra pictures are anyhow used for providing random access points.

Chapter 5

Isolated Regions

Techniques for constraining in-picture prediction were reviewed in Section 4.5.1. In fact, one of the presented techniques, the independent segment decoding mode of H.263, restricts in-picture and inter prediction jointly such a way that slice boundaries remain unchanged within a group of pictures and slice boundaries are considered as picture boundaries in inter prediction. Consequently, inter prediction of a slice can only refer to the co-located slices in reference pictures, and no spatio-temporal error propagation over slice boundaries occurs. The facts that slice boundaries enclose only rectangular regions and remain unchanged within a group of pictures hinder the use of the independent segment decoding mode for many applications, as presented later in this section. Furthermore, because the number of macroblocks in a slice is constant within a group of pictures, the encoder has few means to control the coded size of a slice in terms of bytes. This fact may make the encapsulation of slices into transport packets non-optimal, because the slice size cannot be adjusted according to an optimal packet size for the prevailing network conditions.

The isolated regions technique, presented in depth in [P5], is based on constraining in-picture prediction and inter prediction jointly but avoids the shortcomings of the independent segment decoding mode. The isolated regions technique is therefore suitable for a greater variety of applications and error robustness methods when compared to the independent segment decoding mode as presented later in this chapter.

This chapter is based on [P5] and organized as follows: Section 5.1 provides the fundamentals of the isolated regions technique. The realization of the isolated regions technique in the H.264/AVC standard is reviewed in Section 5.2. The use of isolated regions for error-robust random access, loss-aware macroblock mode selection, and picture partitioning for unequal error protection is reviewed in Sections 5.3, 5.4, and 5.5, respectively.

5.1. OVERVIEW OF THE ISOLATED REGIONS TECHNIQUE

An isolated region in a picture can contain any macroblock locations, and a picture can contain zero or more isolated regions that do not overlap. A leftover region is the area of the

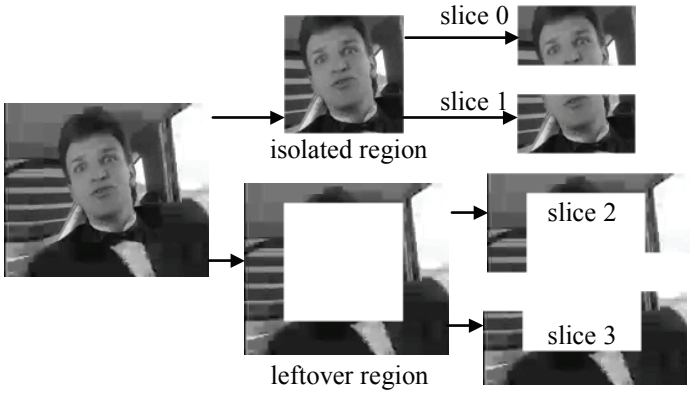


Figure 10. Example partitioning of a picture to an isolated region and a leftover region and further to slices.

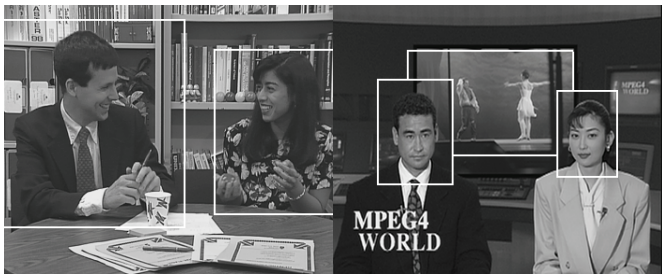
picture that is not covered by any isolated region of a picture. When coding an isolated region, in-picture prediction is disabled across its boundaries. A leftover region may be predicted from isolated regions of the same picture.

A coded isolated region can be decoded without the presence of any other isolated region or the leftover region of the same coded picture. It is usually necessary to decode all isolated regions of a picture before the leftover region. An isolated region or a leftover region contains at least one slice. Figure 10 presents an example where the picture contains one isolated region and a leftover region. Both the isolated region and the leftover region contain two slices.

Pictures, whose isolated regions are predicted from each other, are grouped into an isolated-region picture group. An isolated region can be inter-predicted from the corresponding isolated region in other pictures within the same isolated-region picture group, whereas inter prediction from other isolated regions or pictures outside the isolated-region picture group is disallowed. A leftover region may be inter-predicted from any isolated region. The shape, location, and size of coupled isolated regions may evolve from picture to picture in an isolated-region picture group.

5.2. CODING OF ISOLATED REGIONS IN H.264/AVC CODECS

Coding of isolated regions in H.264/AVC codecs is based on slice groups introduced in Section 2.3. The mapping of macroblock locations to slice groups is specified in the picture parameter set. The H.264/AVC syntax includes efficient methods to code certain slice group patterns, which can be categorized into two types, static and evolving. Static slice groups stay unchanged as long as the picture parameter set is valid, whereas evolving slice groups can change picture by picture according to the corresponding parameters in the picture parameter set and a slice group change cycle parameter in the slice header. Static slice group patterns include interleaved, checkerboard, rectangular-oriented, and freeform. Evolving slice group patterns include horizontal wipe, vertical wipe, box-in, and box-out. The rectangular-oriented



(a) (b)

Figure 11. Examples of rectangular-oriented isolated regions.

pattern and the evolving patterns are especially suited for coding of isolated regions and are described in more detail next.

For a rectangular-oriented slice group pattern, a desired number of rectangles are specified within the picture area. A foreground slice group includes the macroblock locations that are within the corresponding rectangle but excludes the macroblock locations that are already allocated by slice groups specified earlier in the same picture parameter set. A leftover slice group contains the macroblocks that are not covered by the foreground slice groups. Figure 11(a) includes two rectangular foreground slice groups (indicated by a white rectangle) and Figure 11(b) includes three foreground slice groups, two of which are rectangular and the third one, i.e., the screen behind the newsreaders, is composed by excluding the first two rectangles from the bounding rectangle.

An evolving slice group is specified by indicating the scan order of macroblock locations and the change rate of the size of the slice group in number of macroblocks per picture. Each coded picture is associated with a slice group change cycle parameter (conveyed in the slice header). The change cycle multiplied by the change rate indicates the number of macroblocks in the first slice group. The second slice group contains the remaining macroblock locations. Figure 12 shows an example of the first five change cycles of the first slice group of the box-out type with a change rate of 12 macroblocks.

H.264/AVC encoders can form an isolated-region picture group as follows. Each slice group has a unique identification number within a picture. Encoders can restrict the motion vectors in a way that they only refer to the decoded macroblocks belonging to slice groups having the same identification number as the slice group to be encoded. Encoders should take into account the fact that a range of source samples is needed in fractional pixel interpolation and all source samples should be within a particular slice group.

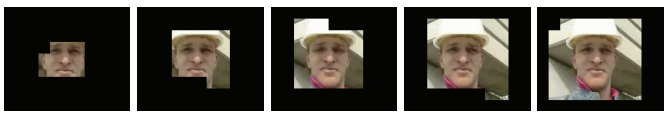


Figure 12. Example of an evolving isolated-region picture group.

The presence of an isolated-region picture group can be indicated with the motion-constrained slice group set SEI message of H.264/AVC, first proposed in [S10]. The SEI message can be present only in an IDR access unit, and it indicates that the listed slice groups form an isolated-region picture group until the next IDR access unit, exclusive. The SEI message can also indicate if the isolated-region picture group represents a certain pan-scan rectangle specified in the pan-scan rectangle SEI message. Pan-scan rectangles can be used to adjust bitstreams coded at a particular picture aspect ratio to another picture aspect ratio by displaying only the area of the pan-scan rectangle. For example, the pan-scan rectangle SEI message can indicate a 4:3 picture within a wide-screen picture.

When an isolated-region picture group is used for gradual decoding refresh (introduced in Section 5.3), its presence can be indicated with the recovery point SEI message of H.264/AVC. The *changing_slice_group_idc* syntax element of the SEI message indicates which one of the two slice groups of the evolving slice group types is a foreground slice group that has to be decoded while the decoding of the leftover slice group can be omitted.

When deblocking loop filtering is turned off at slice boundaries, decoding of the foreground slice group only within an isolated-region picture group results into exactly correct sample values. Otherwise, there are minor differences between some of the sample values resulting from the decoding of the foreground slice group only and the sample values resulting from the decoding of entire coded pictures. However, according to our experiences, mismatches are unperceivable and the picture quality is acceptable without turning off the loop filtering at slice boundaries.

5.3. ERROR-ROBUST RANDOM ACCESS

Gradual decoding refresh (GDR) refers to the ability to start the decoding at a non-IDR picture and recover decoded pictures that are correct in content after decoding a certain amount of pictures. Some reference pictures for inter prediction may not be available between the random access point and the recovery point, and therefore some parts of decoded pictures in the gradual decoding refresh period cannot be reconstructed correctly. However, these parts are not used for prediction at or after the recovery point, which results into error-free decoded pictures starting from the recovery point.

It is obvious that gradual decoding refresh is more cumbersome both for encoders and decoders compared to instantaneous decoding refresh. However, gradual decoding refresh can be desirable in error-prone environments due to two facts: First, a coded intra picture is generally considerably larger than a coded non-intra picture. This makes intra pictures more susceptible to errors than non-intra pictures, and the errors are likely to propagate in time until the corrupted macroblock locations are intra-coded. Second, intra-coded macroblocks are often used in error-prone environments to stop error propagation (see Section 4.5.3 for more details). Thus, it can be beneficial to combine intra macroblock coding for random access and error propagation prevention.

An evolving isolated region can be used to provide gradual decoding refresh. A new evolving isolated-region picture group is established in the picture at the random access point and is completed when the isolated region covers the whole picture area. A picture completely correct in content is obtained when decoding started from the random access point. This process can also be generalized to include more than one evolving isolated region that eventually cover the entire picture area. As explained in Section 5.2, the recovery point SEI message of H.264/AVC can be used to indicate the presence of an isolated-region picture group for gradual decoding refresh. Gradual decoding refresh using isolated regions can also be applied as a response to a fast picture update command or a picture loss indication, which were discussed in Section 4.3.1.

H.264/AVC coding efficiency simulations comparing gradual decoding refresh based on isolated regions with periodic IDR picture coding at a 1-second random access period were performed for [P5] and some of its reference publications. Error-free application environment, such as local storage, was assumed, and therefore the coding options yielding the best coding efficiency were selected. The simulations abided the coding efficiency simulation common conditions specified by ITU-T Video Coding Experts Group [133]. A number of QCIF and Common Intermediate Format (CIF) sequences were coded, and the average bitrate loss of gradual decoding refresh compared to periodic IDR was between 11% and 17 %. More results can be found in [S4].

The simulations in [S4] also measured the error resilience performance of gradual decoding refresh compared with the periodic IDR picture coding. The target was to simulate IP multicast streaming where random access points allow new receivers to start decoding. The results presented in [S4] revealed that gradual decoding refresh performs consistently better compared to periodic IDR in all loss rates. The results suggest that gradual decoding refresh has better error resilience performance than intra pictures when it is used as a response to fast picture update requests in video conferencing applications. In multicast and broadcast streaming services, the error robustness of gradual decoding refresh is usually unnecessary, because relatively strong FEC is typically applied.

5.4. LOSS-AWARE MACROBLOCK MODE DECISION

Macroblock mode decision algorithms were reviewed in Section 4.5.3. Evolving isolated regions can be used as a non-adaptive macroblock mode selection algorithm. A new evolving isolated-region picture group is established at the beginning of an intra refresh period. The intra refresh period is completed when the isolated region covers the entire picture area. The macroblocks in the isolated region of the first picture in the intra refresh period are intra-coded. The newly added macroblocks in the isolated region of later pictures are intra-coded, whereas the other macroblocks in the isolated region can be inter-predicted from the corresponding isolated region within the same intra refresh period. The encoder can select a proper change rate of the isolated region according to the picture size and the assumed transmission error rate. Generally, a good change rate is equivalent to the expected loss rate of macrob-

locks. For example, for a CIF sequence, if the packet loss rate is 20%, a change rate of about 80 macroblocks per picture is appropriate. However, due to the possible large differences in sequence characteristics and different coding options, a content-adaptive change rate may perform better but is left out of the scope of this thesis. Furthermore, in contrast to coding newly added macroblocks in intra mode, the encoder can apply a normal macroblock mode selection algorithm for them. As a result, the newly added macroblocks may be inter-predicted from the corresponding isolated region in the same isolated-region picture group or they may be intra-coded.

Four intra refresh algorithms implemented for a draft H.264/AVC codec were compared in [S3]: conventional circular intra refresh at a rate of one macroblock row per picture (CIR), the loss-aware rate-distortion-optimized macroblock mode selection of the Joint Model reference implementation of H.264/AVC (LA-RDO) [127], isolated regions based circular intra refresh (IREG-CIR), and a combination of LA-RDO and IREG-CIR. Real-time multicast/broadcast to users with different network conditions was assumed. Figure 13 presents the average luma Peak Signal-to-Noise Ratio (PSNR) of all the test sequences for each intra refresh algorithm and each packet loss rate. The simulation results show that the difference in average luma PSNR between IREG-CIR and LA-RDO is within 0.5 dB regardless of the packet loss rate. In packet loss rates greater than or equal to 5 %, the combination of LA-RDO and IREG-CIR outperforms other algorithms, the difference being more than 0.5 dB in the 20 % packet loss rate case, to which the bitstreams were optimized. More details on the simulation conditions and results are available in [S3].

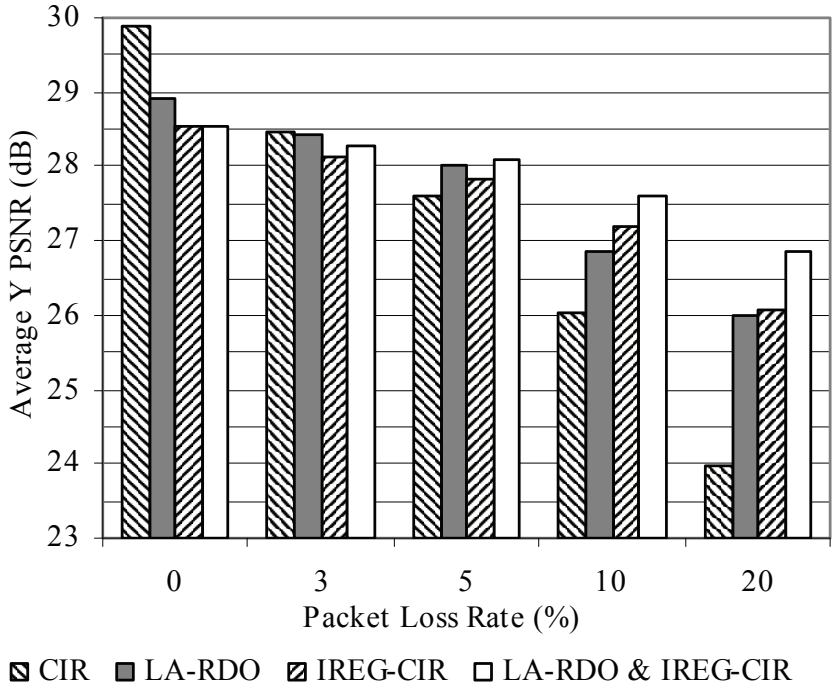


Figure 13. Comparison of macroblock mode selection algorithms at different packet loss rates.

5.5. PICTURE PARTITIONING FOR UNEQUAL ERROR PROTECTION

Isolated regions can be used for unequal error protection similarly to foveation-based unequal error protection proposed by Lee et al. [86], which was reviewed in Section 4.1.4. The encoder first selects at least one region of interest from the first picture to be encoded using, e.g., face detection. Each region of interest is an isolated region, and the remaining macroblocks form the leftover region. In the next picture to be encoded, the encoder tracks the same regions of interest as in the previous picture. Each region of interest is coded as an isolated region that is inter-predicted only from the corresponding isolated region in the previous reference pictures.

In the simulations for [P5] and [S9], a constant rectangular region of interest was selected for each sequence, smaller quantization steps were used within the region of interest, and the regions of interest were coded as an isolated-region picture group. The scheme was compared to the conventional coding without region of interest detection. The region of interest (ROI) was packetized separately from the leftover region, and IETF RFC 2733 [109] (see Section 4.5.8) was used to protect the region of interest. It was found that the average luma PSNR in the region of interest was significantly higher in all tested loss rates when compared to the bitstreams without ROI detection. However, for sequences with changing background, the average luma PSNR for the entire picture area dropped in the proposed ROI coding with UEP compared to conventional coding. However, in our opinion, most errors in the background were less noticeable than errors in the foreground and therefore the overall subjective quality was improved. Further details on the simulation conditions and results can be obtained from [S9].

In [29], isolated regions are used in a transcoder for unicast streaming to code regions of interest with a better quality compared to the leftover region. The quality of the leftover region is selected to adapt the transmitted bitrate according to the prevailing network conditions.

Chapter 6

Sub-sequences and Interleaved Transmission

This chapter describes two techniques, sub-sequences and interleaved transmission, which can be used for various applications. Sub-sequences and interleaved transmission are often used together and therefore this chapter reviews them both. The sub-sequence technique enables disposal of inter-dependent reference pictures while the remaining bitstream remains compliant with the H.264/AVC standard. It can therefore be used for hierarchical temporal scalability among other things. Interleaved transmission refers to sending of coded picture data in an order that differs from the decoding order.

Reference picture selection enables many types of temporal scalability schemes. For example, the use of two temporal layers achieved by forward-predicted inter pictures was presented in [150]. Hierarchical or recursive temporal scalability was proposed in [22] for forward-predicted inter-pictures only and was later expanded to bi-predictive pictures in the sub-sequence technique [S2]. Figure 14 shows an example of hierarchical temporal scalability. I, P, B, and b indicate an intra reference picture, an inter reference picture, a bi-predictive reference picture, and a bi-predictive non-reference picture, respectively. The example scheme can be decoded with three constant frame rates by decoding I and P pictures only, I, P, and B pictures, or all pictures.

When the sub-sequence coding method is in use, coded pictures are first mapped to sub-sequence layers, which are arranged hierarchically based on their dependency on each other. The base layer (layer 0) is independently decodable, while correct decoding of sub-sequence layer N requires decoding of layers from 0 to N-1. It is recommended to organize

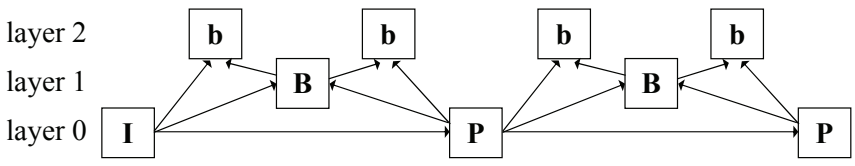


Figure 14. Example of sub-sequences: coding pattern “IbBbP”.

sub-sequences into sub-sequence layers in such a way that discarding of sub-sequence layers results in a constant or nearly constant picture rate. Picture rate and therefore subjective quality increase along with the number of decoded sub-sequence layers. A sub-sequence consists of a number of inter-dependent coded pictures that can be disposed without affecting the decoding of any other sub-sequence in the same sub-sequence layer or any picture in any lower sub-sequence layer. Consequently, any sub-sequence in the highest present sub-sequence layer can be removed, while the thinned bitstream remains standard-compliant.

Hierarchical temporal scalability structures can be used as a temporal segmentation tool to increase the number of priority partitions from the two partitions achieved with the conventional classification into reference and non-reference pictures. As presented in [14], the sub-sequence layer of a picture can directly indicate its priority partition. Hierarchical temporal scalability also provides more flexibility for the adaptation of the bitstream compared to conventional non-scalable and non-hierarchical video coding methods. Furthermore, it has been shown that hierarchical temporal scalability improves compression efficiency as discussed later in Section 6.1.3.

This chapter is organized as follows: Section 6.1 explains how the sub-sequence method is supported in the H.264/AVC standard. Section 6.2 introduces the RTP payload format of H.264/AVC and especially its interleaved packetization mode. Section 6.3 summarizes a few application examples how sub-sequences and interleaved packetization can be used to improve error resilience.

6.1. SUB-SEQUENCES IN H.264/AVC

This section presents how the sub-sequence technique is supported in H.264/AVC. The sub-sequence feature in H.264/AVC is based on our input contribution [S2].

H.264/AVC enables selection of the reference picture for each block, and the reference picture is indicated by an entropy-coded index to a reference picture list. One of the challenges in the design for hierarchical temporal scalability in H.264/AVC was to ensure that the reference picture lists remain unchanged regardless of which coded pictures of the original bitstream are removed. For example, if B and b pictures of the example in Figure 14 are removed from the bitstream, the reference picture list for P pictures must be identical compared to the case in which all pictures are present in the bitstream. The sub-sequence coding technique was developed to process reference picture lists correctly when hierarchical temporal scalability is used. Section 6.1.1 describes which techniques are used to keep reference picture lists unchanged regardless of which sub-sequences of the original bitstream are removed.

The sub-sequence technique enables stream thinning in units that are finer than entire scalability layers and coarser than individual non-reference pictures. H.264/AVC includes SEI messages enabling identification of independent sub-sequences within the layers, which makes bitrate shaping easier. The SEI messages related to sub-sequences are reviewed in Section 6.1.2.

Section 6.1.3 summarizes the study of hierarchical temporal scalability in H.264/AVC published in [P6]. Both the compression efficiency improvement and the bitrate adaptation capability are studied.

The scalable extension of H.264/AVC uses the sub-sequence technique for hierarchical temporal scalability. Section 6.1.4 explains how the sub-sequence technique is supported in the scalable extension of H.264/AVC.

6.1.1. Reference Picture List Construction

The decoding process for the bitstream has to be such that it does not depend on the presence or absence of any disposable sub-sequences. In particular, the reference picture list for any reference picture in the bitstream must be identical regardless of which disposable sub-sequences are present in the bitstream. This section reviews the mechanisms of H.264/AVC that ensure correct reference picture list construction. The mechanisms are complementary with each other and can be classified into three categories: decoding process for absent sub-sequences [S2], assignment of long-term reference pictures, and reference picture list reordering.

The H.264/AVC standard includes a decoding process for gaps in frame_num that creates a "non-existing" decoded picture marked as "used for short-term reference" for each reference picture that was present in an absent sub-sequence. The number of missing reference pictures can be derived from a counter included in the slice header, known as *frame_num*, which is incremented by 1 per each reference picture [S1]. The sample values of a "non-existing" picture can be set freely and, in fact, many decoder implementations may not create a decoded picture but rather an indication of a "non-existing" picture. "Non-existing" pictures are processed identically to ordinary reference pictures according to memory management control operations and reference picture list reordering commands. As any picture in the absent sub-sequence is not used as prediction reference, "non-existing" pictures do not affect the decoding process except for reference picture list construction and ensure that the reference picture list for inter-coded slices is identical compared to the bitstream in which the sub-sequence is present. Only when a "non-existing" picture is referred in the inter prediction process, an unexpected picture loss due to a transmission error, for example, can be deduced. It is required that the pictures in removable sub-sequences must not contain memory management control operations that could change the initial reference picture list. The use of the gaps in frame_num decoding process suits shallow temporal scalability hierarchies that do not cause the previous reference picture of sub-sequence layer 0 to be removed due to the sliding window process of "non-existing" pictures. For example, the three-layer temporal scalability hierarchy presented in Figure 14 can be implemented with the decoding process of gaps in frame_num.

If such a deep temporal scalability hierarchy is used that would cause the previous reference picture of any base sub-sequence layer to be removed due to the sliding window process of "non-existing" pictures, a reference picture can be marked as "used for long-term reference" immediately after its decoding. It is remarked, however, that no motion vector scaling

of temporal and spatial direct mode of bi-predictive slices according to picture order count values is done for long-term reference pictures. Furthermore, equal weights instead of weights calculated according to picture order counts are used for long-term reference pictures in implicit mode weighted prediction.

Methods for allowing deep temporal hierarchies without the use of long-term reference pictures were presented in [123]. The paper also analyzed the negative impact of the use of long-term reference pictures on compression efficiency when compared to the presented methods. However, none of the methods presented in the paper were adopted in H.264/AVC or its scalable extension.

Reference picture list reordering can be used with hierarchical temporal scalability for improvement of compression efficiency and liberalization of the occurrence of memory management control operations. Reference picture list reordering is often beneficial for arranging used reference pictures into the beginning of the reference picture list to improve compression efficiency. In other words, a reference picture list is reordered such that no "non-existing" picture appears before the used reference pictures in the reference picture list. Reference picture reordering commands make any memory management control operations allowed in sub-sequences as long as only those reference pictures which are explicitly reordered are used for inter prediction.

6.1.2. Sub-Sequence SEI Messages

Three SEI messages are specified in H.264/AVC for sub-sequences: sub-sequence information, sub-sequence layer characteristics and sub-sequence characteristics. These SEI messages were originally proposed in [S5] and [S6]. The SEI messages can be used for different purposes as explained below.

The sub-sequence information SEI message maps a coded picture to a certain sub-sequence and sub-sequence layer. It may also include a frame number that increments by one per each reference frame in the sub-sequence in decoding order. The sub-sequence information message can be used in stream thinning for concluding which coded pictures should be removed as a group. Decoders may use the sub-sequence information message for concluding if pictures got unintentionally removed from a sub-sequence.

The sub-sequence layer characteristics SEI message and the sub-sequence characteristics SEI message give statistical information, such as bitrate, on the indicated sub-sequence layer and sub-sequence, respectively. Furthermore, the dependencies between sub-sequences are indicated in the sub-sequence characteristics SEI message. Gateways can use these messages to conclude which sub-sequences or sub-sequence layers should be removed to obtain a desired bitrate and frame rate. Decoders can use these messages to scale the decoding process computationally in case of lack of computational resources.

6.1.3. Hierarchical Temporal Scalability in H.264/AVC

The goals of the simulations presented in [P6] were to conclude if it is beneficial to use temporal scalability in the H.264/AVC Baseline profile from compression efficiency point of view and if hierarchical temporal scalability provides a competitive alternative to conventional non-hierarchical temporal scalability. To evaluate the coding performance, coded picture patterns IbBbP, IpPpP, IbbP, and IppP, shown in Figure 15(a), (b), (c), and (d), respectively, were compared with each other and the IPPP coded picture pattern in which the maximum number of past reference pictures allowed in the profile and level in use were used as reference pictures. "P" and "p" pictures are inter-coded and "B" and "b" pictures are bi-predicted. A small-case letter indicates a non-reference picture.

The simulations for [P6] were carried out for the following range of picture sizes and frame rates: QCIF 15 Hz, QCIF 30 Hz, CIF 30 Hz, and standard-definition television (SD) at 25 Hz. The size of the decoded picture buffer was selected according to level 1 (QCIF), level 2 (CIF) and level 3 (SD) of H.264/AVC. Each original sequence was coded six times with a constant quantization parameter (QP) value 20, 24, 28, 32, 36 or 40 for all pictures in sub-sequence layer 0 and a constant QP value two units larger than the QP value in sub-sequence layer 0 for all pictures in sub-sequence layer 1.

The simulation results of [P6] are summarized next using Bjontegaard delta bitrate [12] values, which indicate weighted average bitrate savings in terms of percentages when picture quality stays unchanged between the compared streams. Table II presents the bitrate savings of the IbBbP, IpPpP, IbbP, and IppP coding patterns compared to the IPPP coding pattern in terms of Bjontegaard delta bitrate. The results of all test sequences for a given picture size and picture rate were averaged. It can be concluded that both non-hierarchical and hierarchical temporal scalability improve compression efficiency significantly. Furthermore, bi-prediction improves compression efficiency (≥8 percentage units), and sub-sequences (IbBbP) improve compression efficiency compared to IbbP (≥5 percentage units), when picture rate is equal to or greater than 25 Hz. Furthermore, it can be seen that there are no remarkable differences between compression efficiency of IpPpP and IppP or IbBbP and IbbP in QCIF 15 Hz.

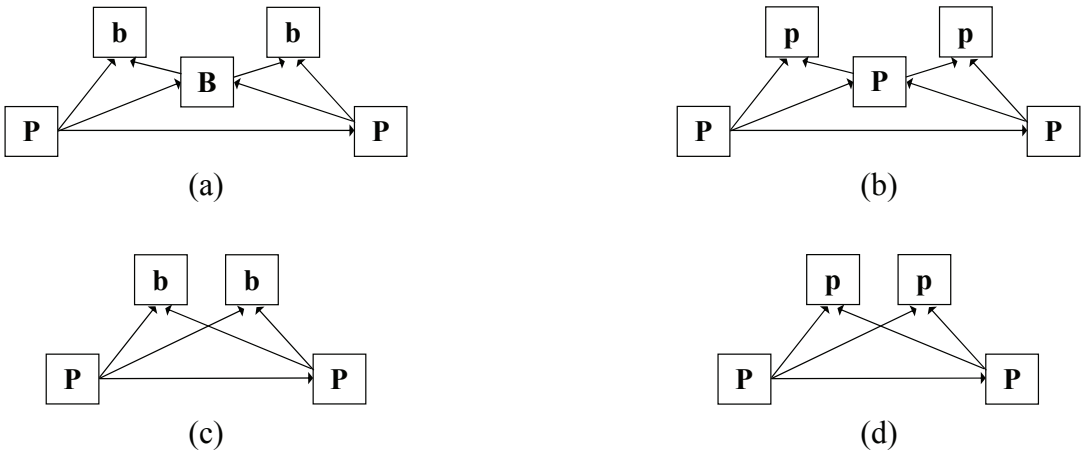


Figure 15. Coding patterns: (a) "IbBbP", (b) "IpPpP", (c) "IbbP", and (d) "IppP"

Table II. Average bitrate saving (%, Bjontegaard Delta Bitrate) compared to non-scalable coding (IPPP)

	QCIF 15 Hz	QCIF 30 Hz	CIF 30 Hz	SD 25 Hz
IppP	7.9	15.0	12.0	3.6
IpPpP	6.0	14.9	13.1	1.2
IbbP	16.6	31.1	27.6	22.4
IbBbP	18.1	36.1	34.9	28.9

The results of [P6] also indicated that the tested sub-sequence schemes provided a larger bitrate range for stream thinning when compared to that of the tested non-hierarchical temporal scalability patterns. Moreover, the tested sub-sequence schemes provided two steps of bitrate scalability that result into a constant picture rate, whereas the IbbP and IppP schemes provided only one such step. On the average, the IpPpP coding scheme provided bitrate steps at constant frame rate at about 50% and about 70% of the full bitrate, whereas the IppP coding scheme could be scaled down to an average of 60% of the full bitrate while maintaining constant frame rate. Similarly, the IbBbP coding scheme provided bitrate steps of about 60% and 80% of the full bitrate, whereas decoding of the reference frames in the IbbP coding scheme resulted into an average of 70% of the full bitrate.

In conclusion, it is beneficial from the compression efficiency point of view to use both non-hierarchical and hierarchical temporal scalability with the H.264/AVC Baseline profile. Furthermore, when bi-prediction is in use, hierarchical temporal scalability outperforms non-hierarchical temporal scalability in terms of compression efficiency and provides a larger range of bitrate scalability compared to non-hierarchical temporal scalability. The presented results have been verified later at least in [119], which also contained simulation results for deeper temporal hierarchies than presented in this section.

6.1.4. Sub-Sequences in Scalable Extension of H.264/AVC

The scalable extension of H.264/AVC uses the sub-sequence coding technique of H.264/AVC as such. The only essential change is the introduction of the *temporal_id* syntax element in the NAL unit headers for SVC NAL units. The *temporal_id* syntax element indicates how deep in the temporal scalability hierarchy the associated NAL unit resides. Correct decoding of a picture at a particular *temporal_id* or temporal layer requires decoding of pictures at and below that temporal layer. The semantics of *temporal_id* are essentially identical to the semantics of the sub-sequence layer. Picture rate is an increasing function of the number of temporal layers or sub-sequence layers.

The *temporal_id* syntax element can also be indicated for VCL NAL units of H.264/AVC with the prefix NAL unit introduced in the scalable extension of H.264/AVC. The prefix NAL unit is associated with the subsequent VCL NAL unit in decoding order and

contains only the SVC NAL unit header. H.264/AVC decoders ignore the prefix NAL units, if they are present.

The sub-sequence SEI messages can be used for SVC with the scalable nesting SEI message. The scalable nesting SEI message contains an ordinary H.264/AVC SEI message and indicates the scalable layers that the message concerns. Consequently, the scalable nesting SEI message enables the reuse of the syntax of H.264/AVC SEI messages for SVC scalable layers.

6.2. RTP PAYLOAD FORMAT FOR H.264/AVC

The RTP payload format specification for H.264/AVC [S12] includes the syntax and semantics of the RTP payload format, RTP packetization rules for H.264/AVC, informative RTP depacketization guidelines, payload type parameters for use in SDP, and guidelines for using the payload type parameters in the SDP offer-answer model for codec capability exchange. The payload format specification contains three packetization modes: single NAL unit mode, non-interleaved mode, and interleaved mode. The single NAL unit mode and the non-interleaved mode provide similar functionality to RTP payload format of other video coding schemes and are therefore reviewed briefly in Section 6.2.1. The interleaved mode provides functions that were not included in any prior RTP payload format and is one of the novelties proposed in this thesis. It is therefore reviewed with more details in Section 6.2.2.

6.2.1. Overview of the Single NAL Unit and Non-Interleaved Packetization Modes

In the single NAL unit packetization mode, one NAL unit is transmitted without any additional payload header in one RTP packet. The single NAL unit mode was introduced to provide compatibility with the packetization scheme specified in the first release of ITU-T Recommendation H.241 [64], which is the specification for H.264/AVC operation in video conferencing systems specified by ITU-T. In the non-interleaved mode, NAL units are transmitted in decoding order and multiple NAL units of one access unit can be encapsulated into the same RTP packet. Encapsulating multiple NAL units into the same RTP packet is especially beneficial when the size of the NAL units is relatively small, which is typically the case for parameter set NAL units and SEI NAL units. The non-interleaved mode therefore helps in reducing the bitrate overhead caused by protocol headers compared to the transmitting relatively small NAL units with the single NAL unit mode.

6.2.2. Overview of the Interleaved Packetization Mode

The interleaved mode allows transmission of NAL units out of NAL unit decoding order and encapsulating of NAL units from different access units into the same RTP packet. In the interleaved mode, a decoding order number (DON) indicating the decoding order of NAL units is conveyed or derived for each NAL unit. The interleaved packetization mode allows for encapsulating NAL units from more than one access unit into the same packet, which helps in

reducing protocol header overhead for low-bitrate streams. Other applications of the interleaved mode include robust packet scheduling for unicast streaming and unequal error protection in broadcast/multicast streaming, which are reviewed in Sections 6.3.1 and 6.3.2, respectively. Furthermore, both the non-interleaved mode and the interleaved mode can be used to match the RTP packet size to the maximum transmission unit size of the underlying network.

In order to enable encapsulating multiple NAL units into the same RTP packet, the RTP payload specification introduces syntax and semantics for aggregation packets. Two types of aggregation packets are defined: single-time aggregation packet (STAP) and multi-time aggregation packet (MTAP). An STAP aggregates NAL units with identical display timestamp, whereas an MTAP aggregates NAL units with potentially differing timestamps.

When interleaved transmission order is used, the decoding order of NAL units must be recovered in the receiver to obtain correct operation of the decoder. The receiver includes a receiver buffer, which is used to compensate for transmission delay jitter and to reorder packets from transmission order to NAL unit decoding order. Compensation for transmission delay jitter and reordering to decoding order can be tackled as separate problems. Therefore, only buffering for deinterleaving is discussed herein. However, receivers should take transmission delay jitter into account in the buffering operation, e.g., by additional initial buffering before the start of decoding and playback.

There are two buffering states for the deinterleaving buffer: initial buffering and buffering while playing. Initial buffering occurs when the RTP session is initialized. The duration of the initial buffering is controlled by certain payload type parameters, the value of which guarantees that the amount of initial buffering is sufficient to recover the decoding order subsequently. After initial buffering, decoding and playback are started and the buffering-while-playing mode is used. Regardless of the buffering state, the receiver stores incoming NAL units in reception order into the deinterleaving buffer. NAL units are removed from the deinterleaving buffer and passed to the decoder in ascending order of DON.

6.3. USE OF SUB-SEQUENCES AND INTERLEAVED TRANSMISSION FOR ERROR ROBUSTNESS

This section presents examples how sub-sequences and interleaved packetization can be used to improve error resilience. Section 6.3.1 discusses how interleaved packetization facilitates robust packet scheduling (see Section 4.2.2). Section 6.3.2 presents a method for unequal error protection to be used for multicast and broadcast streaming. Section 6.3.3 introduces a sub-sequence scheme, known as intra picture postponement, which can be used to improve error resilience in streaming applications.

6.3.1. Bitrate Adaptation and Robust Packet Scheduling

Robust packet scheduling algorithms reviewed in Section 4.2.2 require sending of coded video data in a priority order which differs from the decoding order. No RTP-based mechanism had been specified earlier for interleaved transmission. In other words, the interleaved

packetization mode of H.264/AVC is the first standardized means to use robust packet scheduling algorithms in RTP-based environments. Schierl et al. demonstrated the use of the interleaved packetization mode of H.264/AVC for robust packet scheduling [114][115]. Their work is an extension of the priority based scheduling algorithm by Kampmann and Baldo [77] that uses the 3GPP extended RTCP feedback for receiver buffer status signaling to avoid network and client buffer overflows and to cope with cell handovers gracefully. Schierl et al. took advantage of non-reference pictures of H.264/AVC in priority-based scheduling. Their packet scheduling algorithm aims at achieving certain coded picture buffer occupancy in the receiver for each one of the three significance classes: independent decoding refresh pictures, other reference pictures, and non-reference pictures. Consequently, handovers often result into temporarily dropped picture rate, as the available network throughput is used for obtaining a desired level of IDR and reference picture buffering. The sub-sequence technique, when used for hierarchical temporal scalability as explained in Section 6.1.3, could be used with priority-based packet scheduling to increase the number of priority levels.

6.3.2. Unequal Error Protection in Broadcast/Multicast Streaming

DVB-H and MBMS, discussed in Sections 3.3 and 3.4.3, respectively, provide forward error correction means for equal error protection. However, as discussed in Section 4.1, there are several methods of partitioning video bitstreams to different priorities. Sub-sequences is one of the coding schemes allowing flexible priority partitioning as discussed earlier in this chapter. It was considered worth studying whether UEP applied to priority partitioning based on temporal scalability can be beneficial in MBMS and DVB-H environments. A method for unequal error protection that is compatible with the MBMS and DVB-H services was presented in publication [P7] for the MBMS context and publications [P9] and [P10] for the DVB-H context. The presented UEP method is summarized in the following paragraphs and an example of the presented UEP method is provided in Figure 16.

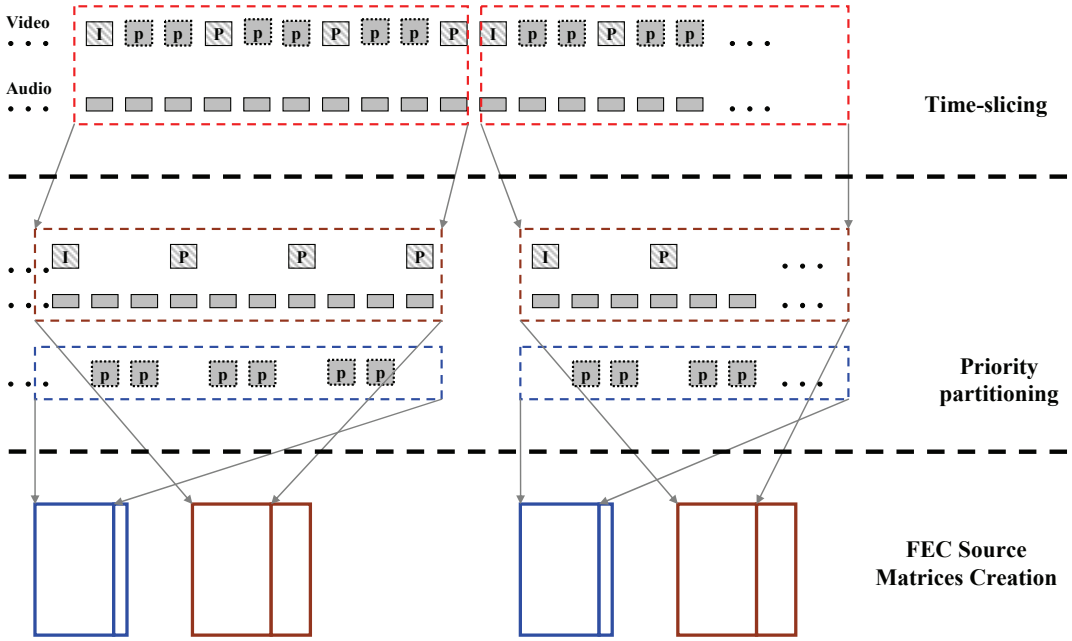


Figure 16. Example of UEP with priority partitioning and interleaved packetization.

The RTP streams of a multimedia service are categorized into a few priority classes. For example, in a news broadcast, the audio RTP packets can have a higher priority than video which in turn has a higher priority than subtitles. Furthermore, datagrams of a single RTP stream can be categorized into priority classes based on their importance for the reconstructed media quality. Without loss of generality, H.264/AVC streams with two temporal layers are considered in this section.

An FEC source block of MBMS and an ADT of MPE-FEC of DVB-H are collectively referred to as the FEC source matrix in this section. The transmitter generates an FEC source matrix for a selected period of media playback that contains data of only one priority class of a multimedia service. An FEC source matrix for each of the other priority classes of the same media playback period is also created. The sizes of FEC source matrices for a media playback period may vary in terms of bytes, as the media bitrate for each priority class may not be equal. The proportion of FEC repair data compared to the occupancy of the FEC source matrix is a function of the priority class, i.e., the largest proportion of FEC repair data is provided for the FEC source matrix of the highest priority class.

MBMS and DVB-H disallow interleaving data of one FEC source matrix with another one in transmission order. Consequently, the interleaved packetization mode of the H.264/AVC RTP payload format is used to arrange packets from decoding order to a transmission order corresponding to consecutive sending of the FEC source matrices. In order to avoid an increase of the initial buffering delay in recipients, the least important FEC source matrix for a period of media playback is transmitted first followed by other FEC source matrices for the same period of media playback in ascending order of importance. When a recipient tunes in and receives at least one but not all of the source FEC matrices for a particular period

of media playback, it can decode and render the period of media playback with reduced quality, such as lower picture rate, compared to the reception of all source FEC matrices.

Publication [S11] compared transmission of non-scalable H.264/AVC Baseline profile stream with equal error protection and temporally scalable H.264/AVC Baseline profile stream with unequal error protection under MBMS environment over UTRAN under 0%, 1%, and 10% RLC PDU loss rates. A subset of the results was also published in publication [P7]. The picture size, picture rate, and total bitrate for video and associated FEC repair packets used in the experiment were QCIF, 7.5 frames per second (fps), and about 44 kilobits per second (kbps), respectively. In conclusion, temporally scalable coding combined with unequal error protection outperformed non-scalable coding with equal error protection by about 0.5 dB in terms of average luma PSNR in the error-free and 1% PDU loss rate cases and by about 0.2 dB in the 10% PDU loss rate case.

Publication [P9] compared the presented UEP method with conventional DVB-H protection under approximated typical urban simulation conditions. The presented UEP method was found to outperform conventional DVB-H transmission by more than 0.5 dB with the tested sequences when the average luma PSNR was about 37 dB or above.

A subjective comparison of the presented UEP scheme and conventional DVB-H transmission was performed for publication [P10]. Several audio-visual streams were processed through a DVB-H channel model for the comparison, and the resulting streams were presented in a comprehensive subjective quality evaluation conducted in a controlled laboratory environment. Two MPE-FEC error rates (MFER) were selected for the evaluation, 6.9% and 13.8%, which resulted into acceptable and unacceptable average quality, respectively, according to a previous study [75]. The results of the evaluation revealed that, at MFER of 6.9%, the presented UEP scheme was at least as good as conventional DVB-H transmission. However, at MFER of 13.8%, the use of the proposed UEP method improved the subjective acceptability of the tested multimedia sequences on average, as the share of participants rating the sequences acceptable was 10 per cent units higher in the UEP case compared to conventional DVB-H transmission.

6.3.3. Intra Picture Postponement

A group of pictures conventionally consists of one chain of reference pictures in which a reference picture is predicted from the earlier reference picture(s) in decoding order. Consequently, one corrupted reference picture affects all subsequent reference pictures in decoding order within the same group of pictures. Temporal scalability reduces the length of inter prediction chains, but the fact that a corrupted picture on the lowest temporal layer generally impacts all subsequent pictures in decoding order remains unchanged. A method called intra picture postponement was proposed in [P1] to reduce the vulnerability of many subsequent inter pictures. The intra picture postponement method is reviewed in this section.

Conventionally, an intra picture is coded immediately after a scene cut or as a response to an expired intra picture refresh period, for example. In the intra picture postponement method, an intra picture is not coded immediately after a need to code an intra picture

arises, but rather a subsequent picture in output order is selected to be coded as an intra picture. Each picture between the coded picture and the conventional location of an intra picture is predicted from the subsequent picture in output order. Figure 17 shows an example of two sequences, one coded conventionally, and another to which intra picture postponement has been applied. The sequence coded using intra picture postponement contains two subsequences, one predicted backwards in output order and a second one predicted conventionally, i.e., in which coding order is identical to output order.

As Figure 17 shows, the intra picture postponement method generates two independent inter picture prediction chains, whereas conventional coding algorithms produce a single inter picture chain. It is intuitively clear that the two-chain approach is more robust against transmission errors than the one-chain conventional approach. If one chain suffers from a packet loss, the other chain may still be correctly received. In conventional coding, a packet loss always causes error propagation to the rest of the inter picture prediction chain.

The intra picture postponement method does not increase the temporal distance between predicted pictures and their reference pictures, but rather it just reverses some of the prediction directions. Thus, intuitively thinking, it should not affect compression efficiency negatively. The simulation results in [P1] indicated a small compression efficiency improvement when intra picture postponement was used.

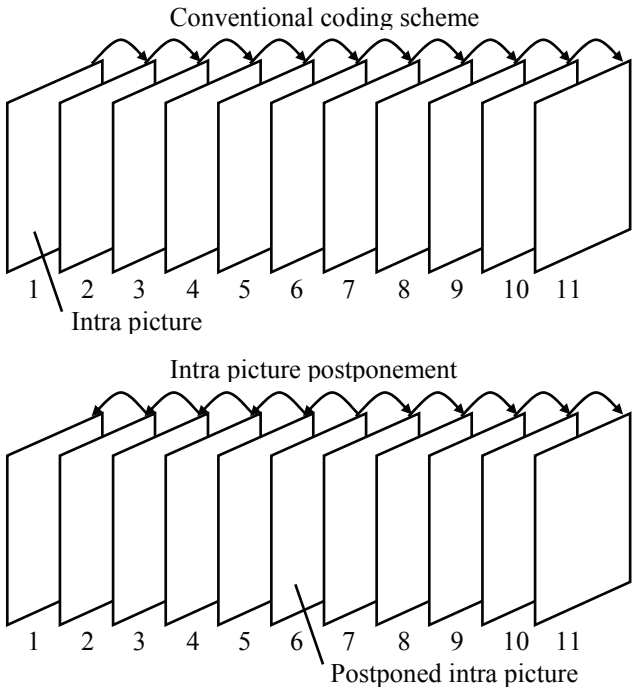


Figure 17. Example of intra picture postponement

The intra picture postponement method requires ordering of decoded pictures to output order, which increases delay and memory requirements. The method is therefore not suitable for low-latency applications. However, many applications, such as unicast and broadcast streaming, are tolerant to decoding delay, which enables the use of the method. As discussed in Section 2.5, H.264/AVC decoders include a decoded picture buffer, which can be exploited by the intra picture postponement method.

It is noted that the intra picture postponement method can be generalized for more complicated coding schemes. For example, intra picture postponement can be used with hierarchical temporal scalability such a way that only the pictures at the lowest temporal layer are considered when deciding the picture coding order for intra picture postponement.

The performed packet loss simulations presented in [P1] indicated intra picture postponement outperformed the conventional coding scheme both in objective and subjective terms. The gain in average luma PSNR was found to be more than 1 dB in many test cases.

Chapter 7

Encoder-Assisted Error Detection and Concealment

Encoder-assisted error detection and concealment techniques were reviewed in Section 4.5.7. In most of these methods, the auxiliary error concealment information helps when a part of the corresponding coded picture is corrupted or lost, and the auxiliary information is attached to the corresponding coded picture. In low bitrate communication or under network conditions that are prone to bursty errors, it is not uncommon that entire pictures are corrupted or lost during transmission. A common novel factor in all encoder-assisted error control methods presented in this chapter is the fact that they all address entire picture losses in addition to partially corrupted pictures.

This chapter reviews those supplemental enhancement information messages of H.264/AVC which fall into the category of assisted error detection and concealment, namely the sub-sequence information SEI message, the scene information SEI message, and the spare picture SEI message. The motivation for each one of these messages is presented briefly below.

When hierarchical temporal scalability is used, a picture loss usually causes only a temporary drop at output picture rate. While the frame sequence numbering, such as the *frame_num* syntax element of H.264/AVC, can be used to detect any picture losses, the sub-sequence information SEI message can be used to conclude in which sub-sequence layer and sub-sequence the loss happened and therefore the impact of the loss to output picture rate can also be estimated. The sub-sequence information SEI message was described in Section 6.1.2.

Many types of video content include frequent scene cuts. When an entire picture is lost during transmission and the lost picture is a scene-cut picture, it is impossible to conceal the lost picture satisfactorily and continuation of decoding would most likely result into dreadful picture quality until the next intra picture or gradual decoding refresh. It is therefore desirable that the receiver is given means to detect losses of scene-cut pictures. Embedded information on scene transitions and their types enables decoders to apply specific error concealment algorithms for particular scene transition types. Furthermore, auxiliary information also saves

computational resources in the decoder, as no scene transition detection algorithm has to be executed, and provides more reliable information on scene changes compared to algorithms executed in the decoder based on partially received pictures. Finally, embedded information on scene changes can also be used for other purposes than error concealment, such as composition of a video summary. The scene information SEI message of H.264/AVC provides embedded information on scene changes. According to received scene information SEI messages, the decoder can infer whether a picture is a scene-cut picture, a gradual scene transition picture or a picture not involved in a scene transition, which can be utilized to help in selecting a proper error concealment method. The scene information SEI message is reviewed in Section 7.1.

One of the fundamental problems of receiver operation in case of erroneous streams is to conclude when the error-concealed decoding result would be subjectively satisfactory for displaying and when it would be better to display the latest correct or satisfying picture instead. In academic literature for video error concealment, it is often just assumed that the goal is to pick the best concealment algorithm even if it would still result into concealed pictures that would be non-acceptable in terms of subjective quality. Often, the error-concealed areas are clearly perceivable and annoying, when movement from the previous picture has been large or non-translational. In contrast, in scenes captured with a stationary camera, a majority of the picture area is often unchanged compared to the previous picture and hence temporal error concealment with zero motion vector recovers unchanged areas perfectly. The spare picture SEI message, introduced in Section 7.2, expresses which areas of indicated pictures are essentially unchanged and can therefore be used complementarily as references for inter prediction. The SEI message helps decoders judge how big a portion of a picture can be reconstructed essentially correctly even if some of the prediction references were error-concealed or lost. Hence, the SEI message helps in concluding whether a decoded picture is good enough for displaying.

7.1. SCENE INFORMATION SEI MESSAGE

This section presents an overview of the encoder-assisted selection of error concealment methods based on the scene information SEI message. The section summarizes publications [P4] and [S7]. The section is organized as follows: Section 7.1.1 provides the terms and definitions related to scene transitions. Section 7.1.2 introduces the scene information SEI message and outlines the encoder operation to create scene information SEI messages. The decoder operation responding to scene information SEI messages is presented in Section 7.1.3. The experimental results of [P4] and [S7] are summarized in Section 7.1.4. Finally, differences between earlier methods and the presented method are discussed in Section 7.1.5.

7.1.1. Definitions for Scene Transitions

A scene transition consists of subsequent pictures over which the video content changes completely from one scene to another scene. A scene, also referred to as a shot, consists of conse-

quent pictures captured with one camera. The scene from which the video content changes is defined as the first scene, and the scene to which the video content changes is defined as the second scene. The set of pictures between the first picture and the last picture in a gradual scene transition, inclusive, is referred to as transition pictures.

Scene transitions can be categorized into abrupt, dissolved, faded, masked, and hybrid ones. An abrupt scene transition (scene cut) is such that the first scene directly changes to the second scene, as shown in Figure 18(a). In a scene cut, the first picture of the second scene is called the scene-cut picture. A dissolved scene transition (dissolve) is such that the pictures of the two scenes in the transition are laid on top of each other in a partially transparent manner, and transparency of the pictures gradually changes in the transition period, as shown in Figure 18(b). A faded scene transition (fade) is such that each picture sample of the first scene gradually changes to the second scene that is of a constant color (fade-in), or each picture sample of the second scene gradually changes from the first scene that is of a constant color (fade-out). If the constant color is black, the fade-in or fade-out becomes fade-to-black or fade-from-black, respectively, as shown in Figure 18(c) and (d), respectively. A masked scene transition is such that the first scene is spatially covered by the second scene while the second scene spatially uncovers from the first scene, both in a gradual manner, and all picture parts are displayed at full intensity. Typical masked scene transitions are wipes, as shown Figure 18(e). A hybrid scene transition is any combination of dissolved, faded and masked transitions.

(a) An example of an abrupt scene transition (scene cut)

(b) An example of a dissolve

(c) An example of a fade to black

(d) An example of a fade from black

(e) An example of a wipe

Figure 18. Example of scene transitions

7.1.2. Encoder Operation

The proposed scene information SEI message associates pictures to particular scenes and scene transitions. In order to generate scene information SEI message, encoders must have knowledge of scene boundaries and transitions. In many cases, the video content has been edited prior to encoding, and consequently the video sequence that is input to the encoder already contains many scenes and transitions between them. Encoders must then execute detection algorithms for scene boundaries and transitions, such as those reviewed and proposed in [50] and [84]. In some cases, encoders may have access to the original camera shots and scene transitions are generated in the encoder from the original shots. Creation of the scene information SEI message is straightforward in such cases.

Each scene information SEI message includes a syntax element *scene_id* to distinguish consecutive scenes in the coded bitstream. A second syntax element is *scene_transition_type*, which indicates in which type of a scene transition, if any, the picture associated with the SEI message is involved. The value of *scene_transition_type* indicates one of the following cases: no transition (0), fade to black (1), fade from black (2), unspecified transition from or to constant color (3), dissolve (4), wipe (5), and unspecified mixture of two scenes (6).

A scene information SEI message should be generated for each access unit. However, for low bitrates it may be desirable to send scene information SEI message only associated with scene transitions as follows. Scene information SEI messages should be generated for each pair of access units having different values of *scene_id* and *scene_transition_type*. If the values of *scene_id* and *scene_transition_type* changed in the previous access unit, a scene information SEI message should also be present in the current access unit in order to guarantee correct reception of at least one occurrence of the message.

7.1.3. Decoder Operation

When a decoder detects a loss or an error, it can either conceal the error in displayed images or freeze the latest correct picture onto the screen until an updated picture is received. The scene information SEI message helps decoders decide a proper action. First, a decoder should infer the type of the erroneous picture according to the received scene information SEI messages. If the erroneous picture is a scene-cut picture and it is lost or largely corrupted, the decoder should stop displaying until an updated picture is decoded. Otherwise, the type of error concealment can be selected as follows. Transmission errors that occurred in a scene-cut picture should be intra-concealed irrespective of the coding type of the scene-cut picture. With this mechanism, the decoder can correctly choose intra error concealment for scene-cut pictures and inter error concealment for intra pictures that are coded for picture refresh or to provide random access points. Moreover, special error concealment algorithms designed for indicated types of gradual scene transitions can be applied to improve error concealment performance. For other cases, conventional error concealment methods, such as those reviewed in Section 4.6, can be applied.

7.1.4. Experimental Results

Two experiments were performed for [P4] and [S7]. In the first experiment, the scene information SEI message and the assisted selection between spatial and temporal error concealment were tested using the VCEG common test conditions for packet-lossy environments [153]. The proposed method improved the performance up to several dBs in terms of average luma PSNR when compared to the JM codec [131]. Detailed results are available in [S7]. The improved concealment of transition pictures was tested in the second experiment. Two sequences with fades to and from black were used in the test. A concealment algorithm for fades was implemented (see [P4] for details), and the JM codec with the proposed selection of the error concealment algorithm and the fade error concealment algorithm was compared against the standard JM codec [131]. As a result of arbitrary picture losses, the proposed method outperformed the JM codec with several dBs in terms of average luma PSNR.

7.1.5. Discussion

When compared to the earlier methods for assisted selection of error concealment algorithms (reviewed in Section 4.5.7), the presented method based on the scene information SEI message has two essential differences. First, rather than specifying exactly which error concealment algorithm is used under a particular condition, the scene information SEI message provides a hint of the type of the error concealment algorithm that should be used. The design of the scene information SEI message therefore leaves possibilities for improving the error concealment algorithms in decoder implementations and gives freedom for decoder developers to optimize their error concealment implementation according to their own criteria. Second, the scene information SEI message is helpful also if entire coded pictures are lost, whereas the earlier methods were applicable to partial losses of coded pictures only.

7.2. SPARE PICTURE SEI MESSAGE

This section presents an overview of the use of the spare picture SEI message for encoder-assisted error concealment. The spare picture concept was first proposed for H.263 in [S1] and later enhanced for H.264/AVC in [S8]. This section summarizes publication [P2]. The section is organized as follows: An introduction to the spare picture SEI message and an outline of codec operations for handling the messages are provided in Section 7.2.1. Then, the experimental results of [P2] are summarized in Section 7.2.2. Finally, the differences between earlier similar methods and the presented method are discussed in Section 7.2.3.

7.2.1. Encoder and Decoder Operation

Sometimes two pictures or respective parts of two pictures resemble each other significantly. Consequently, if one of the pictures is lost or corrupted during transmission, the other picture could be used as an inter prediction reference for subsequent pictures without remarkable quality degradation. This phenomenon is utilized in the spare picture SEI message of

(a) Spare macroblock map

(b) Frame 74 in Hall monitor

(c) Frame 75 in Hall monitor

Figure 19. Example of a spare macroblock map between frames 74 and 75 of the Hall monitor sequence.

H.264/AVC, which indicates that certain macroblocks in an indicated earlier picture, referred to as the target picture, and the listed spare pictures are essentially identical.

The macroblocks that are similar in the target picture and a spare picture are specified by coding a binary spare macroblock map of all macroblock locations of the picture. To improve compression efficiency of coding similar spare macroblock maps, the map can be coded differentially compared to the previous map in the same SEI message. When delivered over a packet network, the spare picture SEI message should not be packetized into the same packet as the target picture in order to avoid the spare picture SEI message to be lost together with the target picture. If a referred block in inter prediction is lost or seriously damaged, decoders may use the co-located block in an indicated spare picture, instead.

Figure 19 shows an example where frame 74 in the Hall monitor sequence, reproduced in Figure 19(b), can be used as a spare picture of frame 75, reproduced in Figure 19(c), except for an area in which the people are moving. Figure 19(a) shows the spare macroblock map between frames 74 and 75. Similar areas which can be used as spare macroblocks are shown in black.

7.2.2. Experimental Results

In the simulations performed for [P2], it was found that a decoder that always continues decoding produces sometimes outputs pictures of unacceptable quality. On the other hand, a decoder that always freezes the latest correct picture produces a lower frame rate compared to a decoder utilizing the spare picture information. Two examples of results can be found in

Table III. Examples of sequences error-concealed based on spare picture information

Hall, QCIF, 64kbps, 15fps, 5% packet loss rate		
Number of correctly decoded frames	77	51%
Number of frozen frames	24	16%
Number of concealed frames thanks to spare reference	49	33%

News, QCIF, 144kbps, 15fps, 3% packet loss rate		
Number of correctly decoded frames	103	69%
Number of frozen frames	34	23%
Number of concealed frames thanks to spare reference	13	9%

Table III indicating that a considerable share of pictures (33% and 9%, respectively) could be satisfactorily concealed based on available spare picture information, but another share of pictures (16% and 23%, respectively) cannot be properly reconstructed with temporal error concealment using zero motion vectors. A comprehensive description of test conditions and a more extensive set of results are available in [P2].

7.2.3. Discussion

The spare picture SEI message could be classified as a method for encoder-assisted motion vector and texture recovery. It indicates the areas in reference pictures for which an error concealment algorithm using zero motion and unmodified texture results into imperceptible differences between error-free decoded pictures and concealed decoded pictures. When compared to the earlier methods for assisted motion vector and texture recovery (reviewed in Section 4.5.7), the presented method has two essential differences. First, the spare picture SEI message is helpful also if entire coded pictures are lost, whereas the earlier methods were applicable to partial losses of coded pictures only. Second, the spare picture SEI message suggests that the quality of concealed areas is close to error-free quality, whereas the earlier methods gave no information about the concealment quality. Consequently, decoders can make more justified decisions whether decoded pictures are subjectively satisfactory for displaying or whether it would be more desirable to display the previous correct picture. Moreover, it is possible to focus feedback messages indicating erroneously recovered picture areas more exactly thanks to spare picture SEI messages and potentially avoid unnecessary feedback messages altogether if transmission errors have hit areas that can be recovered based on spare picture SEI messages.

Chapter 8

Conclusions and Future Work

Compression efficiency and resilience to transmission errors are the major factors affecting video quality in many communication systems, especially when streaming delivery is used. The thesis gave a comprehensive overview of error resilience techniques applicable in video communication systems. The aim of the thesis was to improve the error robustness of H.264/AVC in real-time video communication. Several error resilience methods falling into three categories, isolated regions, sub-sequences and interleaved transmission, and encoder-assisted error concealment, were proposed in this thesis. Support for all the presented source coding methods was adopted into the H.264/AVC standard. Additionally, the interleaved transmission technique was accepted into the RTP payload format for H.264/AVC. The contributions of the thesis are summarized below.

The isolated regions technique is based on limiting in-picture and inter prediction jointly. A similar earlier method in the H.263 standard, namely the independent segment decoding mode of rectangular slices, was identified to have two shortcomings. First, the boundaries of rectangular slices remain unchanged throughout a group of pictures, and hence the technique is not suitable for enclosing moving objects. Second, as the number of macroblocks in a slice is fixed, the slice size cannot be optimized in terms of bytes according to the underlying protocol stack or the prevailing network conditions. Isolated regions are coded as slice groups, and consequently the shortcomings of independent segment decoding of rectangular slices are avoided.

Two error resilience methods utilizing the isolated regions technique were presented: gradual decoding refresh and region-of-interest coding combined with unequal error protection. Gradual decoding refresh updates macroblocks in intra mode gradually within an isolated-region picture group. The method was shown to be useful for providing random access points in an error-prone environment when compared to intra picture coding. Therefore, gradual decoding refresh could be a preferred mechanism to respond to intra picture update requests. Furthermore, gradual decoding refresh was tested as a non-adaptive error-robust macroblock mode selection algorithm. Simulation results showed that gradual decoding refresh is comparable to some other error-robust macroblock mode selection algorithms, and, under

most tested packet loss rates, enhances the performance of a rate-distortion-optimized algorithm when applied together with it. Coding of areas of interest as isolated regions was tested with regular XOR-based FEC coding and was found to improve subjective quality compared to conventional H.264/AVC coding.

The sub-sequence technique enables disposal of inter-dependent reference pictures while the remaining bitstream remains compliant with the H.264/AVC standard. Compared to earlier techniques, especially the reference picture selection in H.263, it manages the reference picture lists for inter prediction such a way that the removal of sub-sequences does not change the decoding process of the remaining bitstream.

The sub-sequence technique can be used for hierarchical temporal scalability among other things. It was shown that hierarchical temporal scalability improves compression efficiency when compared to non-hierarchical temporal scalability and non-scalable bitstreams. The sub-sequence technique remained therefore as one of the fundamental elements of the scalable extension of H.264/AVC.

Interleaved transmission refers to sending of coded picture data in an order that differs from the decoding order. Interleaved video transmission over RTP was introduced in the RTP payload format of H.264/AVC. It can be used together with temporal priority segmentation, e.g., based on sub-sequences, for error robust packet scheduling, which has been shown to work efficiently in unicast streaming applications. When video is priority-segmented temporally and ordered in separate blocks of packets using the interleaved packetization mode, conventional FEC coding schemes can be used to provide unequal error protection as was demonstrated in MBMS and DVB-H environments.

In the intra picture postponement method, two independent inter picture prediction chains are generated instead of a single chain by selecting a later picture as conventionally to be intra coded and coding one of the chains in reverse output order. The method was shown to improve error resilience, but it increases delay and memory requirements compared to conventional coding order.

Two encoder-assisted error concealment methods were proposed, one based on the scene information SEI message and the other based on the spare picture SEI message. In contrast to earlier methods, the presented methods address entire picture losses in addition to partially corrupted pictures. The scene information SEI message indicates which type of a scene transition, if any, pictures are involved with. Consequently, it suggests which type of an error concealment algorithm should be used. The spare picture SEI message indicates that certain macroblocks in an indicated earlier picture and the listed spare pictures are essentially identical. Hence, the spare picture SEI message suggests that the quality of concealed areas is close to error-free quality, whereas the earlier methods gave no information about the concealment quality.

Even though most of the proposed methods of the thesis are supported in standards, some of the presented methods should be developed further for product implementations. For example, for many of the presented methods, there is a need to develop encoding algorithms that are robust in any network environment and utilize information across the layers of the

protocol stack. Furthermore, many of the methods should be tested more thoroughly in various network environments and conditions as well as with more comprehensive application and system environments. Finally, the subjective impact of the methods should be valued against the implementation cost and the requirements for the processing power.

The thesis has concentrated on the analysis of the error resilience impact of the presented methods, leaving other aspects, such as delay considerations, outside of the scope of the thesis. For example, the gradual decoding refresh using isolated regions provides a steadier bitrate compared to the use of intra pictures and could therefore provide a smaller end-to-end delay in some network environments. Moreover, the method for unequal error protection using temporal priority partitioning and interleaved transmission decreases the startup delay when starting the reception of a multicast or broadcast at a random position. Thus, potential future work could provide more analysis on aspects beyond error resilience.

Bibliography

[1] 3GPP Technical Specification 23.246, "Multimedia broadcast/multicast service (MBMS), architecture and functional description, (release 6)," version 6.12.0, Jun. 2007.

[2] 3GPP Technical Specification 26.234, "Transparent end-to-end packet-switched streaming service (PSS), protocols and codecs, (release 6)," version 6.13.0, Mar. 2008.

[3] 3GPP Technical Specification 26.346, "Multimedia broadcast/multicast service (MBMS), protocols and codecs, (release 6)," version 6.11.0, Mar. 2008.

[4] J. Aaltonen, H. Pekonen, T. Auranen, K. Laiho, P. Talmola, "Power saving considerations in mobile datacasting terminals," *Proceedings of IEEE International Symposium on Consumer Electronics (ISCE)*, Sep. 2002.

[5] A. Albanese, J. Blömer, J. Edmonds, M. Luby, and M. Sudan, "Priority encoding transmission," *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 1737-1744, Nov. 1996.

[6] I. Amonou, N. Cammas, S. Kervadec, and S. Pateux, "Optimized rate-distortion extraction with quality layers in the scalable extension of H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1186-1193, Sep. 2007.

[7] J. G. Apostolopoulos, "Error-resilient video compression through the use of multiple states," *Proceedings of IEEE International Conference on Image Processing (ICIP)*, vol. 3, pp. 352-355, Sep. 2000.

[8] P. Baccichet, S. Rane, and B. Girod, "Systematic lossy error protection based on H.264/AVC redundant slices and flexible macroblock ordering," *Journal of Zhejiang University – Science A*, vol. 7, no. 5, pp. 900-909, May 2006.

[9] M. Baldi and Y. Ofek, "End-to-end delay analysis of videoconferencing over packet-switched networks," *IEEE/ACM Transactions on Networking*, vol. 8, no. 4, pp. 479-492, Aug. 2000.

[10] S. Belfiore, M. Grangetto, E. Magli, and G. Olmo, "An edge and texture preserving algorithm for video error concealment," *Proceedings of IEEE Workshop on Multimedia Signal Processing (MMSP)*, pp. 121-124, Dec. 2002.

[11] S. Biaz and N. H. Vaidya, “Discriminating congestion losses from wireless losses using inter-arrival times at the receiver,” *Proceedings of IEEE Symposium on Application-Specific Systems and Software Engineering and Technology (ASSET)*, pp. 10-17, Mar. 1999.

[12] G. Bjontegaard, “Calculation of average PSNR differences between RD-curves,” ITU-T Video Coding Experts Group document VCEG-M33, Mar. 2001 < http://ftp3.itu.ch/av-arch/video-site/0104_Aus/VCEG-M33.doc>.

[13] C. Bormann, L. Cline, G. Deisher, T. Gardos, C. Maciocco, D. Newell, J. Ott, G. Sullivan, S. Wenger, and C. Zhu, “RTP payload format for the 1998 version of ITU-T Rec. H.263 video (H.263+),” IETF Request for Comments 2429, Oct. 1998.

[14] I. Bouazizi, M. M. Hannuksela, and U. Rauf, “Coping with handover effects in video streaming over cellular networks,” *Journal of Zhejiang University - Science A*, vol. 7, suppl. 1, pp. 137-144, May 2006.

[15] M. Budagavi and J. Gibson, “Error propagation in motion compensated video over wireless channels,” *Proceedings of IEEE International Conference on Image Processing (ICIP)*, vol. 2, pp. 89-92, Oct. 1997.

[16] S. Cen and P. Cosman, “Comparison of error concealment strategies for MPEG video,” *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, vol. 1, pp. 329-333, Sep. 1999.

[17] J. Chakareski and B. Girod, “Rate-distortion optimized packet scheduling and routing for media streaming with path diversity,” *Proceedings of Data Compression Conference (DCC)*, pp. 203-212, Mar. 2003.

[18] D.-M. Chiu and R. Jain, “Analysis of the increase and decrease algorithms for congestion avoidance in computer networks,” *Computer Networks and ISDN Systems*, vol. 17, no. 1, pp. 1-14, Jun. 1989.

[19] P. A. Chou and Z. Miao, “Rate-distortion optimized streaming of packetized media,” Microsoft Research Technical Report MSR-TR-2001-35, Feb. 2001.

[20] P. A. Chou, A. E. Mohr, A. Wang, and S. Mehrotra, “Error control for receiver-driven layered multicast of audio and video,” *IEEE Transactions on Multimedia*, vol. 3, no. 1, pp. 108-122, Mar. 2001.

[21] P. A. Chou and Z. Miao, “Rate-distortion optimized streaming of packetized media,” *IEEE Transactions on Multimedia*, vol. 8, no. 2, pp. 390-404, Apr. 2006.

[22] G. J. Conklin and S. Hemami, “A comparison of temporal scalability techniques,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 6, pp. 909-919, Sep. 1999.

[23] G. J. Conklin, G. S. Greenbaum, K. O. Lillevold, A. F. Lippman and Y. A. Reznik, "Video coding for streaming media delivery on the Internet," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 3, pp. 269-281, Mar. 2001.

[24] G. Côté and F. Kossentini, "Optimal intra coding of blocks for robust video communication over the internet," *Signal Processing: Image Communication*, vol. 15, no. 1–2, pp. 25–34, Sep. 1999.

[25] P. Cuenca, T. Olivares, A. Garrido, F. Quiles, and L. Orozco-Barbosa, "Techniques to increase MPEG-2 error resilience in the VBR video transmission over ATM networks," *Proceedings of IEEE International Conference on Communications (ICC)*, vol. 2, pp. 869-873, Jun. 1998.

[26] I. D. D. Curcio and D. Leon, "Application rate adaptation for mobile streaming," *Proceedings of IEEE International Symposium on World of Wireless Mobile and Multimedia Networks (WoWMoM)*, pp. 66-71, Jun. 2005.

[27] I. D. D. Curcio and D. Leon, "Evolution of 3GPP streaming for improving QoS over mobile networks," *Proceedings of IEEE International Conference on Image Processing (ICIP)*, vol. 3, pp. 692-695, Sep. 2005.

[28] S. Deering and R. Hinden, "Internet Protocol, version 6 (IPv6) specification," IETF Request for Comments 2460, Dec. 1998.

[29] Y. Dhondt, P. Lambert, S. Notebaert, and R. Van de Walle, "Flexible macroblock ordering as a content adaptation tool in H.264/AVC," *Proceedings of SPIE*, vol. 6015, doi: 10.1117/12.630759, Oct. 2005.

[30] Y. Dhondt, P. Lambert, and R. Van de Walle, "A flexible macroblock scheme for unequal error protection," *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pp. 829-832, Oct. 2006.

[31] N. Doulamis, A. Doulamis, D. Kalogeras, and S. Kollias, "Low bit-rate coding of image sequences using adaptive regions of interest," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 8, pp. 928-934, Dec. 1998.

[32] C. Dovrolis, D. Tull, and P. Ramanathan, "Hybrid spatial/temporal loss concealment for packet video," *Proceedings of the International Packet Video Workshop*, May 1999.

[33] Y. Eisenberg, F. Zhai, T. N. Pappas, R. Berry, and A. K. Katsaggelos, "VAPOR: Variance-Aware Per-Pixel Optimal Resource Allocation," *IEEE Transactions on Image Processing*, vol. 15, no. 2, pp. 289-299, Feb. 2006.

[34] A. Eleftheriadis and A. Jacquin, "Automatic face location detection for model-assisted rate control in H.261-compatible coding of video," *Signal Processing: Image Communication*, vol. 7, no. 4-6, pp. 435-455, Nov. 1995.

[35] A. Eleftheriadis, M. R. Civanlar, and O. Shapiro, "Multipoint videoconferencing with scalable video coding," *Journal of Zhejiang University – Science A*, vol. 7, no. 5, pp. 696-705, May 2006.

[36] M. Etoh and T. Yoshimura, "Advances in wireless video delivery," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 111-122, Jan. 2005.

[37] ETSI European Standard EN 300 744, "Digital video broadcasting (DVB); framing structure, channel coding and modulation for digital terrestrial television," version 1.6.1, Jan. 2009.

[38] ETSI European Standard EN 301 192, "Digital video broadcasting (DVB); DVB specification for data broadcasting," version 1.4.1, Nov. 2004.

[39] N. Färber and B. Girod, "Robust H.263 compatible video transmission for mobile access to video servers," *Proceedings of IEEE International Conference on Image Processing (ICIP)*, vol. 2, pp. 73-76, Oct. 1997.

[40] G. Faria, J. A. Henriksson, E. Stare, and P. Talmola, "DVB-H: digital broadcast services to handheld devices," *Proceedings of the IEEE*, vol. 94, no. 1, pp. 194-209, Jan. 2006.

[41] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext Transfer Protocol – HTTP/1.1," IETF Request for Comments 2616, Jun. 1999.

[42] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the Internet," *IEEE/ACM Transactions on Networking*, vol. 7, no. 4, pp. 458-472, Aug. 1999.

[43] N. Freed and J. Klensin, "Media type specifications and registration procedures," IETF Request for Comments 4288, Dec. 2005.

[44] P. Fröjdh, U. Horn, M. Kampmann, A. Nohlgren, and M. Westerlund, "Adaptive streaming within the 3GPP packet-switched streaming service," *IEEE Network*, vol. 20, no. 2, pp. 34-40, Mar.-Apr. 2006.

[45] P. Frossard and O. Verscheure, "AMISP: a complete content-based MPEG-2 error-resilient scheme," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 9, pp. 989-998, Sep. 2001.

[46] S. Fukunaga, T. Nakai, and H. Inoue, "Error resilient video coding by dynamic replacing of reference pictures," *Proceedings of Global Telecommunications Conference*, vol. 3, pp. 1503-1508, Nov. 1996.

[47] B. Girod and N. Färber, "Feedback-based error control for mobile video transmission," *Proceedings of the IEEE*, vol. 87, no. 10, pp. 1707-1723, Oct. 1999.

[48] B. Girod, "Efficiency analysis of multihypothesis motion-compensated prediction for video coding," *IEEE Transactions on Image Processing*, vol. 9, no. 2, pp. 173-183, Feb. 2000.

[49] M. Handley and V. Jacobson, "SDP: session description protocol," IETF Request for Comments 2327, Apr. 1998.

[50] A. Hanjalic, "Shot-boundary detection: unraveled and resolved," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 2, pp. 90-105, Feb. 2002.

[51] M. M. Hannuksela, "H.263 picture header recovery in H.324 videophone," *Proceedings of European Signal Processing Conference (EUSIPCO)*, Sep. 2000.

[52] M. M. Hannuksela and Y.-K. Wang, "Coding of parameter sets," Joint Video Team document JVT-C078, May 2002.

[53] M. M. Hannuksela and A. Hourunranta, "Error concealment in a video signal," United States Patent 6,744,924, 1 Jun. 2004.

[54] M. M. Hannuksela, "Data transmission," United States Patent 7,289,506, 30 Oct. 2007.

[55] O. Harmanci and A. M. Tekalp, "Stochastic frame buffers for rate distortion optimized loss resilient video communications," *Proceedings of IEEE International Conference on Image Processing (ICIP)*, vol. 1, pp. 789-792, Sep. 2005.

[56] F. Hartung, U. Horn, J. Huschke, M. Kampmann, T. Lohmar, and M. Lundevall, "Delivery of broadcast services in 3G networks," *IEEE Transactions on Broadcasting*, vol. 53, no. 1, part 2, pp. 188-199, Mar. 2007.

[57] S. S. Hemami and R. M. Gray, "Image reconstruction using vector quantized linear interpolation," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5, pp. 629-632, Apr. 1994.

[58] U. Horn, K. Stuhlmüller, M. Link, and B. Girod, "Robust internet video transmission based on scalable coding and unequal error protection," *Signal Processing: Image Communication*, vol. 15, no. 1, pp. 77-94. Sep. 1999

[59] A. Hourunranta, "Video error resilience in 3G-324M videophones," Licentiate Thesis, Tampere University of Technology, May 2002.

[60] S.-K. Im and A. J. Pearmain, "Unequal error protection with the H.264 flexible macroblock ordering," *Proceedings of SPIE*, vol. 5960, pp. 1033-1040, Jul. 2005.

[61] ISO/IEC International Standard 11172-2:1995, "Information technology – coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – part 2: video."

[62] ISO/IEC International Standard 14496-2:2001, “Information technology – coding of audio-visual objects – part 2: visual.”

[63] ISO/IEC International Standard 14496-5:2001/Amd.6:2005, “Reference software for AVC and audio HE-AAC.”

[64] ITU-T Recommendation H.241 (05/2006), “Extended video procedures and control signals for H.300-series terminals.”

[65] ITU-T Recommendation H.261 (03/93), “Video codec for audiovisual services at p × 64 kbits.”

[66] ITU-T Recommendation H.262 (02/2000) | ISO/IEC International Standard 13818-2:2000, “Information technology – generic coding of moving pictures and associated audio information: video.”

[67] ITU-T Recommendation H.263 (01/2005), “Video coding for low bit rate communication.”

[68] ITU-T Recommendation H.264 (05/2003), “Advanced video coding for generic audiovisual services.”

[69] ITU-T Recommendation H.264 (03/2005), “Advanced video coding for generic audiovisual services.”

[70] ITU-T Recommendation H.264 (11/2007), “Advanced video coding for generic audiovisual services.”

[71] ITU-T Recommendation H.264.2 (03/2005), “Reference software for H.264/AVC advanced video coding.”

[72] ITU-T Recommendation H.271 (05/2006), “Video back channel messages for conveyance of status information and requests from a video receiver to a video sender.”

[73] H. Jenkac, T. Stockhammer, and W. Xu, “Permeable-Layer Receiver for Reliable Multicast Transmission in Wireless Systems,” *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, vol. 3, pp. 1805-1811, Mar. 2005.

[74] S. R. Joshi and I. Rhee, “Lazy hybrid packet-loss recovery for video transmission”, *Proceedings of the International Packet Video Workshop*, May 2000.

[75] S. Jumisko-Pyykkö, V. K. Malamal Vadakital, M. Liinasuo, and M. M. Hannuksela, “Acceptance of audiovisual quality in erroneous television sequences over a DVB-H channel,” *Proceedings of the International Workshop on Video Processing and Quality Metrics for Consumer Electronics (VPQM)*, Jan. 2006.

[76] M. Kalman, P. Ramanathan, and B. Girod, "Rate-distortion optimized video streaming with multiple deadlines," *Proceedings of IEEE International Conference on Image Processing (ICIP)*, vol. 3, pp. 661-664, Sep. 2003.

[77] M. Kampmann and N. Baldo, "Adaptive wireless video streaming using transmission rate control and priority-based packet scheduling," *Proceedings of the International Packet Video Workshop*, Dec. 2004.

[78] S. H. Kang and A. Zakhor, "Packet scheduling algorithm for wireless video streaming," *Proceedings of the International Packet Video Workshop*, Apr. 2002.

[79] S. H. Kang and A. Zakhor, "Effective bandwidth based scheduling for streaming media," *IEEE Transactions on Multimedia*, vol. 7, no. 6, pp. 1139-1148, Dec. 2005.

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

[81] W. Karner and M. Rupp, "Measurement based analysis and modeling of UMTS DCH error characteristics for static scenarios," *Proceedings of International Symposium on DSP and Communications Systems (DSPCS) and Workshop on the Internet, Telecommunications and Signal Processing (WITSP)*, Dec. 2005.

[82] C.-S. Kim, R.-C. Kim, and S.-U. Lee, "Robust transmission of video sequence using double-vector motion compensation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 9, pp. 1011-1021, Sep. 2001.

[83] R. Koodli and M. Puuskari, "Supporting packet-data QoS in next-generation cellular networks," *IEEE Communications Magazine*, vol. 39, no. 2, pp. 180-188, Feb. 2001.

[84] I. Koprinska and S. Carrato, "Temporal video segmentation: A survey," *Signal Processing: Image Communication*, vol. 16, no. 5, pp. 477-500, Jan. 2001.

[85] W. M. Lam, A. R. Reibman, and B. Liu, "Recovery of lost or erroneously received motion vectors," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5, pp. 417-420, Apr. 1993.

[86] S. Lee, C. Podilchuck, and A. C. Bovik, "Foveation-based error resilience for video transmission over mobile networks," *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, vol. 3, pp. 1451-1454, Jul.-Aug. 2000.

[87] C. Leicher, "Hierarchical encoding of MPEG sequences using priority encoding transmission (PET)," International Computer Science Institute technical report TR-94-058, Nov. 1994.

[88] A. Leontaris and P. C. Cosman, "Video compression for lossy packet networks with mode switching and a dual-frame buffer," *IEEE Transactions on Image Processing*, vol. 13, no. 7, pp. 885-897, Jul. 2004.

[89] A. Li (ed.), "RTP payload format for generic forward error correction," IETF Request for Comments 5109, Dec. 2007.

[90] J. Y. Liao and J. D. Villasenor, "Adaptive intra update for video coding over noisy channels," *Proceedings of IEEE International Conference on Image Processing (ICIP)*, vol. 3, pp. 763-766, Oct. 1996.

[91] M. Luby, M. Watson, T. Gasiba, T. Stockhammer, and W. Xu, "Raptor codes for reliable download delivery in wireless broadcast systems," *Proceedings of IEEE Consumer Communications and Networking Conference*, vol. 1, pp. 192-197, Jan. 2006.

[92] V. K. Malamal Vadakital, M. M. Hannuksela, M. Rezaei, and M. Gabbouj, "Optimal IP packet size for efficient data transmission in DVB-H," *Proceedings of the 7th Nordic Signal Processing Symposium (NORSIG)*, pp. 82-85, Jun. 2006.

[93] Z. Miao and A. Ortega, "Optimal scheduling for streaming of scalable media," *Proceedings of Asilomar Conference of Signals, Systems, and Computers*, vol. 2, pp. 1357-1362, Nov. 2000.

[94] Z. Miao and A. Ortega, "Expected run-time distortion based scheduling for delivery of scalable media," *Proceedings of the International Packet Video Workshop*, Apr. 2002.

[95] O. Ndili and T. Ogunfunmi, "On the performance of a 3D flexible macroblock ordering for H.264/AVC," *Proceedings of International Conference on Consumer Electronics (ICCE)*, pp. 37-38, Jan. 2006.

[96] J.-R. Ohm, "Advances in scalable video coding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 42-56, Jan. 2005.

[97] J. Ott, S. Wenger, N. Sato, C. Burmeister, and J. Rey, "Extended RTP profile for real-time transport control protocol (RTCP)-based feedback (RTP/AVPF)," IETF Request for Comments 4585, Jul. 2006.

[98] V. Paxson, "End-to-end Internet packet dynamics," *IEEE/ACM Transactions on Networking*, vol. 7, no.3, pp. 277-292, Jun. 1999.

[99] J. Postel (editor), "DoD standard Internet protocol," IETF Request for Comments 760, Jan. 1980.

[100] J. Postel, "User datagram protocol," IETF Request for Comments 768, Aug. 1980.

[101] A. Puri, X. Chen, and A. Luthra, "Video coding using the H.264/MPEG-4 AVC compression standard," *Signal Processing: Image Communication*, vol. 19, no. 9, pp. 793-849, Oct. 2004.

[102] H. Radha and D. Loguinov, "Channel modeling and analysis for the Internet," in M. van der Schaar and P. A. Chou (ed.), "Multimedia over IP and wireless networks," Elsevier, 2007.

[103] I. Radulovic, Y.-K. Wang, S. Wenger, A. Hallapuro, M. M. Hannuksela, and P. Frossard, "Multiple description H.264 video coding with redundant pictures," *Proceedings of Mobile Video Workshop, ACM Multimedia 2007*, pp. 37-42, Sep. 2007.

[104] S. Rane, A. Aaron, and B. Girod, "Systematic lossy forward error protection for error-resilient digital video broadcasting - a Wyner-Ziv coding approach," *Proceedings of IEEE International Conference on Image Processing (ICIP)*, vol. 5, pp. 3101-3104, Oct. 2004.

[105] I. S. Reed and G. Solomon, "Polynomial Codes over Certain Finite Fields," *SIAM Journal of Applied Mathematics*, vol. 8, pp. 300-304, 1960.

[106] J. Rey, D. Leon, A. Miyazaki, V. Varsa, and R. Hakenberg, "RTP retransmission payload format," IETF Request for Comments 4588, Jul. 2006.

[107] I. Rhee, "Error control techniques for interactive low-bit rate video transmission over the Internet," *Proceedings of ACM SIGCOMM Computer Communication Review*, vol. 28, no. 4, pp. 290-301, Oct. 1998.

[108] I. Rhee and S. R. Joshi, "Error recovery for interactive video transmission over the Internet," *IEEE Journal on Selected Areas of Communications*, vol. 18, no. 6, Jun. 2000.

[109] J. Rosenberg and H. Schulzrinne, "An RTP payload format for generic forward error correction," IETF Request for Comments 2733, Dec. 1999.

[110] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: session initiation protocol," IETF Request for Comments 3261, Jun. 2002.

[111] J. Rosenberg and H. Schulzrinne, "An offer/answer model with the session description protocol (SDP)," IETF Request for Comments 3264, Jun. 2002.

[112] P. Salama, N. B. Shroff, E. J. Coyle, and E. J. Delp, "Error concealment techniques for encoded video streams," *Proceedings of IEEE International Conference on Image Processing (ICIP)*, vol. 1, pp. 9-12, Oct. 1995.

[113] M. van der Schaar and S. Shankar N, "Cross-layer wireless multimedia transmission: challenges, principles, and new paradigms," *IEEE Wireless Communications*, vol. 12, no. 4, pp. 50-58, Aug. 2005.

[114] T. Schierl, M. Kampmann, and T. Wiegand, "H.264/AVC interleaving for 3G wireless video streaming," *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, Jul. 2005.

[115] T. Schierl, T. Wiegand, and M. Kampmann, "3GPP compliant adaptive wireless video streaming using H.264/AVC," *Proceedings of IEEE International Conference on Image Processing (ICIP)*, vol. 3, pp. 696-699, Sep. 2005.

[116] H. Schulzrinne, A. Rao, and R. Lanphier, "Real time streaming protocol (RTSP)," IETF Request for Comments 2326, Apr. 1998.

[117] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: a transport protocol for real-time applications," IETF Request for Comments 3550, Jul. 2003.

[118] H. Schulzrinne and S. Casner, "RTP profile for audio and video conferences with minimal control," IETF Request for Comments 3551, Jul. 2003.

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

[120] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103-1120, Sep. 2007.

[121] E. Setton and B. Girod, "Congestion-distortion optimized scheduling of video over a bottleneck link," *Proceedings of IEEE Workshop on Multimedia Signal Processing (MMSP)*, pp. 179-182, Sep.-Oct. 2004.

[122] Y. Shan, "Cross-layer techniques for adaptive video streaming over wireless networks," *EURASIP Journal on Applied Signal Processing*, vol. 2005:2, pp. 220-228.

[123] Q. Shen, Y.-K. Wang, M. M. Hannuksela, H. Li, and Y. Wang, "Buffer requirement analysis and reference picture management for temporal scalable video coding," *Proceedings of the International Packet Video Workshop*, pp. 91-97, Nov. 2007

[124] S. Shirani, F. Kossentini, and R. Ward, "Reconstruction of motion vector missing macroblocks in H.263 encoded video transmission over lossy networks," *Proceedings of IEEE International Conference on Image Processing (ICIP)*, vol. 3, pp. 487-491, Oct. 1998.

[125] J. Song and K. J. R. Liu, "A data embedding scheme for H.263 compatible video coding," *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 4, pp. 390-393, Jul. 1999.

[126] E. Steinbach, N. Färber, and B. Girod, "Standard compatible extension of H.263 for robust video transmission in mobile environments," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 6, pp. 872-881, Dec. 1997.

[127] T. Stockhammer, D. Kontopodis, and T. Wiegand, "Rate-distortion optimization for JVT/H. 26L video coding in packet loss environment," *Proceedings of the Packet Video Workshop*, May 2002.

[128] T. Stockhammer, T. Wiegand, T. Oelbaum, and F. Obermeier, "Video coding and transport layer techniques for H.264/AVC-based transmission over packet-lossy networks," *Proceedings of IEEE International Conference on Image Processing (ICIP)*, vol. 3, pp. 481-484, Sep. 2003.

[129] T. Stockhammer, "Robust system and cross-layer design for H.264/AVC-based wireless video applications," *EURASIP Journal on Applied Signal Processing*, vol. 2006, article ID 89371, pp. 1-15.

[130] J.-W. Suh and Y.-S. Ho, "Error concealment based on directional interpolation," *IEEE Transactions on Consumer Electronics*, vol. 43, no. 3, pp. 295-302, Aug. 1997.

[131] K. Sühring (coordinator), "Joint model H.264/AVC reference software," http://iphome.hhi.de/suehring/tml/.

[132] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 74-90, Nov. 1998.

[133] G. J. Sullivan and G. Bjontegaard, "Recommended simulation common conditions for H.26L coding efficiency experiments on low-resolution progressive-scan source material," ITU-T Video Coding Experts Group document VCEG-N81, Sep. 2001 <http://ftp3.itu.ch/av-arch/video-site/0109_San/VCEG-N81.doc>.

[134] H. Sun and W. Kwok, "Concealment of damaged block transform coded images using projections onto convex sets," *IEEE Transactions on Image Processing*, vol. 4, no. 4, pp. 470-477, Apr. 1995.

[135] H. Sun, J. W. Zdepski, W. Kwok, and D. Raychaudhuri, "Error concealment algorithms for robust decoding of MPEG compressed video", *Signal Processing: Image Communication*, vol. 10, no. 4, pp. 249-268, Sep. 1997.

[136] S. Valente, C. Dufour, F. Grolière, and D. Snook, "An efficient error concealment implementation for MPEG-4 video streams," *IEEE Transactions on Consumer Electronics*, vol. 47, no. 3, pp. 568-578, Aug. 2001.

[137] V. Varsa and M. Karczewicz, "Slice interleaving in compressed video packetization," *Proceedings of the International Packet Video Workshop*, May 2000.

[138] A. Vetro, P. Pandit, H. Kimata, A. Smolic, and Y.-K. Wang (editors), "Joint draft 8.0 on multiview video coding," Joint Video Team document JVT-AB204 (rev. 1), Nov. 2008.

[139] W3C Recommendation, "Synchronized multimedia integration language (SMIL 2.0) – [second edition]," Jan. 2005.

[140] W3C Candidate Recommendation, "Scalable vector graphics (SVG) tiny 1.2 specification," Aug. 2006.

[141] M. Wada, "Selective recovery of video packet loss using error concealment," *IEEE Journal on Selected Areas in Communications*, vol. 7, no. 5, pp. 807-814, Jun. 1989.

[142] Y. Wang, Q.-F. Zhu, and L. Shaw, "Maximally smooth image recovery in transform coding," *IEEE Transactions on Communications*, vol. 41, no. 10, pp. 1544-1551, Oct. 1993.

[143] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communication: a review," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 974-997, May 1998.

[144] Y. Wang, S. Wenger, J. Wen, and A. K. Katsaggelos, "Error resilient video coding techniques," *IEEE Signal Processing Magazine*, vol. 17, no. 4, pp. 61-82, Jul. 2000.

[145] Y. Wang, A. R. Reibman, and S. Lin, "Multiple description coding for video delivery," *Proceeding of the IEEE*, vol. 93, no. 1, pp. 57-70, Jan. 2005.

[146] Y.-K. Wang, M. M. Hannuksela, V. Varsa, A. Hourunranta, and M. Gabbouj, "The error concealment feature in the H.26L test model," *Proceedings of IEEE International Conference on Image Processing (ICIP)*, vol. 2, pp. 729-732, Sep. 2002.

[147] Y.-K. Wang, M. M. Hannuksela, and M. Gabbouj, "Error resilient video coding using unequally protected key pictures," *Proceedings of International Workshop VLBV03, Lecture Notes in Computer Science*, vol. 2849/2003, pp. 290-297, DOI 10.1007/b13938, Springer Berlin / Heidelberg, Sep. 2003.

[148] L. F. Wei, "Coded modulation with unequal error protection," *IEEE Transactions on Communications*, vol. 41, no. 10, pp. 1439-1449, Oct. 1993.

[149] S. Wenger, "Video redundancy coding in H.263+," *Proceedings of the International Workshop on Audio-Visual Services over Packet Networks*, Sep. 1997.

[150] S. Wenger, "Temporal scalability using P-pictures for low-latency applications," *Proceedings of IEEE Workshop on Multimedia Signal Processing (MMSP)*, pp. 559-564, Dec. 1998.

[151] S. Wenger, G. Côté, and M. Gallant, "Why do we need normative text for error concealment: a justification," ITU-T Video Coding Experts Group document Q15-I-17, Oct. 1999 <http://ftp3.itu.ch/av-arch/video-site/9910_Red/q15i17.doc>.

[152] S. Wenger and T. Stockhammer, "H.26L over IP and H.324 framework," ITU-T Video Coding Experts Group document VCEG-N52, Sep. 2001 <http://ftp3.itu.ch/av-arch/video-site/0109_San/VCEG-N52.doc>.

[153] S. Wenger, "Common conditions for wire-line, low delay IP/UDP/RTP packet loss resilient testing," ITU-T Video Coding Experts Group document VCEG-N79, Sep. 2001 <http://ftp3.itu.ch/av-arch/video-site/0109_San/VCEG-N79r1.doc>.

[154] S. Wenger and M. Horowitz, "Flexible macroblock ordering: a new error resilience tool for IP based video," *Proceedings of Tyrrhenian International Workshop on Digital Communications (IWDC)*, Sep. 2002.

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

[156] S. Wenger, U. Chandra, M. Westerlund, and B. Burman, "Codec control messages in the audio-visual profile with feedback (AVPF)," IETF Request for Comments 5104, Feb. 2008.

[157] T. Wiegand, X. Zhang, and B. Girod, "Long-term memory motion-compensated prediction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 1, pp. 70-84, Feb. 1999.

[158] T. Wiegand, N. Färber, K. Stuhlmüller, and B. Girod, "Error-resilient video transmission using long-term memory motion-compensated prediction," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 1050-1062, Jun. 2000.

[159] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 688-703, Jul. 2003.

[160] A. Wyner, "On source coding with side information at the decoder," *IEEE Transactions on Information Theory*, vol. 21, no. 3, pp. 294-300, May 1975.

[161] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Transactions on Information Theory*, vol. 22, no. 1, pp. 1-10, Jan. 1976.

[162] R. Yallapragada, P. Sagetong, H. Garudadri, and N. Srinivasamurthy, "Video delivery over cellular wireless networks using EBR techniques," *Proceedings of IEEE International Conference on Personal Wireless Communications (ICPWC)*, pp. 249-253, Jan. 2005.

[163] H. Yang and K. Rose, "Recursive end-to-end distortion estimation with model-based cross-correlation approximation," *Proceedings of IEEE International Conference on Image Processing (ICIP)*, vol. 3, pp. 469-472, Sep. 2003.

[164] P. Yin, B. Liu, and H. H. Yu, "Error concealment using data hiding," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 3, pp. 1453-1456, May 2001.

[165] R. Zhang, S. L. Regunathan, and K. Rose, "Robust video coding for packet networks with feedback," *Proceedings of Data Compression Conference (DCC)*, pp. 450-459, Mar. 2000.

[166] R. Zhang, S. L. Regunathan, and K. Rose, "Video coding with optimal inter/intra-mode switching for packet loss resilience," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 966-976, Jun. 2000.

[167] R. Zhang, S. L. Regunathan, and K. Rose, "Prescient mode selection for robust video coding," *Proceedings of IEEE International Conference on Image Processing (ICIP)*, vol. 1, pp. 974-977, Oct. 2001.

[168] Y. Zhang, W. Gao, H. Sun, Q. Huang, and Y. Lu, "Error resilience video coding in H.264 encoder with potential distortion tracking," *Proceedings of IEEE International Conference on Image Processing (ICIP)*, vol. 1, pp. 163-166, Oct. 2004.

[169] Y. Zhao, D. Tian, M. M. Hannuksela, and M. Gabbouj, "Spatial Error Concealment Based on Directional Decision and Intra Prediction," *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2899-2902, May 2005.

[170] C. Zhu, Y.-K. Wang, M. M. Hannuksela, and H. Li, "Error resilient video coding using redundant pictures," *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pp. 801-804, Oct. 2006.

[171] C. Zhu, Y.-K. Wang, and H. Li, "Adaptive redundant picture coding," Joint Video Team document JVT-U114, Oct. 2006 <http://ftp3.itu.ch/av-arch/jvt-site/2006_10_Hangzhou/JVT-U114.zip>.

[172] C. Zhu, Y.-K. Wang, M. M. Hannuksela, and H. Li, "Error resilient video coding using redundant pictures," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 1, pp. 3-14, Jan. 2009.

[173] Q.-F. Zhu, "Device and method of signal loss recovery for realtime and/or interactive communications," United States Patent 5,550,847, 27 Aug. 1996.

[174] H. Zimmermann, "OSI reference model – the ISO model of architecture for open systems interconnection," *IEEE Transactions on Communications*, vol. 28, no. 4, pp. 425-432, Apr. 1980.

Publications

[P1] M. M. Hannuksela, “Simple packet loss recovery method for video streaming,” *Proceedings of the 11th International Packet Video Workshop*, pp. 138-143, Apr. 2001.

Proceedings of The 11th International Packet Video Workshop, 30 April – 1 May, 2001 · Kyungju, Korea

SIMPLE PACKET LOSS RECOVERY METHOD FOR VIDEO STREAMING

Miska M. Hannuksela
Nokia Mobile Phones,
P.O.Box 68, 33721 Tampere, FINLAND
Tel: +358 40 5212845; Fax: +358 10 5057662
E-mail: miska.hannuksela@nokia.com

ABSTRACT

Streaming applications can have relaxed transmission delay requirements, since two-way real-time conversation is not needed. This enables the usage of interactive error control schemes, such as selective retransmission of lost transmission packets. However, interactive error control cannot be widely used in IP multicast streaming due to a risk of an excessive number of retransmission requests. Thus, multicast streaming systems should tackle packet loss recovery with non-interactive techniques. Furthermore, point-to-point streaming systems may also benefit from non-interactive error control methods, because certain factors, such as large transmission delay, may reduce the efficiency of interactive error control techniques. This paper presents a simple non-interactive video coding method, called INTRA frame postponement, which improves error resilience in streaming applications. We show that the method is superior to the conventional coding scheme in terms of objective and subjective quality, when video sequences are transmitted over loss-prone networks. Moreover, our simulations indicate that the method has a positive impact on compression efficiency in certain cases. We also analyze the application of the method to the current video communication standards.

1 INTRODUCTION

1.1 Streaming Systems

A multimedia streaming system consists of a streaming server and a number of players, which access the server via a network. The players fetch either pre-stored or live multimedia content from the server and play it back in real-time while the content is being downloaded. The type of communication can be either point-to-point or multicast. In point-to-point streaming, the server provides a separate connection for each player. In multicast streaming, the server transmits a single data stream to a number of players, and network elements duplicate the stream only if it is necessary.

When a player has established a connection to a server and requested for a multimedia stream, the server begins to transmit the desired stream. The player does not start playing the stream back immediately, but rather it typically buffers the incoming data for a few seconds. Herein, this buffering is referred to as initial buffering. Initial buffering helps to maintain pauseless playback, because, in case of occasional increased transmission delays or network throughput drops, the player can decode and play buffered data.

In order to avoid unlimited transmission delay, it is uncommon to favor reliable transport protocols in streaming systems. Instead, the systems prefer unreliable transport protocols, such as UDP, which, on one hand, inherit a more stable transmission delay, but, on the other hand, also suffer from data corruption or loss.

RTP and RTCP protocols can be used on top of UDP to control real-time communications. RTP provides means to detect losses of transmission packets, to reassemble the correct order of packets in the receiving end, and to associate a sampling time-stamp with each packet. RTCP conveys information about how large a portion of packets were correctly received, and, therefore, it can be used for flow control purposes.

1.2 Handling of Packet Losses

Packet losses can either be corrected or concealed. Loss correction refers to the capability to restore lost data perfectly as if no losses had ever been introduced. Loss concealment refers to the capability to conceal the effects of transmission losses so that they should not be visible in the reconstructed video sequence.

When a player detects a packet loss, it may request for a packet retransmission. Because of the initial buffering, the retransmitted packet may be received before its scheduled playback time. Some commercial Internet streaming systems implement retransmission requests using proprietary protocols. Work is going on in IETF to standardize a selective retransmission request mechanism as a part of RTCP [1][2][3][4]. A common feature for all of these retransmission request protocols is that they are not suitable for multicasting to a large number of players, as the network traffic may increase drastically. Consequently, multicast streaming applications have to rely on non-interactive packet loss control.

Point-to-point streaming systems may also benefit from non-interactive error control techniques. First, some systems may not contain any interactive error control mechanism or they prefer not to have any feedback from players in order to simplify the system. Second, retransmission of lost packets and other forms of interactive

error control typically take a larger portion of the transmitted data rate than non-interactive error control methods. Streaming servers have to ensure that interactive error control methods do not reserve a major portion of the available network throughput. In practice, the servers may have to limit the amount of interactive error control operations. Third, transmission delay may limit the number of interactions between the server and the player, as all interactive error control operations for a specific data sample should preferably be done before the data sample is played back.

Non-interactive packet loss control mechanisms can be categorized to forward error control and loss concealment by post-processing. Forward error control refers to techniques in which a transmitter adds such redundancy to transmitted data that receivers can recover at least part of the transmitted data even if there are transmission losses. Error concealment by post-processing is totally receiver-oriented. These methods try to estimate the correct representation of erroneously received data.

Most video compression algorithms generate temporally predicted INTER or P pictures. As a result, a data loss in one picture causes visible degradation in the consequent pictures that are temporally predicted from the corrupted one. Video communication systems can either conceal the loss in displayed images or freeze the latest correct picture onto the screen until a frame which is independent from the corrupted frame is received.

A simple forward error control mechanism to reduce the duration of a visible error caused by a packet loss is to decrease the length of INTER picture prediction paths. Temporal scalability techniques can be used to achieve this target. Temporal scalability refers to the capability to decode the same bit-stream with different picture rates.

1.3 Overview of the Paper

This paper focuses on video error resilience in streaming applications and introduces a simple forward error control method. The method, which is referred to as INTRA frame postponement, can be categorized as a temporal scalability technique.

The paper is organized as follows: Section 2 reviews temporally scalable video coding methods. The INTRA frame postponement scheme is described in section 3, and the performed simulations are presented in section 4. Then, in section 5, we analyze how to implement the proposed coding method in today's streaming systems. Finally, section 6 concludes the paper.

2 TEMPORALLY SCALABLE VIDEO CODING

INTRA pictures can be decoded independently from INTER pictures, and this can be considered as a basic form of temporal scalability. There are two reasons why INTRA frames are commonly used in video streaming. First, INTRA frames are frequently introduced to prevent packet loss artifacts from propagating temporally. Second, INTRA frames enable new receivers to start the decoding of the stream in multicast streaming. Video retrieval systems typically enable insertion of INTRA frames periodically. In addition, it is advantageous to utilize natural scene cuts where the image content changes so drastically that temporal prediction from the previous image is not likely to succeed well.

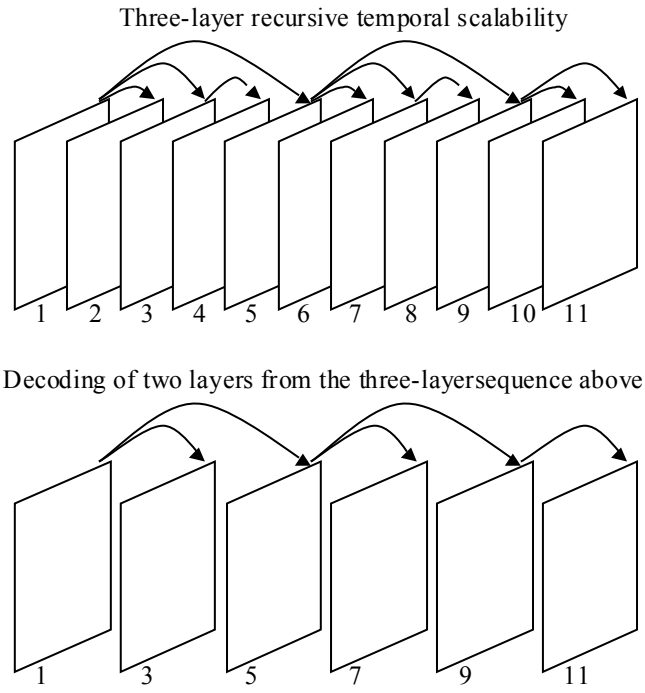


Figure 1. An example of recursive temporal scalability.

Most video compression schemes include temporally bidirectionally predicted frames which are commonly referred to as B-pictures or B-frames. B-pictures are inserted between two consecutive INTRA or INTER pictures and predicted from either one or both of these pictures. B-pictures may be beneficial from compression efficiency point of view when compared to INTER pictures. However, when the time between the reference frames for a B-picture increases, the compression efficiency decreases, as the reference frames are less similar to the B-picture and a worse predicted B-frame can be obtained. INTRA or INTER pictures are never predicted from B-pictures. Therefore, B-pictures can be discarded for temporal scalability purposes.

Modern video coding standards, such as ITU-T H.263 and ISO/IEC MPEG-4, enable selection of the reference frame for motion compensation per each picture segment, e.g., per each slice in H.263. Furthermore, the Enhanced Reference Picture Selection mode of H.263 enables selection of the reference frame for each macroblock separately. Reference picture selection enables many types of temporal scalability schemes. Figure 1 shows an example of a temporal scalability scheme which is herein referred to as recursive temporal scalability. The example scheme can be decoded with three constant frame rates. Figure 2 depicts a scheme referred to as Video Redundancy Coding [5], where a sequence of pictures is divided into two or more independently coded threads in an interleaved manner. The

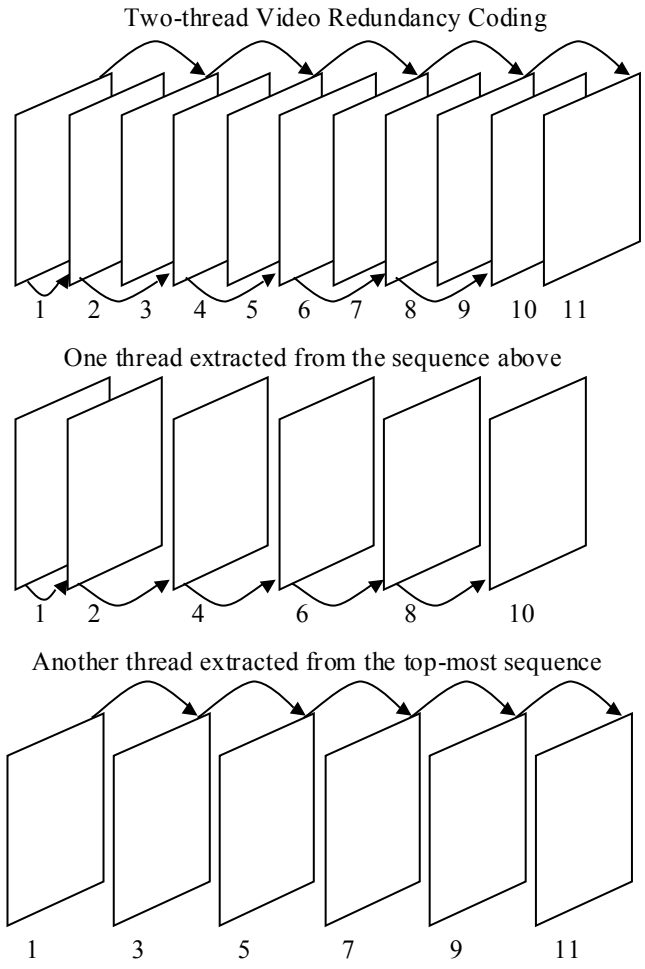


Figure 2. An example of two-thread Video Redundancy Coding.

arrows in these and all the subsequent figures indicate the direction of motion compensation and the values under the frames correspond to the relative capturing and displaying times of the frames.

3 INTRA PICTURE POSTPONEMENT

Initial buffering in players enables streaming systems to reorder video pictures in the transmitting end, transmit the pictures in the reshuffled order, and restore the original picture order in the receiving end. All reordering procedures must happen within the limits of the buffer sizes and the initial buffering time of the player. This reordering possibility allows the video coder to compress pictures in non-chronological order. To be more exact, it allows the encoder to predict a frame with reference to a frame occurring temporally after the frame to be coded. One application of such a coding technique, proposed now in this paper, is to postpone the compression of an INTRA frame. Conventionally, an INTRA frame is coded immediately after a scene cut or as a response to an expired INTRA frame refresh period, for example. In the proposed coding scheme, an INTRA frame is not coded immediately after a need to code an INTRA frame arises, but rather a

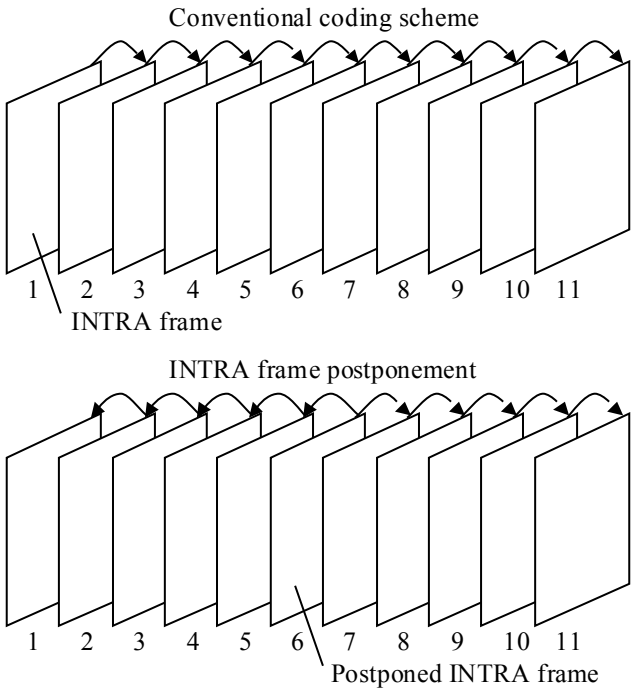


Figure 3. A comparison of conventional coding and INTRA picture postponement.

temporally subsequent picture is selected as an INTRA frame. Each frame between the coded INTRA frame and the conventional location of an INTRA frame is predicted from the next temporally subsequent picture. Figure 3 shows an example of a sequence coded according to the proposed coding scheme that is herein referred to as INTRA picture postponement.

As Figure 3 shows, the INTRA picture postponement method generates two independent INTER picture prediction chains, whereas conventional coding algorithms produce a single INTER picture chain. It is intuitively clear that the two-chain approach is more robust against erasure errors than the one-chain conventional approach. If one chain suffers from a packet loss, the other chain may still be correctly received. In conventional coding, a packet loss always causes error propagation to the rest of the INTER picture prediction chain.

The INTRA picture postponement method does not increase the temporal distance between predicted frames and their reference frames, but rather it just reverses some of the prediction directions. Thus, intuitively thinking, it should not affect compression efficiency.

The proposed method can be combined with any of the temporal scalability methods presented in section 139. The method further improves the loss resiliency of the mentioned temporal scalability methods.

4 SIMULATIONS

To demonstrate the benefits of the INTRA picture postponement method in practice, we performed a set of simulations. We followed the common conditions for the low-latency H.323/Internet communication specified by

	Nokia decoder		TMN decoder	
	Conventional coding	INTRA frame postponement	Conventional coding	INTRA frame postponement
0 % packet loss				
Number of error-free frames	750	750	750	750
Bit-rate (bits per second)	60 787	59 618	60 787	59 618
Average Y PSNR	29.35	29.56	29.35	29.56
3 % packet loss				
Number of error-free frames	265	338	265	338
Average Y PSNR	26.62	26.91	25.70	25.74
5 % packet loss				
Number of error-free frames	162	232	162	232
Average Y PSNR	24.64	25.75	24.33	25.85
10 % packet loss				
Number of error-free frames	107	184	107	184
Average Y PSNR	23.03	24.40	23.23	24.59
20 % packet loss				
Number of error-free frames	38	86	38	86
Average Y PSNR	20.38	21.59	19.32	20.44

Table 1. Summary of the simulation results.

ITU-T Advanced Video Coding Group [6] as much as it was reasonable. This section first describes the simulation conditions and then presents and analyzes the results.

As relatively frequent scene cuts are common in video streaming and as benefits of the proposed method can be more clearly seen if there are frequent scene cuts and/or periodic INTRA pictures, we chose Glasgow Tour as the input sequence. The uncompressed sequence contains 750 QCIF-sized frames captured at 12.5 Hz resulting into a sequence lasting one minute. There are 23 scenes in the sequence, and the duration of the scenes varies from 15 to about 80 pictures (from 1.2 to 6.4 seconds).

We modified an H.263 coder to support temporally backward prediction, and two sequences were generated. One sequence was conventionally coded, i.e., INTRA picture appeared at the beginning of each scene and the rest of the pictures in the scene were temporally predicted from the previous picture. INTRA picture postponement was applied to the other sequence, and each INTRA picture was located approximately in the middle of a scene. All pictures in the input sequence were coded. We used a constant quantization parameter (15) in order to minimize the impact of INTRA picture position in the PSNR results.

We encapsulated the video data according to the common simulation conditions [6], i.e., there was a slice for each macroblock line and every other slice was located in the same RTP packet resulting into two packets per picture. We used the software provided by ITU-T [7], which consists of:

- code for RTP packetization according to the H.263+ RTP payload specification RFC2429 [8],
- a packet loss simulator, and
- code for RFC2429 depacketization.

At first, sequences were packetized with the packetization tool. Then, the packet loss simulator erased some of the generated packets according to error patterns released in [9]. The resulting stream of packets was decapsulated using the depacketization tool. Finally, the generated H.263 streams were decoded with the Nokia H.263 decoder and with the TMN 3.2 decoder (by Telenor and University of British Columbia). Both decoders can conceal lost data, but their loss concealment algorithms differ from each other. We used two decoder implementations to demonstrate that the benefits of the INTRA frame postponement method are independent of differences in implementations.

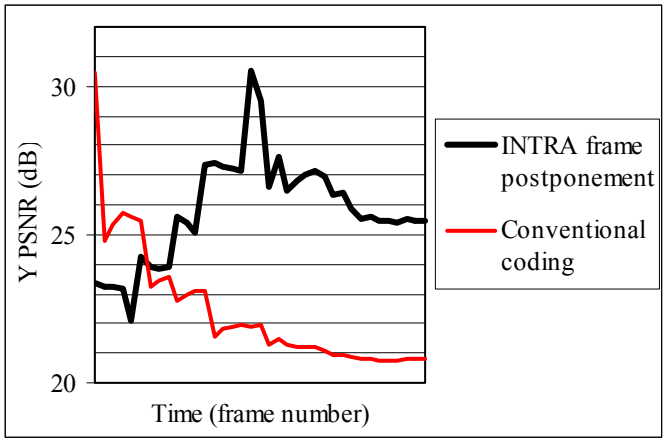


Figure 4. Luminance PSNR of one scene (10 % packet loss rate).

Table 1 summarizes the simulation results. The error-free case indicates that INTRA frame postponement improves coding efficiency somewhat probably due to more efficient coding of blended scene changes. The packet loss simulations show that the proposed method is superior both in terms of the number of error-freely reconstructed

Figure 5. Reconstructed frames of an example scene. The top row shows frames reconstructed from a sequence coded according to the INTRA frame postponement method. The bottom row is conventionally coded.

pictures and in terms of average luminance PSNR. The luminance PSNR difference between corresponding results obtained with the Nokia decoder and with the TMN decoder is caused by different error concealment algorithms.

Figure 4 shows the luminance PSNR curves for one reconstructed scene that suffered from a 10 % packet loss rate. The peaks in the PSNR curves represent INTRA frames. The curves reveal that there is a major packet loss in the first half of the scene, which degrades the quality of the remaining frames in the conventionally coded scene. An opposite behavior can be seen in the curve for the INTRA frame postponement method: the frames preceding the packet loss were corrupted, whereas an acceptable image quality was maintained throughout the rest of the reconstructed scene. In average, the PSNR for the INTRA frame postponement method is better than the PSNR for the conventional coding scheme, since there are fever seriously corrupted frames in the scenes coded according to the proposed method.

Figure 5 presents selected frames from our example scene. The frames confirm the observation made from the PSNR curves: The conventionally coded first frame of the sequence looks better than the frame coded according to the proposed method, because the conventionally coded frame has not suffered from any propagated packet losses. However, the other conventionally coded example images have noticeable artifacts, whereas the frames coded according to the proposed method are nearly error-free. Notice also that if a major packet loss had happened in the latter half of the scene, the quality of the reconstructed frames would have been approximately equal for both of the coding methods.

5 PRACTICAL IMPLEMENTATION

Current video coding standards, such as H.263 and MPEG-4, require that frames are coded in chronological

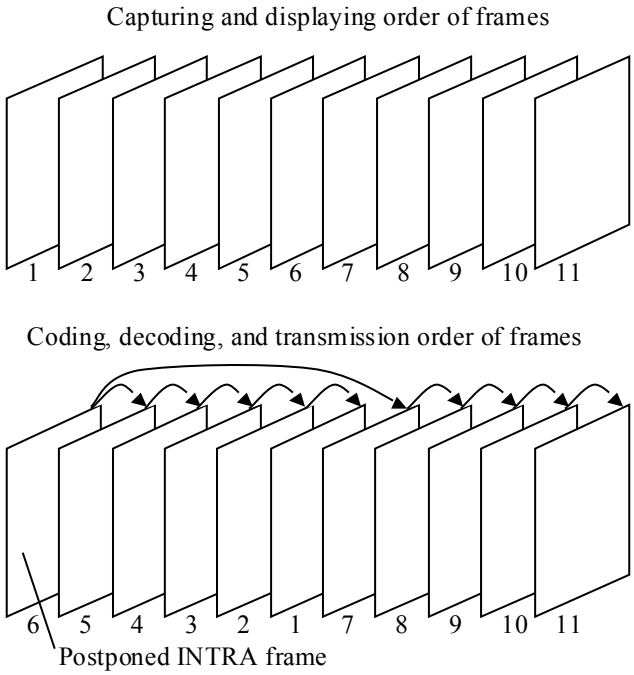


Figure 6. Illustration how to apply INTRA frame postponement in current video coding standards.

order except for B-frames. The time-stamping mechanism of the video streams allows only time increments from the previous coded frame. Moreover, the coding standards define that the transmitted time-stamp must correspond to the sampling instance of the picture.

Each RTP packet header contains a time-stamp too. RTP payload formats for H.263 [8] and MPEG-4 [10] define that the RTP packet time-stamp must correspond to the same time that is indicated in the time-stamp of the H.263 or MPEG-4 payload.

Due to the facts described above, it is not possible to implement the INTRA frame postponement method in today's standard-compliant video communication systems. However, the implementation of the proposed coding method would be possible by relaxing the rules that determine how to set time-stamps within a video bit-stream and how to associate RTP packet time-stamps with the time-stamps of the video bit-stream. Video bit-stream time-stamps would be used to indicate a correct coding and decoding order of frames, but they would not correspond to sampling or displaying times. Figure 6 demonstrates how to reorder frames for coding, decoding, and transmission, and which reference frames to use in order to code a sequence according to the example in Figure 3. RTP packet time-stamps would tell receivers how to restore the correct order of frames for displaying.

In order to allow a standard-compliant way to apply the INTRA frame postponement method, the reference picture selection tool of the future ITU-T H.26L video coding standard will enable backward prediction from pictures occurring temporally after the picture to be coded.

6 CONCLUSION

IP multicast streaming is a demanding application from packet loss recovery point of view, because interactive error control methods, such as retransmission, cannot be widely used. Furthermore, certain factors may reduce the efficiency of interactive error control techniques in point-to-point streaming. Thus, the usage of non-interactive error control methods is justified both in multicast and in point-to-point streaming. This paper introduced a simple non-interactive video coding method, called INTRA picture postponement, which improves error resilience in streaming systems. The method has two stages: At first, an INTRA picture is postponed from its conventional location. Then, the frames between the conventional location and the selected INTRA frame location are coded in a temporally backward manner. As a result, each independently coded section of a video sequence contains two independent INTER picture prediction chains and becomes more robust against transmission losses when compared to the conventional coding scheme having one INTER picture prediction chain per each section. The performed packet loss simulations showed that the proposed method is superior to the conventional coding scheme both in objective and subjective terms. Moreover, the simulations revealed that the method does not cause any reduction in compression efficiency. Finally, we came to the conclusions that the method cannot be implemented according to the current video communication standards and that the upcoming ITU-T H.26L video coding standard will correct this shortcoming.

REFERENCES

[1] S. Wenger and J. Ott, " RTCP-based Feedback for Predictive Video Coding", Internet Draft "draft-wenger-avt-rtcp-feedback-00.txt", July 2000.

[2] A. Miyazaki, H. Fukushima, T. Wiebke, R. Hakenberg, and C. Burmeister, " RTP Payload Format to Enable Multiple Selective Retransmissions", Internet Draft "draft-miyazaki-avt-rtp-selret-01.txt", July 2000.

[3] S. Fukunaga and H. Kimata, " Low delay RTCP packet format for backward messages", Internet Draft "draft-fukunaga-low-delay-rtcp-00.txt", July 2000.

[4] K. Yano, M. Podolsky, and S. McCanne, " RTP Profile for RTCP-based Retransmission Request", Internet Draft " draft-ietf-avt-rtprx-00.txt", July 2000.

[5] S. Wenger, "Video Redundancy Coding in H.263+", 1997 International Workshop on Audio-Visual Services over Packet Networks, September 1997.

[6] S. Wenger, "Common conditions for the Internet/H.323 case", ITU-T SG16 document Q15-I-61, October 1999.

[7] S. Wenger, M. Luttrell, and M. Gallant, "Packet loss simulation environment", ITU-T SG16 document Q15-I-9-R1, October 1999.

[8] IETF RFC2429, “RTP Payload Format for the 1998 Version of ITU-T Rec. H.263 Video (H.263+)”, October 1998.

[9] S. Wenger, "Error patterns for Internet video experiments", ITU-T SG16 document Q15-I-16-R1, October 1999.

[10] Y. Kikuchi, T. Nomura, S. Fukunaga, Y. Matsui, and H. Kimata, " RTP payload format for MPEG-4 Audio/Visual streams", Internet Draft draft-ietf-avt-rtp-mpeg4-es-05.txt, September 2000.

[P2] D. Tian, M. M. Hannuksela, Y.-K. Wang, and M. Gabbouj, “Error resilient video coding techniques using spare pictures,” *Proceedings of the International Packet Video Workshop*, Apr. 2003.

ERROR RESILIENT VIDEO CODING TECHNIQUES USING SPARE PICTURES

Dong Tian[1], Miska M. Hannuksela[2], Ye-Kui Wang[3] and Moncef Gabbouj[4]

[1]tian@cs.tut.fi, [2]miska.hannuksela@nokia.com, [3]ye-kui.wang@nokia.com, [4]moncef.gabbouj@tut.fi
[1,3]Tampere International Center for Signal Processing (TICSP), Tampere, Finland
[2]Nokia Mobile Software, Tampere, Finland
[4]Tampere University of Technology, Finland

ABSTRACT

In error prone environments not all the picture data can be received correctly during video transmission. Error propagation introduced by employing predictive coding may degrade the sequence quality severely. A novel method, called spare pictures, is proposed to indicate the similarity between a reference picture and other pictures. With the help of the signaled spare pictures information, receivers may avoid unnecessary picture freezing, feedback and complex error concealment, which are normally done as a response to missing picture data. Furthermore, spare pictures may also help 1) to improve the quality of some decoded pictures by replacing the reference pictures by better ones, and 2) to improve error concealment by concealing a lost block as a corresponding block in one of the spare pictures. An efficient coding method for the spare macroblock maps is also proposed. The mechanism for signaling the spare pictures was adopted to H.263 version 3 and ITU-T H.264 | MPEG-4 AVC as Supplemental Enhancement Information [1], [2].

1. INTRODUCTION

In video transmission via error prone channels, errors are likely to be found in the received picture data. When a picture is lost or corrupted so severely that the concealment result is not acceptable, the receiver typically pauses video playback and waits for the next INTRA picture to restart decoding and playback. If possible, the receiver also requests the transmitter for an INTRA picture update. In some applications, e.g., in multicast video streaming, the transmitter cannot react to INTRA update requests, but rather the transmitter encodes an INTRA picture relatively frequently, such as every few seconds, to enable new clients to join the multicast session and to enable recovery from transmission errors. Consequently, receivers may have to pause video playback for a relatively long time after a lost picture, and users typically find this behavior annoying.

There are numerous ways to decrease the probability of such transmission errors that would force the decoder to pause playback. Multiple description coding will produce two or more correlated bitstreams so that a high-quality reconstruction can be obtained from all the bitstreams together, while a lower, but still acceptable, quality reconstruction is guaranteed if only one bitstream is received [3]. Video redundancy coding (VRC) generates several independent bitstreams by using independent prediction loops [4]. For example, an even frame is predicted from the previous even frame, and an odd frame from the previous odd frame. Reference picture selection (RPS) [5], etc. can be utilized if a feedback channel is available. Systematic description of error resilience video coding can be found in [6].

In this paper, we propose an alternative method, spare pictures, to mitigate the effect of error propagation introduced by employing predictive coding. Section 2 presents the idea of spare pictures and how to signal the spare pictures information. Section 3 discusses more details in the processing of the codec. Section 4 gives the simulation results and section 5 concludes the paper.

2. SPARE PICTURES

2.1. The principle of spare pictures

The idea behind spare pictures is based on the fact that sometimes another picture (or a part of another picture) resembles the actual motion compensation reference picture (or part of the picture) so well that it could be used as a reference instead of the actual one when the actual one is lost. The actual reference picture is called target picture herein. In other words, the quality degradation caused by the alternative reference picture is so small that the quality is still considered acceptable. Thus, it would be beneficial to signal which coded pictures are similar entirely or partly.

For example, we coded Hall at 15 pictures per second with constant quantization parameter of 10 with JM 2.0 [7]. Figure 1 illustrates the beginning of the coded example sequence. Figure 3 shows the second P frame (P4) of the sequence, and the image on the left is error-free. Then, we discarded the bitstream of the first P frame of the sequence (P2) and decoded the sequence. In other words, the image on the right was decoded using the frame preceding the deleted frame as a reference as illustrated in Figure 2. Practically, you cannot see the

Figure 1. *Example of coding pattern: P2 is predicted from I0 and P4 is predicted from P2*

Figure 2. *Usage of a spare reference picture: Use I0 instead of P2 to recover P4 when P2 is lost*

Decoded P4 using the original reference Decoded P4 using the spare picture

Figure 3. *Example of spare reference pictures*

Spare macroblock map Frame 74 in Hall Frame 75 in Hall

Figure 4. *Example of spare pictures map*

difference, and therefore the encoder could have signaled that I0 can be used as a spare picture for P2.

Figure 4 shows a more complicated example where frame 74 in Hall can be used as a spare picture of frame 75 except for a scattered area. In this case, a bi-level map is used to represent the spare macroblock information. In Figure 4, the picture on the left is the spare macroblock map between frames 74 and 75. The similar areas which can be used as spare macroblocks are shown in black. The spare macroblock map indicates to the decoder which parts of frame 74 can be used to replace corresponding parts of frame 75, in case the latter is a reference frame that is lost or corrupted.

2.2. Signaling of spare pictures

We propose signaling of entire and partial spare pictures with the Supplemental Enhancement Information (SEI) mechanism in JVT. If the actual reference block is lost or seriously damaged, decoders may use an entire picture or partial spare pictures instead.

Typically the spare picture SEI message contains the resemblance between a certain target picture and couple of other pictures. When delivered over packet network, the spare picture SEI message shall not be packetized together with the target picture in order to avoid the spare picture SEI message to be lost together with the target picture. In other words, spare picture SEI messages should be conveyed independently or together with the slices or data partitions belonging to pictures other than the target picture.

The indication of entire picture being spare picture is straightforward. If only parts of the picture resemble the target picture, the bi-level map of such areas need be signaled. To code the spare macroblock maps, note the fact that the neighboring spare macroblock maps resemble each other. Therefore we encode the first spare macroblock map directly, which is between the target picture and the first spare picture. For the other spare macroblock maps, the current spare macroblock map is first processed by applying an exclusive or operation on the previous spare macroblock map. Last, the macroblock

map will be arranged into 1-demensional signal so that the number of consecutive spare macroblocks can be encoded using exponential-Golomb codes. Thus the coded macroblock map need be scanned in a certain scan order. The counter-clockwise box-out proves to be an efficient scan order in our simulations.

3. PROCESSING IN THE CODEC

To determine whether a candidate reconstructed macroblock can be a spare macroblock of the current reconstructed macroblock in a target picture during encoding, the average pixel difference in luminance is used as the criterion to evaluate the similarity between the target macroblock and the candidate macroblock. If the average pixel difference is less than a certain threshold, it will infer that the candidate macroblock can be a spare macroblock. The empirical average pixel difference threshold is 6.

If the number of the macroblocks in a candidate picture that can be used as spare macroblock is large enough, the encoder will assume that the entire picture can serve as a spare picture to reduce the overhead in signaling spare picture information. The empirical percentage threshold is selected as 92%.

During decoding, if the spare pictures information is received for a picture that is a reference picture and is partly or entirely lost during transmission, the lost parts of the picture should be replaced with correctly reconstructed spare reference macroblocks or picture, if available.

4. SIMULATIONS

This section presents the simulations of using spare pictures information to improve quality of the displayed sequences in error prone environments. We also present the performance of encoding spare macroblock maps using different methods in terms of coding efficiency.

4.1. Evaluation method

Delivery of video over Internet is likely to be facilitated by the real-time transport protocol (RTP). Some RTP packets may be lost during transmission. The lossy bitstream is fed to the decoder implemented based on JM 2.0 [7]. Three types of output sequences (displayed pictures) will be created according to different decoders.

Decoder 1: Reconstruct the sequence normally without freezing or spare pictures. This case corresponds to a “careless” decoder that displays any decoded picture regardless whether the picture is corrupted or not.

Decoder 2: Reconstruct the sequence normally with freezing only. If a picture is correctly reconstructed, output it normally. Otherwise, repeat the latest displayed picture until the next INTRA picture. This case corresponds to a “cautious” decoder that freezes the output of pictures if there are any errors in the received data stream.

Decoder 3: Reconstruct the sequence with help of spare pictures. If a picture is correctly reconstructed, output it normally. If a reference picture of the current picture is missing but the reference picture has a correctly reconstructed spare picture, decode and output the current picture using the alternative reference picture. Otherwise, repeat the latest displayed picture, that is, freeze the output of reconstructed pictures.

To assess these different decoders, we shall compute the following quantities:

- the number of displayed pictures (NumDispPics),
- the number of correctly received pictures (NumCorRecPics), and
- the number of purposely repeated pictures (NumPurRepPics) for each decoder, and
- the number of rescued pictures using spare pictures (NumResPics) for the proposed technique.

NumResPics indicates how many pictures would have been frozen in decoder 2, which were rescued using spare pictures in the proposed decoder.

4.2. Simulation method and conditions

The simulations are performed based on JM 2.0, and the detailed simulation conditions are as shown below, which follow closely the common conditions in [8].

Sequences, Intra Coding Period and Error Concealment

Simulations are performed using three QCIF sequences Hall at 64 kbps and 15 fps, Akiyo, and News at 144 kbps and 15 fps, with periodical INTRA coded frames inserted to stop error propagation. The INTRA refresh period is one second. Except for the first frame, if an INTRA coded frame contains losses, INTER (spatial-temporal) error concealment is used instead of INTRA (spatial) error concealment. It is considered that the shot change signaling proposed in [9] is used to choose an appropriate error concealment method for INTRA pictures.

Bitrate Calculation

As stated in the common conditions specified in [8], coding parameters such as quantization parameter are chosen to make the resulting bitrate as close as possible to but not larger than the channel bitrate, taking into account the 40 bytes of IP/UDP/RTP headers per packet.

Packet Loss Simulation

We assume that the packet containing the parameter set is conveyed reliably (possibly out-of-band during the session setup). At least one packet of the first frame should be

Figure 5. *Snapshots of Hall @ packet loss rate 5%, 16 frames before frame 90 are frozen by Decoder 2 and rescued by Decoder 3*

received to avoid decoder crash. To meet the requirement, the first packet of the first frame is assumed to be always received regardless of the corresponding error pattern. The simulations are run with packet loss rates being selected as 3% and 5%.

4.3. Simulation results

The simulation results are shown in Table 1 - 3. As can be seen from the results, spare pictures information can

1) prevent displaying some corrupted frames that would be displayed by a "careless" decoder 1, and
2) help the decoder to rescue some pictures with acceptable quality that would have been frozen by a "cautious" decoder 2.

Both "careless" decoder 1 and "cautious" decoder 2 will cause annoying effects: unacceptable picture quality or slow frame rate (see Figure 5: Snapshots of Hall), respectively. Furthermore, from the results of Akiyo, we can see that no frames were frozen in decoder 3. By observing the output sequences, no rescued frames can be identified by human eyes. In sequences which have more motion, such as in News, fewer frames can be rescued.

4.4. Simulations for encoding spare macroblock maps

We encode two spare macroblock maps for all the frames. Direct encoding and differential encoding are compared under different scan modes:

1) Direct encode using raster/counter-clockwise box-out scan order, namely DirRaster/DirBox, that is, no exclusive-or operation between spare macroblock maps is applied, and,
2) Differential encode using raster/counter-clockwise box-out scan order, namely DiffRaster/DiffBox, that is, when encoding the second spare macroblock map, an exclusive or operation is applied first as explained in section 2.2.

The ratio of the average data amount of the encoded maps to that of the original bi-level maps (one bit per macroblock), is given in terms of percentages for each sequence in Table 4.

As can be seen from the results, the most efficient way is the differential encoding method combined with counter-clockwise box-out scan order.

5. CONCLUSIONS

In this paper, we introduced an alternative method, spare pictures, to improve the quality of the received sequences, which can help the receiver to recover pictures referring to a lost picture, and prevent unnecessary picture freezing, feedback and complex error concealment. An efficient coding method for the spare macroblock map is also proposed. Simulation results show that the solution is efficient in combating packet losses in video transmission over Internet. The mechanism for signaling the spare pictures was adopted to H.263 version 3 and ITU-T H.264 | MPEG-4 AVC as Supplemental Enhancement Information.

6. REFERENCES

[1] Thomas Wiegand, "Editor's Proposed Draft Text Modifications for Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC), Geneva modifications draft 37", document JVT-E146d37, JVT of ISO/IEC MPEG & ITU-T VCEG, October 2002

[2] Dong Tian, Ye-Kui Wang and Miska M. Hannuksela, "Spare Pictures", document JVT-D100, JVT of ISO/IEC MPEG & ITU-T VCEG, July 2002

[3] Vivek K Goyal, "Multiple Description Coding: Compression Meets the Network", IEEE Signal Processing Magazine, Volume: 18 Issue: 5 , Sept. 2001

[4] Stephan Wenger, "Video Redundancy Coding in H.263+", Workshop on Audio-Visual Services for packet networks (aka Packet Video Workshop), 1997

[5] Yi J. Liang, Markus Flierl and Bernd Girod, "Low-Latency Video Transmission over Lossy Packet Networks Using Rate-Distortion Optimized Reference Picture Selection", IEEE ICIP 2002

[6] Yao Wang, Stephan Wenger, Jiangtao Wen and Aggelos K. Katsaggelos, "Error Resilient Video Coding Techniques", IEEE Signal Processing Magazine, July 2000

[7] JVT, "JVT reference software 2.0", available at ftp://ftp.imtc-files.org/jvt-experts/reference_software

[8] Stephan Wenger, "Common Conditions for wire-line, low delay IP/UDP/RTP packet loss resilient testing", document VCEG-N79r1, ITU-T Q.6/SG16 VCEG

[9] Ye-Kui Wang and Miska M. Hannuksela, "Signaling of Shot Changes", document JVT-D099, JVT of ISO/IEC MPEG & ITU-T VCEG, July 2002

Table 1. *Hall@64kbps + 15fps, result bitrate=62.6kbps, QP=16, totally 150 coded frames*

3% packet loss rate, NumDispPics = 150, NumCorRecPics = 97

Number of Frames	Decoder 1	Decoder 2	Decoder 3
NumPurRepPics	0	53	0
NumResPics	NA	NA	53

5% packet loss rate, NumDispPics = 150, NumCorRecPics = 77

Number of Frames	Decoder 1	Decoder 2	Decoder 3
NumPurRepPics	0	73	24
NumResPics	NA	NA	49

Table 2. *Akiyo@144kbps + 15fps, result bitrate=140.31kbps, QP=5, totally 150 coded frames*

3% packet loss rate, NumDispPics = 150, NumCorRecPics = 98

Number of Frames	Decoder 1	Decoder 2	Decoder 3
NumPurRepPics	0	52	0
NumResPics	NA	NA	52

5% packet loss rate, NumDispPics = 150, NumCorRecPics = 33

Number of Frames	Decoder 1	Decoder 2	Decoder 3
NumPurRepPics	0	117	56
NumResPics	NA	NA	61

Table 3. *News@144kbps + 15fps, result bitrate=132.66kbps, QP=12, totally 150 coded frames*

3% packet loss rate, NumDispPics = 150, NumCorRecPics = 103

Number of Frames	Decoder 1	Decoder 2	Decoder 3
NumPurRepPics	0	47	34
NumResPics	NA	NA	13

5% packet loss rate, NumDispPics = 150, NumCorRecPics = 51

Number of Frames	Decoder 1	Decoder 2	Decoder 3
NumPurRepPics	0	99	99
NumResPics	NA	NA	0

Table 4. *Comparison between different methods for encoding spare macroblock maps*

DirRaster	DirBox	DiffRaster	DiffBox
Foreman@144kbps + 15fps, result bitrate = 142.96kbps, QP = 15, totally 200 coded frames			
101 %	98 %	84 %	82 %
Akiyo@144kbps + 15fps, result bitrate = 140.31kbps, QP = 5, totally 150 coded frames			
31 %	24 %	27 %	22 %
News@144kbps + 15fps, result bitrate = 132.66kbps, QP = 12, totally 150 coded frames			
52 %	43 %	43 %	38 %

[P3] T. Stockhammer, M. M. Hannuksela, and T. Wiegand, “H.264/AVC in wireless environments,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 657-673, Jul. 2003.

H.264/AVC in Wireless Environments

Thomas Stockhammer, Miska M. Hannuksela, and Thomas Wiegand

***Abstract*—Video transmission in wireless environments is a challenging task calling for high-compression efficiency as well as a network friendly design. Both have been major goals of the H.264/AVC standardization effort addressing "conversational" (i.e., video telephony) and "nonconversational" (i.e., storage, broadcast, or streaming) applications. The video compression performance of the H.264/AVC video coding layer typically provides a significant improvement. The network-friendly design goal of H.264/AVC is addressed via the network abstraction layer that has been developed to transport the coded video data over any existing and future networks including wireless systems. The main objective of this paper is to provide an overview over the tools which are likely to be used in wireless environments and discusses the most challenging application, wireless conversational services in greater detail. Appropriate justifications for the application of different tools based on experimental results are presented.**

***Index Terms*—Error concealment, error-resilient video coding, H.264/AVC, multiple reference frames, rate-distortion optimization, video coding standards, wireless video transmission.**

I. Introduction

SINCE 1997, the ITU-T's Video Coding Experts Group (VCEG) has been working on a new video coding standard with the internal denomination H.26L. In late 2001, the Moving Picture Expert Group (MPEG) and VCEG decided to work together as a Joint Video Team (JVT), and to create a single technical design called H.264/AVC for a forthcoming ITU-T Recommendation H.264/AVC and for a new part of the MPEG-4 standard called AVC [1][1] , [2]. Since the meeting in November 2002, the technical specification is frozen and the standard text and software have been finalized. The primary goals of H.264/AVC are *improved coding efficiency* and *improved network adaptation*. The syntax of H.264/AVC typically permits a significant reduction in bit rate [3] compared to all previous standards such as ITU-T Rec. H.263 [4] and ISO/IEC JTC 1 MPEG-4 [5] at the same quality level.

The demand for fast and location-independent access to multimedia services offered on today's Internet is steadily increasing. Hence, most current and future cellular networks, like GSM-GPRS, UMTS, or CDMA-2000, contain a variety of packet-oriented transmission modes allowing transport of practically any type of IP-based traffic to and from mobile terminals, thus providing users with a simple and flexible transport interface. The third generation partnership project (3GPP) has selected several multimedia codecs for the inclusion into its multimedia specifications [6]. To provide basic video service in the first release of the 3G wireless systems, the well-established and almost identical baseline H.263 and the MPEG-4 visual simple profile have been integrated. The choice was based on the manageable complexity of the encoding and decoding process, as well as on the maturity and simplicity of the design.

However, due to the likely business models in emerging wireless systems in which the end-user's costs are proportional to the transmitted data volume and also due to limited resources bandwidth and transmission power, compression efficiency is the main target for wireless video and multimedia applications. This makes H.264/AVC coding an attractive candidate for all wireless applications including multimedia messaging services (MMS), packet-switched streaming services (PSS), and conversational applications. However, to allow transmission in different environments, not only is coding efficiency relevant, but also seamless and easy integration of the coded video into all current and possible future protocol and multiplex architectures. In addition, for conversational applications the video codec's support of enhanced error-resilience features is of major importance. This has also been taken into account in the standardization of this codec.

This paper is structured as follows. Section II introduces applications and transmission characteristics for wireless video applications. The transport of H.264/AVC video is briefly discussed and common test conditions for mobile video transmission are presented. Section III provides an overview over the H.264/AVC video coding standard from the perspective of wireless video applications. We categorize features according to their applicability in different video services. Section IV discusses the most challenging application in terms of delay constraints and error resilience, namely wireless conversational applications. A system description and problem formulation is followed by providing several alternatives on the system design using H.264/AVC as well as the combination of several modes. Section V provides experimental results for selected system concepts based on the common test conditions.

II. Video in Mobile Networks

A. Overview: Applications and Constraints

Video transmission for mobile terminals is likely to be a major application in emerging 3G systems and may be a key factor in their success. The video-capable display on mobile devices paves the road to several new applications. Three

Manuscript received April 9, 2002; revised May 10, 2003.

T. Stockhammer is with the Institute for Communications Engineering (LNT), Munich University of Technology (TUM), 80290 Munich, Germany (e-mail: stockhammer@ei.tum.de).

M. M. Hannuksela is with the Nokia Corporation, 33721 Tampere, Finland (e-mail: miska.hannuksela@nokia.com).

T. Wiegand is with the Fraunhofer-Institute for Telecommunications—Heinrich-Hertz-Institute Einsteinufer 37, 10587 Berlin, Germany (e-mail: wiegand@hhi.de).

Digital Object Identifier 10.1109/TCSVT.2003.815167

[1]All referenced standard documents can be accessed via anonymous ftp at ftp://standard.pictel.com/video_site, ftp://ftp.imtc-files.org/jvt-experts, ftp://ftp.ietf.org/, or ftp://www.3gpp.org/Specs/archive.

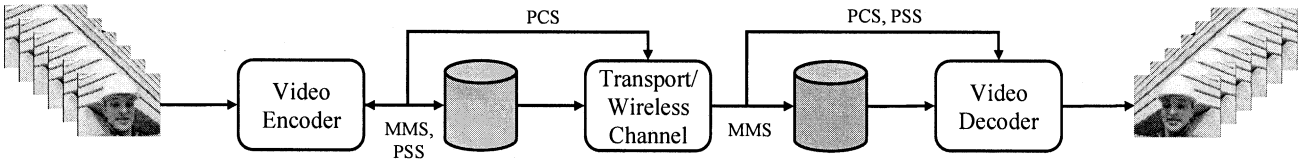


Fig. 1. Wireless video application MMS, PSS, and PCS: differentiation by real-time or offline processing for encoding, transmission, and decoding.

major service categories were identified in the H.264/AVC standardization process [7].

1) Circuit-switched [8] and packet-switched conversational services (PCS) [9] for video telephony and conferencing.
2) Live or pre-recorded video packet-switched streaming services (PSS) [10].
3) Video in multimedia messaging services (MMS) [11].

Although new services such as multimedia broadcast/multicast services (MBMS) [12] are planned for future releases of wireless networks, we restrict ourselves to single receiver applications. Mobile devices are hand-held and constrained in processing power and storage capacity. Therefore, a mobile video codec design must minimize terminal complexity while remaining consistent with the efficiency and robustness goals of the design. As complexity issues are discussed elsewhere in this issue [13], [14], we restrict ourselves to transmission constraints and properties.

The transmission requirements for the three identified applications can be distinguished with respect to requested data rate, the maximum allowed end-to-end delay, and the maximum delay jitter. This results in different system architectures for each of these applications. A simplified illustration is provided in Fig. 1. As MMS does not include any real-time constraints, encoding, transport, and decoding are completely separated. The recorded video signal is offline encoded and locally stored. The transmission is started using the stored signal at any time. The decoding process at the receiver is in general not started until the completion of the download. In PSS applications, the user typically requests pre-coded sequences, which are stored at a server. Whereas encoding and transmission are separated, decoding and display is started during transmission to minimize the initial delay and memory usage in mobile devices. Finally, in conversational services, the end-to-end delay has to be minimized to avoid any perceptual disturbances and to maintain synchronicity of audio and video. Therefore, encoding, transmission and decoding is performed simultaneously in real-time and, moreover, in both directions. These different ancillary conditions permit and require different strategies in encoding, transport, decoding, as well as in the underlying network and control architecture.

In general, the available bandwidth and therefore the bit-rate over the radio link are limited and the costs for a user are expected to be proportional to the reserved bit rate or the number of transmitted bits over the radio link. Thus, low bit rates are likely to be typical, and *compression efficiency* is the main requirement for a video coding standard to be successful in a mobile environment. This makes H.264/AVC a prime candidate for the use in wireless systems, because of its superior compression efficiency [3].

In addition, the mobile environment is characterized by harsh transmission conditions in terms of attenuation, shadowing, fading, and multi-user interference, which result in time- and location-varying channel conditions. The frequency of the channel variations highly depends on the environment, the user topology, the velocity of the mobile user, and the carrier frequency of the signal. For sufficiently long code words averaging over channel statistics is possible and transmission strategies can be used that are based on the long-term averages of fading states and the ergodic behavior of the channel. Many highly sophisticated radio link features such as broadband access, diversity techniques, space-time coding, multiple antenna systems, fast power control, interleaving, and forward error correction (FEC) by Turbo codes are used in 3G systems to reduce variations in channel conditions. However, only for fast-moving users and relatively large tolerated maximum delay can these advanced techniques provide a negligible bit error and radio block loss rate. Usually, some amount of residual errors has to be tolerated for low-delay applications due to the nonergodic behavior of the channel and the imperfectness of the applied signal processing. Therefore, in addition to high compression efficiency and reasonable complexity, a video coding standard to be applicable for conversational services in wireless environments has to be error resilient.

In addition, it is worth noting at this point that new directions in the design of wireless systems do not necessarily attempt to minimize the error rates in the system, but to maximize the throughput. This is especially appealing for services with relaxed delay constraints such as PSS and MMS. The nonergodic behavior of the channel is exploited such that in case of good channel states significantly higher data rate is supported than in bad channel states. In addition, reliable link layer protocols with persistent Automatic Repeat reQuest (ARQ) are usually used to guarantee error-free delivery. For example, in the high-speed downlink packet access (HSDPA) concept [15] ARQ, adaptive modulation schemes, and multiuser scheduling taking into account the channel states are combined to significantly enhance the throughput in wireless systems.

3G wireless transmission stacks usually consist of two different bearer types, dedicated and shared channels. Whereas in dedicated channels one user gets assigned a fixed data rate for the entire transmission interval, shared channels allow a dynamic bit-rate allocation similar to ATM or GSM GPRS. HSDPA will be an extension of the shared channel concept on the air interface. Except for MMS all streaming and conversational applications are assumed to use dedicated channels in the initial phase of 3G wireless systems due to their almost constant bit-rate behavior. In modern system designs, an application can request one of many different quality-of-service (QoS) classes. QoS classes contain parameters like a maximum error rate,

TABLE I
QOS SERVICE CLASSES IN PACKET RADIO SYSTEMS [15]

Traffic Class	Fundamental Characteristics	Typical Examples
Conversational	Preserve time relation between information entities of the stream Conversational pattern (stringent and low delay)	Voice and video telephony, Video conferencing
Streaming	Preserve time relation (variation) between information entities of the stream	Streaming multimedia (video, audio, etc.)
Interactive	Request response pattern, Preserve data integrity	Web browsing, network games
Background	Destination is not expecting the data within a certain time Preserve data integrity	Background download of e-mails, files, etc.

maximum delay, and a guaranteed maximum data rate. Furthermore, according to [16], applications are usually divided into different service classes: conversational, streaming, interactive, and background traffic. Characteristics and typical examples are shown in Table I.

B. Transport of H.264/AVC Video in Wireless Systems

According to Fig. 2, H.264/AVC distinguishes between two different conceptual layers, the video coding layer (VCL) and the network abstraction layer (NAL). Both the VCL and the NAL are part of the H.264/AVC standard. The VCL specifies an efficient representation for the coded video signal. The NAL of H.264/AVC defines the interface between the video codec itself and the outside world. It operates on NAL units which give support for the packet-based approach of most existing networks. At the NAL decoder interface, it is assumed that the NAL units are delivered in decoding order and that packets are either received correctly, are lost, or an error flag in the NAL unit header can be raised if the payload contains bit errors. The latter feature is not part of the standard as the flag can be used for different purposes. However, it provides a way to signal an error indication through the entire network. Additionally, interface specifications are required for different transport protocols that will be specified by the responsible standardization bodies. The exact transport and encapsulation of NAL units for different transport systems, such as H.320 [17], MPEG-2 Systems [18], and RTP/IP [19], are also outside the scope of the H.264/AVC standardization. The NAL decoder interface is normatively defined in the standard, whereas the interface between the VCL and the NAL is conceptual and helps in describing and separating the tasks of the VCL and the NAL.

For real-time video services over 3G mobile networks, two protocol stacks are of major interest. 3GPP has specified a multimedia telephony service for circuit-switched channels [8] based on ITU-T Recommendation H.324M. For IP-based packet-switched communication, 3GPP has chosen to use SIP and SDP for call control [20] and RTP for media transport [9]. In other words, the IP-based protocol stack as presented in [21] will be used in packet-switched 3G mobile services. While the H.324 and the RTP/UDP/IP stacks have different roots and a completely different switching philosophy, the loss and delay effects on the media data when transmitting over wireless dedicated channels are very similar.

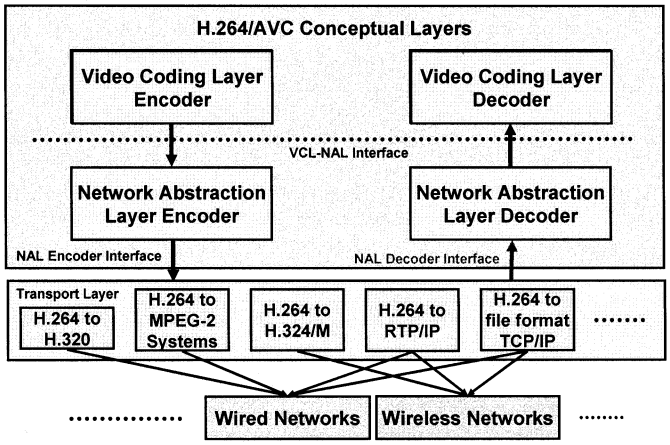


Fig. 2. H.264/AVC standard in transport environment.

H.324 [22] was primarily established by the ITU-T for low-bit-rate circuit-switched modem links and error prone extensions for mobile circuit switched low-bit-rate conversational services have been added. H.324M officially known as ITU-T Rec. H.324 Annex C, allows transmission over low, moderate, and high bit-error rate circuit switched links. 3GPP adopted H.324M including an error robust extension of the multiplexing protocol H.223 known as H.223 Annex B as the protocol used for circuit-switched video communication. This multiplexing protocol includes two layers: an error-resilient packet-based multiplex layer and an adaptation layer featuring common error detection capabilities, such as sequence numbering and cyclic redundancy checks (CRSs). Therefore, it is very similar to the RTP/UDP/IP stack (see [21]).

For packet-switched services, 3GPP/3GPP2 agreed on an IP-based stack. Fig. 3 shows a typical packetization of a NAL unit encapsulated in RTP/UDP/IP [19] through the 3GPP2 user plane protocol stack. After robust header compression (RoHC) [23] this IP/UDP/RTP packet is encapsulated into one packet data convergence protocol/point-to-point protocol (PDCP/PPP) packet that becomes a radio link control (RLC)-service data unit (SDU). The RLC protocol can operate in three modes: 1) transparent; 2) unacknowledged; and 3) acknowledged mode [26]. The RLC protocol provides segmentation and retransmission services for both users and control data. The transparent and unacknowledged mode RLC entities are defined to be

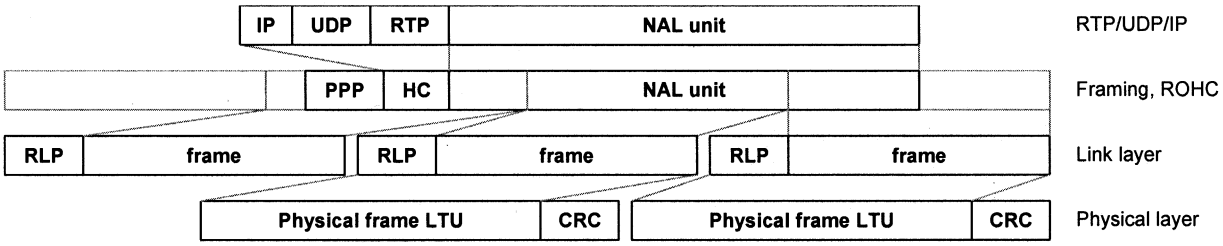


Fig. 3. Packetization through 3GPP2 user plane protocol stack.

TABLE II
BIT-ERROR PATTERNS

No.	Bit rate	Length	BER	Mobile Speed	Application
1	64 kbit/s	60 s	9.3e-3	3 km/h	Streaming
2	64 kbit/s	60 s	2.9e-3	3 km/h	Streaming
3	64 kbit/s	180 s	5.1e-4	3 km/h	Conversational
4	64 kbit/s	180 s	1.7e-4	50km/h	Conversational
5	128 kbit/s	180 s	5.0e-4	3 km/h	Conversational
6	128 kbit/s	180 s	2.0e-4	50km/h	Conversational

unidirectional and acknowledged mode entities are described as bi-directional. For all RLC modes, CRC error detection is performed on the physical layer and the result of the CRC check is delivered to the RLC together with the actual data. In the transparent mode no protocol overhead is added to higher layer data. Erroneous protocol data units (PDUs) can be discarded or marked erroneous. In the unacknowledged mode, no retransmission protocol is in use and data delivery is not guaranteed. Received erroneous data is either marked or discarded depending on the configuration. In the acknowledged mode, an automatic repeat request mechanism is used for backward error correction.

As video packets are of varying length by nature, the length of RLC-SDU's varies as well. If an RLC-SDU is larger than an RLC-PDU, the SDU is segmented into several PDUs. In the used, unacknowledged, and acknowledged modes, the flow of variable-size RLC-SDUs is continuous to avoid padding of bits as necessary for the transparent mode. In unacknowledged mode, if any of the RLC-PDUs containing data from a certain RLC-SDU have not been received correctly, the RLC-SDU is typically discarded. In acknowledged mode, the RLC/radio link protocol (RLP) layer can perform re-transmissions.

Additionally, both protocol stacks H.324 and RTP/IP/UDP use reliable setup and control protocols, H.245 and SIP, respectively. Hence, it can be assumed that a small amount of control information can be transported out-of-band in a reliable way. The resulting properties of real-time low-delay video transmission are therefore very similar in both cases. Packets are transmitted over underlying transports protocols and channels, which provide framing, encapsulation, error detection, and reliable setup. We focus hereafter on the RTP/IP-based transmission over wireless channels [21].

C. Common Test Conditions for Wireless Video

In the H.264/AVC standardization process, the importance of mobile video transmission has been recognized by adopting appropriate common test conditions for 3G mobile transmission for circuit switched conversational services based on H.324M [24] and for packet-switched conversational and streaming services [25]. These test conditions permit the selection of appropriate coding features, testing and evaluating error-resilience features, as well as meaningful anchor results. In this paper, we focus on the IP-based test conditions. The common test conditions define six test-case combinations for packet-switched conversational services as well as packet-switched streaming services over 3G mobile networks. Additionally, the test conditions include simplified offline 3GPP/3GPP2 simulation software, programming interfaces and evaluation criteria. Radio channel conditions are simulated with bit-error patterns, which were generated from simulated mobile radio channel conditions. The bit-error patterns are captured above the physical layer and below the RLC/RLP layer and, therefore, they are used as the physical layer simulation in practice. The properties bit rate, length, bit-error rate, and the mobile speed of the bit-error patterns are presented in Table II.

The bit errors in the files are statistically dependent, as channel coding and decoding included in 3G systems produces burst errors. This has been taken into account by evaluating the bit-error pattern files in the following. Patterns 1 and 2 are mostly suited to be used in video streaming applications, where RLP/RLC layer re-transmissions can correct many of the frame losses. The applied channel-coding scheme is a Turbo code scheme and power control targeting throughput maximization rather than error minimization. Patterns 1 and 2 are unrealistic for conversational service, as an acceptable quality cannot be achieved with such high error rates without retransmissions.

Patterns 3–6 are meant to simulate a more reliable, lower error-rate bearer that is required in conversational applications. Assuming a random byte starting position within the file the packet error probability $p_e(r)$ depending of the length of the packet r in bytes can be determined. These error probabilities $p_e(r)$ for all bit-error patterns are shown in Fig. 4. It is obvious

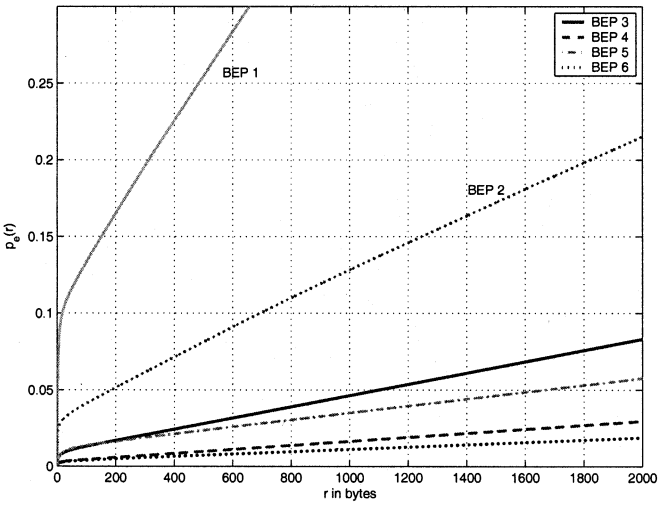


Fig. 4. Packet-loss probability $p_e(r)$ over packet length r in bytes for bit-error patterns according to [25].

that the error rate increases significantly with increasing packet length. For bit-error patterns (BEP1 and 2), the loss rates are not acceptable as already short packets, e.g., 500 bytes, have loss probabilities up to 25%. The high error rates require retransmission on the link layer.

The patterns useful for conversational services provide very decent error characteristics for reasonable packet lengths. The loss probability for reasonable packet sizes up to 1000 bytes is below 5%. This means that typical fixed Internet packet loss probabilities (compare [21]) are not exceeded. Note that with higher speed (50 km/h), the channel tends to be "more ergodic" than in case of the walking user (3 km/h). Therefore, the error rates are usually higher for slowly moving users than for fast-moving users.

During the standardization process, it was agreed in the very beginning that the standard should include error-resilience features for IP-based wired and wireless transmission. Usually two kinds of errors are present in today's transmission systems: bit inversion errors or packet losses. However, all relevant multiplexing protocols like H.223 and UDP/IP and almost all underlying mobile systems include packet loss and bit-error detection capabilities applying sequence numbering and block check sequences, respectively. Therefore, it can be assumed that a vast majority of erroneous transmission packets can be detected. Moreover, even if packets were detected to contain bit errors, decoding could be attempted. Some research has been conducted in this area and, in a few scenarios, gains have been reported for still image transmission [27]. However, in the standardization of H.264/AVC for the development of the reference software, bit-erroneous packets have been considered as being discarded by the receiver for the following reasons.

1) Processing of bit-erroneous packets is likely to be possible in a receiving mobile device only, as gateways and receivers having fixed network connection typically drop erroneous packets.
2) Joint source-channel decoding, such as trellis-based decoding of variable-length codes or iterative source and channel decoding might be applied. However, these techniques have not yet shown significant improvements for video decoding.
3) The decoding based on lost packets serves as a lower bound on the performance in bit-error-prone environments and, therefore, provides a valid benchmark of the performance of H.264/AVC in error-prone environment.
4) Finally, handling of bit errors generally complicates the implementation of decoder software significantly. As the test model software for H.264 was developed for multiple purposes, only simple but meaningful network interfaces have been integrated.

III. H.264/AVC—An Efficient and Flexible Video Coding Toolbox

A. Compression Efficiency and Encoder Flexibility

The features for compression efficiency are discussed elsewhere in this issue, we will only briefly present the key features of the standard, for more details we refer to [2]. Although the design of the H.264/AVC codec basically follows the hybrid design (motion compensation with lossy coding of residual signal) of prior video coding standards such as MPEG-2, H.263, and MPEG-4, it contains many new features that enable it to achieve a significant improvement in terms of compression efficiency. This is the main reason why H.264/AVC will be very attractive for use in wireless environments with the costly resource bit rate. The main features for significantly increased coding efficiency are multiframe motion-compensated prediction, adaptive block size for motion compensation, generalized B-pictures concepts, quartel-pel motion accuracy, intra coding utilizing prediction in the spatial domain, in-loop deblocking filters, and efficient entropy-coding methods.

The normative part of a video coding standard in general only consists of the appropriate definition of the order and semantics of the syntax elements and the decoding of error-free bit streams. This allows a significant flexibility at the encoder, which can, on the one hand, be exploited for pure compression efficiency, and on the other hand, several included features in the standard can be selected by the encoder for other purposes such as error resilience, random access, etc. A typical encoder with the main encoding options is shown in Fig. 5.

The encoding options relevant for wireless transmission are highlighted. The recorded video data is preprocessed by appropriate spatial and temporal preprocessing such that the data rates and displays in a wireless environment are well-matched. For the quantization of transform coefficients, H.264 coding uses scalar quantization. The quantizers are arranged in a way that there is an increase of approximately 12.5% from one quantization parameter (QP) to the next. The quantized transform coefficients are converted into coding symbols and all syntax elements of a macroblock (MB) including the coding symbols are conveyed by entropy coding methods. A MB can always be coded in one of several intra modes with and without prediction, as well as various efficient inter modes. Each motion-compensated mode corresponds to a specific partition of the MB into fixed-size blocks used for motion description, and up to 16 motion vectors may be transmitted for a MB. In addition, for each MB, a different reference frame can be selected. Finally, a NAL

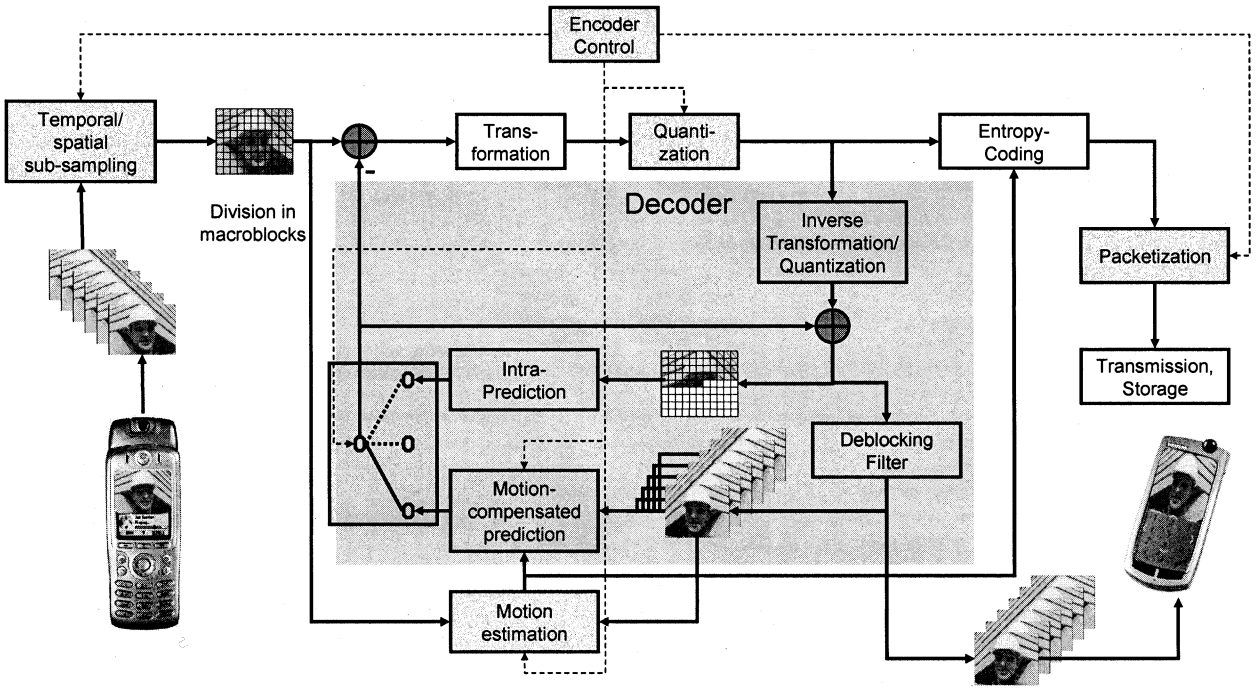


Fig. 5. H.264/AVC encoder realization with coding options.

unit in single-slice mode consists of the coded MBs of an entire frame or a subset of frame. More details on the packetization as well as on the appropriate selection of the presented different modes are discussed in Section IV.

B. Features for Multimedia Messaging and Wireless Packet-Based Streaming

In addition to pure compression efficiency features, additional tools for different purposes have been included in H.264/AVC. We will highlight those features with application to wireless video transmission. Because of the strict separation of encoding, transmission, and decoding, the main issue for MMS is compression efficiency. Other helpful features include the insertion of regular intra frames with instantaneous decoder refresh (IDR) for random access and fast forward. The rate control is typically applied such that video quality is almost constant over the sequence, regardless of the scene complexity except for constraints from the hypothetical reference decoder (HRD) [28]. If time, memory capabilities, and battery power permit, even several encoding passes for optimized rate-distortion (R-D) performance would be possible. On the link layer, reliable transmission strategies as known for data transmission such as file download are used.

Due to on-line transmission and decoding, streaming applications involve more technical challenges than MMS. Usually, pre-encoded data is requested by the user, which inherently does not allow an adaptation to the transmission conditions such as bit rate or error rate in the encoding process. However, the receiver usually buffers the received data and starts play-back after a few seconds. Once starting playback, a continuous presentation of the sequence should be guaranteed. As wireless channels usually show ergodic behavior within a window of a few seconds, reliable transmission schemes can be applied on the link layer, especially when the channel is known at the transmitter or retransmissions for erroneous link layer packets can be used as for example in the acknowledged mode. Slow variance due to distance, shadowing, or varying multiuser topology in the supported cell with renewed resource allocation transform the wireless channel in a slowly varying variable-bit-rate channel. With an appropriate setting of the initial delay and receiver buffer a certain quality of service can be guaranteed [28].

Furthermore, general channel-adaptive streaming technologies, which allow reacting to variable bit-rate channels, have gained significant interest recently. According to [30], these techniques can be grouped into three different categories. Firstly, *adaptive media playout* [31] is a new technique that allows a streaming media client, without the involvement of the server, to control the rate at which data is consumed by the playout process. Therefore, the probability of decoder buffer underflows and overflows can be reduced, but still noticeable artifacts in the displayed video occur. A second technology for a streaming media system is proposed, which makes decisions that govern how to allocate transmission resources among packets. Recent work [32] provides a flexible framework to allow *R-D optimized packet scheduling*. Finally, it is shown that this R-D-optimized transmission can be supported, if media streams are pre-encoded with appropriate packet dependencies, possibly adapted to the channel (*channel-adaptive packet dependency control*) [33].

The latter techniques are supported by H.264/AVC by various means. As the streaming server is in general aware of the current channel bit rate, the transmitter can decide to send one of several pre-encoded versions of the same content taking into account the expected channel behavior. If the channel rate fluctuates only in a small range, frame dropping of nonreference frames might be sufficient resulting in well-known temporal scalability.

Switching of versions can be applied at I-frames that are also indicated as instantaneous decoder refresh (IDR) pictures to compensate large scale variations of the channel rate. In addition, H.264/AVC supports efficient version switching with the introduction of synchronization-predictive (SP) pictures. For more details on SP pictures, see [35]. Note that quality scalable video coding methods such as MPEG-4 fine-grain scalability (FGS) [34] are not supported by H.264/AVC and such extensions of H.264/AVC are currently not planned.

C. Features for Wireless Conversational Services—Error Resilience

A necessary requirement for conversational services is a low end-to-end delay being less that 250 ms. This delay constraint has two main impacts on the video transmitted over wireless bearer services with constant bit rate. Firstly, features have to be provided which allow adapting the bit-rate such that over a short window a constant bit-rate can be maintained. In addition, usually only temporally backward references in motion compensation are used in conversational applications, since prediction from future frames would introduce additional delay. Secondly, within the round-trip time of the communication, the channel usually shows nonergodic behavior and transmission errors cannot be avoided. Whereas the first issue can be solved by adapting the QP appropriately, the second issue requires error-resiliency tools in the video codec itself. More exactly, an error-resilient video coding standard suitable for conversational wireless services has to provide features to combat two problems: on the one hand, it is necessary to minimize the visual effect of errors within one frame. On the other hand, as errors cannot be avoided, the well-known problem of spatio-temporal error propagation in hybrid video coding has to be limited. We will present all error-resilience features included in the H.264/AVC standard and provide further details on the exact application in Section IV.

Packet loss probability and the visual degradation from packet losses can be reduced by introducing slice-structured coding. A slice is a sequence of MBs and provides spatially distinct resynchronization points within the video data for a single frame. No intra-frame prediction takes place across slice boundaries. With that, packet loss probability can be reduced if slices and, therefore, transmission packets are relatively small, since the probability of a bit-error hitting a short packet is generally lower than for large packets (see, e.g., Fig. 4). Moreover, short packets reduce the amount of lost information and, hence, the error is limited and error concealment methods can be applied successfully. However, the loss of intra-frame prediction and the increased overhead associated with decreasing slice sizes adversely affect coding performance and requires additional overhead per slice. Especially for mobile transmission, where the packet size clearly affects loss probability, a careful selection of the packet size is necessary. H.264/AVC specifies several enhanced concepts to reduce the artifacts caused by packet losses within one frame. Slices can be grouped by the use of aggregation packets into one packet and, therefore, concepts such as group-of-block (GOB) and *slice interleaving* [37], [38] are possible. This does not reduce the coding overhead in the VCL, but the costly RTP overhead of up to 40 bytes per packet can be avoided.

A more advanced and generalized concept is provided by a feature that has been called by the proponents *flexible MB ordering* (FMO) [39]. FMO permits the specification of different patterns for the mapping of MBs to slices including checkerboard-like patterns, sub-pictures within a picture (e.g., splitting a CIF picture into four QCIF pictures), or a dispersed mapping of MBs to slices. FMO is especially powerful in conjunction with appropriate error concealment when the samples of a missing slice are surrounded by many samples of correctly decoded slices. For more details on FMO, see [21].

Another error-resilience feature in H.264/AVC is *data partitioning*, which can also reduce visual artifacts resulting from packet losses, especially if prioritization or unequal error protection is provided by the network. For more details on the data-partitioning mode, we refer to [21]. In general, any kind of *forward error protection* (FEC) in combination with interleaving for packet lossy channels can be applied. A simple solution is provided by RFC2733 [40], more advanced schemes have been evaluated in many papers, e.g., [41], [42]. However, in the following, we do not consider FEC schemes in the transport layer as this requires a reasonable number of packets per codeword.

Despite all these techniques, packet losses and resulting reference frame mismatches between encoder and decoder are usually not avoidable. Then, the effects of spatio-temporal error propagation are, in general, severe. The impairment caused by transmission errors decays over time to some extent. However, the leakage in standardized video decoders, such as H.264/AVC, is not very strong, and quick recovery can only be achieved when image regions are encoded in intra mode, i.e., without reference to a previously coded frame. Completely intra-coded frames are usually not inserted in real-time and conversational video applications as the instantaneous bit rate and the resulting delay is increased significantly. Instead, H.264/AVC allows encoding of single MBs for regions that cannot be predicted efficiently as it is also known from other standards. In H.264/AVC, the efficient intra prediction can be constrained to intra MBs only to avoid error propagation from inter-coded MBs to refreshing intra-coded MBs. Another feature in H.264/AVC is the possibility to select the reference frame from the multiframe buffer. Both features have mainly been introduced for improved coding efficiency, but they can efficiently be used to limit the error propagation. Conservative approaches transmit a number of intra-coded MBs anticipating transmission errors. In this situation, the selection of intra-coded MBs can be done either randomly or preferably in a certain update pattern. For details and early work on this subject, see [43]–[45]. Multiple reference frames can also be used to limit the error propagation, for example in *video redundancy coding* schemes (see, e.g., [46]). In addition, a method known from H.263 under the acronym *redundant slices* will be supported in JVT coding. This will allow sending the same slice predicted from different reference frames which provides the decoder the possibility to predict this slice from error-free reference areas. Finally, multiple reference frames can be successfully combined with a feedback channel, which will be discussed in detail among others in Section IV.

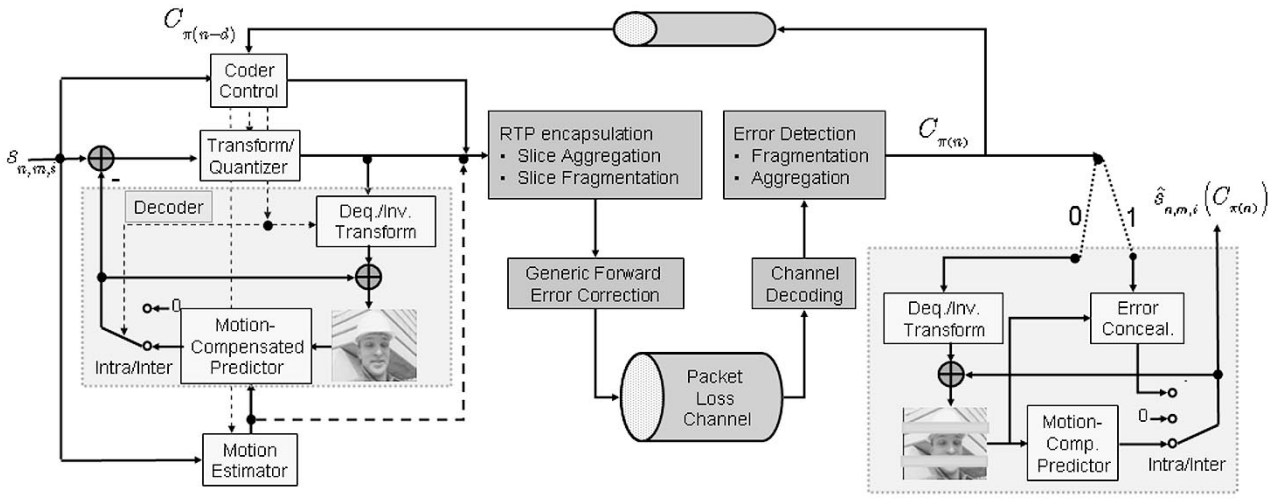


Fig. 6. H.264/AVC in IP-based packet-lossy environment with RTP encapsulation, generic forward error correction, delayed feedback information, and error concealment.

IV. Using H.264/AVC in Wireless Conversational Services

A. Problem Formulation and System Setup

The various error-resilience tools as described in the previous section provide a significant amount of freedom and flexibility for the implementation of H.264/AVC in wireless conversational services. A main criterion for the performance of the system is the appropriate selection of one or several of the mentioned tools together with the exact parameters, e.g., the position and number of intra MBs.

We will discuss in the following channel behavior and error concealment by formalizing the notation for sample representation. The investigated video transmission system is shown in Fig. 6. H.264/AVC video encoding is based on a sequential encoding of frames denoted with the index $n = 1, \ldots, N$ with N being the total number of frames to be encoded. In most existing video coding standards including H.264/AVC, within each frame video encoding is typically based on sequential encoding of MBs (except for FMO) denoted by index $m = 1, \ldots, M$, where M specifies the total number of MBs in one frame and depends on the spatial resolution of the video sequence. The encoding process creates slices by grouping a certain number of MBs. Picture number n and start MB address m_j are binary coded in the slice header. The coded representation of a slice is the payload of a NAL unit. The RTP payload specification specifies simple encapsulation of NAL units. In addition, several NAL units can be combined into one aggregation packet or one NAL unit can be fragmented into several transport packets [19], [21].

For notational convenience, let us define the number of transmission packets to transmit all frames up to n as $\pi(n)$. With that, we can define the packet loss or channel behavior c as a binary sequence $\{0,1\}^{\pi(n)}$ indicating whether a slice is lost (indicated by 1) or correctly received (indicated by 0). Obviously, if a NAL unit with the encapsulated slice is lost, all MBs contained by this slice are lost. It can be assumed that the decoder is aware of any lost packet as discussed previously. The channel-loss sequence is random and, therefore, we denote it as $C_{\pi(n)}$, where the statistics are in general unknown to the encoder. According to Fig. 6, in addition to the forward link it possible that a low-bit-rate reliable back-channel from the decoder to the encoder is available which allows reporting a d-frame delayed version $C_{\pi(n-d)}$ of the observed channel behavior at the decoder to the encoder. In RTP/IP environments, this is usually based on RTCP messages, and in wireless environments, internal protocols might be used.

The decoder processes the received sequence of packets. Whereas correctly received packets are decoded as usual for the lost packet, an error-concealment algorithm has to be invoked. The reconstructed sample $s_{n,m,i}$ at position i in MB m and frame n depends on the channel behavior and on the decoder error concealment. In inter-coding mode, i.e., when motion-compensated prediction (MCP) is utilized, the loss of information in one frame has a considerable impact on the quality of the following frames, if the concealed image content is referenced for MCP. Because errors remain visible for a longer period of time, the resulting artifacts are particularly annoying. Therefore, due to the motion-compensation process and the resulting error propagation, the reconstructed image depends not only on the lost packets for the current frame, but in general on the entire channel-loss sequence $C_{\pi(n)}$. We denote this dependency by $\hat{s}_{n,m,i}(o, C_{\pi(n)})$

In the following, we will discuss appropriate extensions of the encoder, the decoder, and the transmission environment, which are either necessary or at least beneficial to enhance the quality of the transmitted video.

B. Decoder Extensions—Loss Detection and Error Concealment

The H.264 standard defines how a decoder reacts to an error-free bit stream. In addition, a decoder implementation has also to deal with transmission errors. As we discussed earlier, it is assumed that bit errors are detected by the lower layer

entities and any remaining transmission error results in a packet loss. Therefore, we address the reaction of the decoder to slice losses. Two major issues are important for an error-resilient decoder: a robust video decoder has to detect transmission errors, and appropriate concealment on detected errors has to be applied. In this section, we present how the H.264 test model decoder meets the goal of error resiliency.

The error detection capabilities of the test model decoder are based on two assumptions about the error detection operation of the underlying system. First, bit-erroneous slices are discarded prior to passing the slices to the test model decoder. Second, received data is buffered in a way that the correct decoding order is recovered. In other words, the test model decoder expects noncorrupted slices in a correct decoding order. Temporal and spatial localization of lost packets is left to the decoder. In particular, the decoder has to detect if an entire picture or one or more slices of a picture were lost. Losses of entire pictures are detected using frame numbers associated with each frame and carried in slice headers. A frame number n is incremented by one for each coded and transmitted frame that is further used for motion compensation. These frames are herein referred to as reference frames. For disposable nonreference pictures, such as conventional B-pictures, the frame number is incremented relative to the value in the most recent reference frame that precedes the disposable picture in the bit-stream order.

The decoder generates a slice structure for each received single slice packet and forwards it to the VCL decoder which maintains a state machine that keeps track of the expected frame number n_e. Moreover, the decoder maintains a loss indication for each MB within the current picture. At the beginning of a new picture, the binary map is reset to indicate that all MBs were lost. If the frame number n of the next slice to be decoded equals n_e, the decoder decodes the slice and updates the binary map. If n is greater than n_e, it is deduced that all the received slices of the previous picture have been decoded. Then, the binary map is investigated, and if the picture is not fully covered by correctly decoded MBs, a slice loss is inferred and losses are concealed as presented in the following. Moreover, if n is greater than $n_e + 1$, the decoder infers a loss of pictures and inserts concealed pictures to the reference picture buffer as if the lost pictures were decoded. The concealment is accomplished by copying the previous decoded picture. Finally, the decoder resets the binary map and decodes the next slice of the next picture.

Error concealment is a nonnormative feature in the H.264 test model. The target for the selected error concealment is to provide a basic level of error resiliency for the decoder. Any error-robust coding scheme proposed for H.264 should be compared against the H.264 test model equipped with the selected error concealment algorithms. Two well-known concealment algorithms, weighted pixel value averaging for intra pictures [47] and boundary-matching-based motion vector recovery for inter pictures [48], were tailored for H.264 as summarized below and described in details in [49] and [50].

Weighted pixel value averaging operates as follows. If a MB has not been received, it is concealed from the pixel values of spatially adjacent MBs. If a lost MB has at least two correctly decoded neighboring MBs, only these neighbors are used in the concealment process. Otherwise, previously concealed neighboring MBs take part in the process, too. Each pixel value in a MB to be concealed is formed as a weighted sum of the closest boundary pixels of the selected adjacent MBs. The weight associated with each boundary pixel is relative to the inverse distance between the pixel to be concealed and the boundary pixel. The performance of the intra concealment method is shown in Fig. 7

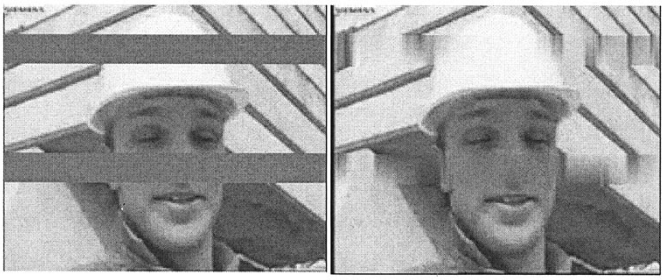


Fig. 7. Intra frame error concealment.

In the motion vector recovery algorithm, the motion activity of the correctly received slices of the current picture is investigated first. If the average length of a motion vector component is smaller than a pre-defined threshold (currently 1/4 pixels), all the lost slices are copied from co-located positions in the reference frame. Otherwise, motion-compensated error concealment is used, and the motion vectors of the lost MBs are predicted as described in the following paragraphs. The image is scanned MB-column-wise from left and right edges to the center of the image. Consecutive lost MBs in a column are concealed starting from top and bottom of the lost area toward to center of the area. This processing order is used to ensure that lost MBs at the center of an image are concealed using as many neighboring concealed MBs as possible.

Each 8×8 luminance block of a MB to be concealed is handled separately. If a block has spatially adjacent blocks whose motion vectors are correctly received, these motion vectors and their reference pictures are used to obtain a candidate prediction block. If all the motion vectors in the adjacent blocks are lost, neighboring concealed motion vectors are used similarly. In addition, the spatially co-located block from the previous frame is always one of the candidates. The candidate prediction block whose boundary matching error is the smallest is chosen. The boundary matching error is defined as the sum of the pixel-wise absolute differences of the adjacent luminance pixels in the concealed block and its decoded or concealed neighbor blocks. The lost prediction error block is not concealed. The performance of the inter-frame concealment is shown in Fig. 8.

C. Encoder Extensions

Let us now consider algorithms and rules for the appropriate selection of different encoding parameters according to the presentation in Fig. 5 for wireless conversational services. First, the rate control has to guarantee that the delay jitter is as small as possible. A good choice is to adapt for each frame to obtain almost constant encoding, but keep the QP within one frame constant. In the case that the highest QP cannot achieve the required bit rate, frame dropping is introduced; otherwise, the frame rate

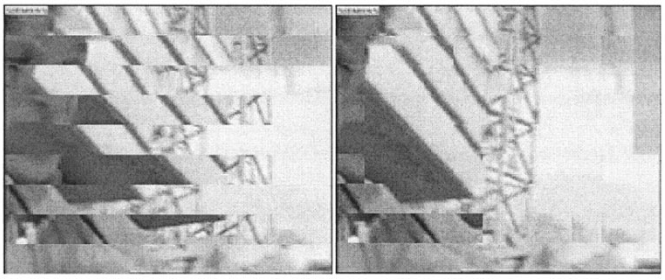


Fig. 8. Inter-frame error concealment.

is kept constant. Obviously, this results in varying quality depending of the complexity of the scene. Low-motion scenes with little complexity have higher quality than for example frames at the beginning of a sequence or at a scene cut.

As discussed in Section III, H.264/AVC provides a large flexibility on the selection of appropriate coding modes. The concept of selecting appropriate coding options in optimized encoder designs for many video coding standards is based on R-D optimization algorithms [51],[52]. The two cost terms "rate" and "distortion" are linearly combined and the mode is selected such that the total cost is minimized. This can be formalized by defining the set of selectable coding options for one MB as $\mathcal{O}$. Assuming the introduced distortion when encoding with a certain mode o as $D(o)$ and the corresponding rate as $R(o)$, the rate-constrained mode decision selects the coding option o^* such that the Lagrangian cost functional is minimized, i.e.,

$$o^* = \arg\min_{o\in\mathcal{O}}(D(o) + \lambda R(o)) \tag{1}$$

with λ being the Lagrange parameter for appropriate weighting of rate and distortion. In the H.264/AVC test model, the Lagrangian mode selection is used for motion vector search as well as MB mode and reference frame selection. Then, for the distortion $D(o)$, the sum of squared sample differences (SSD) or the sum of absolute sample difference (SAD) are used, and for the rate $R(o)$, the number of bits for the encoding is used. The selection of the Lagrange parameter depends on the selected QPs (for details, see [53]). In addition, the mode selection might be extended by selecting the appropriate QP for each MB, as the QP can be changed at least in small ranges for each MB. This is not considered in the following.

As discussed in Section III-C, the tools for increased error resilience, in particular those to limit error propagation, do not significantly differ from those used for compression efficiency. Features like multiframe prediction or intra MB coding are not exclusively error-resilience tools. They are also used to increase coding efficiency in error-free environments providing a trade-off that is left to the encoder. This also means that bad decisions at the encoder can lead to poor results in coding efficiency or error resiliency or both. Therefore, the selection of the coding mode according to (1) is modified taking into account the influence of the random lossy channel. In the case of error-prone transmission, the distortion in (1) is replaced with the expected decoder distortion when encoding with mode o. Assuming that the encoder can access information about the channel statistics $C_{\pi(n)}$, the encoder can get an estimate of the expected decoder distortion $D_{n,m}(o, C_{\pi(n)})$ when encoding MB m in frame n at the decoder by the expected distortion as

$$D_{n,m}\left(o, C_{\pi(n)}\right) = \sum_{i=1}^{I} E_{C_{\pi(n)}} \left| s_{n,m,i} - \hat{s}_{n,m,i}\left(o, C_{\pi(n)}\right)\right|^2 \tag{2}$$

where the expectation is over the random process characterizing the channel $C_{\pi(n)}$. With the expected distortion measure, the mode selection for lossy channels is identical to that in (1) except for modified distortion term according to (2). However, the same parameter λ [54] and the same set of possible coding options $\mathcal{O}$ are used.

This leaves the problem of computing the expected decoder distortion at the encoder which depends on the coding mode o, the channel statistics $C_{\pi(n)}$, and the applied error concealment in the decoder. The estimate of the expected sample distortion in packet loss environment has been addressed in several publications. For example, in [58], [55], or [56], methods to estimate the distortion introduced due the transmission errors and the resulting drift are presented. A similar approach has recently been proposed within the H.264/AVC standardization process which attempts to measure the drift noise between encoder and decoder [57]. In suboptimal approaches [55]–[57], the quantization noise and the distortion introduced by the transmission errors are linearly combined. The encoder keeps track of an estimated sample distortion and, therefore, requires additional complexity in encoder, which is dependent on the actual method chosen.

The most recognized out of these methods, called recursive optimal per-sample estimate (ROPE) algorithm [58], provides an estimation by keeping track of the first- and second-order moment of $\hat{s}_{n,m,i}, E\left\{\hat{s}_{n,m,i}(o, C_{\pi(n)})\right\}$, and $E\left\{\hat{s}^2_{n,m,i}(o, C_{\pi(n)})\right\}$, respectively. For H.263-like encoding ROPE can provide an exact estimate of the expected decoder distortion. As two moments for each sample have to be tracked in the encoder, the added complexity of ROPE is approximately twice the complexity of the decoder. However, the extension of the ROPE algorithm to H.264/AVC is not straightforward. The in-loop deblocking filter, the fractional sample motion accuracy, the complex intra prediction and the advanced error concealment require taking into account the expectation of products of samples at different positions to obtain an accurate estimation which makes the ROPE either infeasible or inaccurate in this case.

Therefore, a powerful yet complex method has been introduced into the H.264/AVC test model to estimate the expected decoder distortion [54]. Let us assume that we have K copies of the random variable channel behavior at the encoder, denoted as $C_{\pi(n)}(k)$. Additionally, assume that the set of random variables $C_{\pi(n)}(k), k = 1, \cdots, K$ are *identically* and *independently* distributed (*i.i.d.*). Then, as $K \to \infty$, it follows by the strong law of large numbers that

$$\frac{1}{K}\sum_{k=1}^{K}\left| s_{n,m,i} - \hat{s}_{n,m,i}\left(C_{\pi(n)}(k)\right)\right|^2 = E_{C_{\pi(n)}}\left| s_{n,m,i} - \hat{s}_{n,m,i}\left(C_{\pi(n)}\right)\right|^2 \tag{3}$$

holds with probability 1. An interpretation of the left-hand side leads to a simple solution of the previously stated problem to estimate the expected sample distortion. In the encoder, K copies of the random variable channel behavior and the decoder are operated. The reconstruction of the sample value depends on the channel behavior $C_{\pi(n)}(k)$ and the decoder including error concealment. The K copies of channel and decoder pairs in the encoder operate independently. Therefore, the expected distortion at the decoder can be estimated accurately in the encoder if K is chosen large enough. However, the added complexity in the encoder is obviously at least K times the decoder complexity. For details on the implementation as well as comparison of suboptimal modes based on ROPE, we refer to [54].

It was later demonstrated in [59] and [60] that a technique known as isolated regions, which combines periodical intra MB coding and limitation of inter prediction, can provide competitive simulation results compared to the error-robust MB mode selection of H.264 test model, and the best simulation results were obtained when the two MB mode-selection algorithms were combined.

The packet size is usually upper bounded by a MTU size of the underlying network. However, if bit errors cause entire packet losses, it might be an advantage to reduce the packet size. Conservative approaches limit the packet or (in our simulation environment) slice size to a fixed number of MBs, e.g., a line of MBs is transmitted in one packet. However, this makes especially important image parts very susceptible to packet losses as for example suddenly appearing objects need intra coding or high motion areas require an extensive motion vector rate. A more suitable, but still conservative, approach is the introduction of slice boundaries such that slices have almost the same length in number of bytes. This makes all packets similarly susceptible to bit errors. This method has been proven to work quite well already for MPEG-4 and can also be applied with the flexible encoding framework of H.264/AVC.

D. Exploiting Feedback in H.264/AVC

Multiple reference frames used in error-prone environment can most successfully be combined with a feedback channel. Due to the bidirectional nature of conversational applications, it is common that the encoder has knowledge of the experienced channel at the decoder, usually with a small delay. In our framework this can be expressed by the knowledge of a d-frame delayed version of the random channel $C_{\pi(n-d)}$ at the encoder. This characteristic can be conveyed from the decoder to the encoder by acknowledging correctly received slices (ACK), sending a not-acknowledge message (NAK) for missing slices or both types of messages. In general, it can be assumed that the reverse channel is error-free and the overhead is negligible. In common data transmission applications, the lost packets would obviously be re-transmitted. However, in a low-delay environment, this is not feasible, but the observed channel characteristic are still useful at the encoder even if the erroneous frame has already been decoded and concealed. The support of these techniques is out of the focus of the standardization of H.264, and a full support of the presented concepts might not be possible with current transport and control protocols. However, it seems worth to mention and discuss these concepts in here in detail as it shows the flexibility of H.264 as well as it provides motivation for inclusion of feedback in system designs.

In previous standards and transport environments similar approaches have already been discussed, usually limited by the reduced syntax capabilities of the video standard. A simple yet powerful approach suitable for video codecs using just one reference frame such as MPEG-2, H.261, or H.263 version 1 has been introduced in [61] and [62] under the name *error tracking*. When receiving a NAK on parts of frame $n-d$ or the entire frame $n-d$, the encoder attempts to track the error to obtain an estimate of the quality of frame $n-1$, which serves as reference for frame n. Appropriate actions after having tracked the error are for example presented in [56], and [61]–[64]. Note that with this concept, error propagation in frame n is only removed if frames $n-d+1,\ldots,n-1$ have been received at the decoder without any error.

A technique addressing the problem of continuing error propagation has been introduced, among others, in [65]–[67] under the acronym *NEWPRED*. Based on these early nonstandard compliant solutions in H.263 Annex N [68], a reference picture selection (RPS) for each GOB is specified such that the NEWPRED technique can be applied. RPS can be operated in two different modes. In the negative acknowledgment mode (NAM), the encoder only alters its operation in the case of reception of a NAK. Then, the encoder attempts to use an intact reference frame for the erroneous GOBs. To completely eliminate error propagation, this mode has to be combined with independent segment decoding (ISD) according to Annex R of H.263 [68]. In the positive acknowledgment mode (PAM), the encoder is only allowed to reference confirmed GOBs. If no GOBs are available to be referenced, intra coding has to be applied. NEWPRED allows to completely eliminate error propagation in frame n, even if additional errors have occurred in frames $n-d+1,\ldots,n-1$.

The flexibility provided in H.263 Annex U [68] and especially H.264/AVC to select the MB mode and reference frames on MB or subMB basis allows incorporating NEWPRED in a straightforward manner [56]. Therefore, let us define three different states of transmitted packets at the encoder: ACK, NAK, and outstanding acknowledgment (OAK). Then, based on the exploitation of these messages in the encoder, we discuss three modes which can elegantly be integrated into the R-D optimized mode selection according to (1).

1) Feedback Mode 1: Restricting Reference Areas to ACK: In this case, the reference area is restricted to frames or slices which have been acknowledged. This can be formalized in the context of (1) by applying the encoding distortion $D(o)$ in (1), but altering the accessible coding options $\mathcal{O}$ such that only acknowledged areas can be used for reference. In addition, if no reference area is available, or if no satisfying match is found in the allowed area, intra coding is applied. Although in the presentation of a single frame an error might be visible, error propagation and reference frame mismatch between encoder can be completely avoided independent of the error concealment applied at the decoder if correctly decoded samples are not altered. However, in the used test model JM1.7, the deblocking filter operation in the motion-compensation

loop is applied over slice boundaries, which restricts the area to be referenced significantly, or a complete removal of encoder and decoder mismatch is not possible. Although the influence of this mismatch is, in general, negligible, in the final design of H.264/AVC the deblocking filter can adaptively be switched on and off at slice boundaries to allow mismatch-free operation.

2) Feedback Mode 2: Referencing ACK and Error Concealed NAK: In this case, the reference area is again restricted; however, in addition to acknowledged areas, also the areas which the decoder signaled as lost can be referenced. Therefore, the reference frames in the encoders multi-frame frame buffer are updated with reception of each ACK and NAK by applying the identical error concealment as the decoder applies. This is obviously very critical, as the error concealment is—for good reasons—nonnormative in common video coding standards including H.264/AVC. Only if the encoder is aware of the decoder's error concealment by any external means, this mode can provide benefits compared to feedback mode 1. In the context of the mode decision in (1), we have again a restricted set of coding modes $\mathcal{O}$ and, in addition, the encoding distortion is replaced by the deterministic decoder distortion.

3) Feedback Mode 3: Unrestricted Reference Areas With Expected Distortion Updating: In [56] and [58], techniques have been proposed which allow combining the error-resilient MB mode selection with feedback information. In this case, the expected decoder distortion computation is altered such that, for all packets $1, \ldots, \pi(n-d)$, the channel is deterministic at the encoder, and the expected distortion is computed for the packets with status OAK. In our case, packets containing MBs in frames $\pi(n-d)+1, \ldots, \pi(n)$ are random. The set of selectable coding options $\mathcal{O}$ is not altered compared to pure coding efficiency mode selection. This method is especially beneficial compared to mode 1 and 2, if the feedback is significantly delayed. In the case of the multiple decoder approach, this can be integrated by applying feedback mode 2 not only to the reference frames, but also to all decoders in the encoder. In combination with ROPE, however, the complexity of this method increases since the moments of the previous d frames have to be re-tracked [58].

V. Selected Simulation Results for Different System Concepts

A. Simulation Conditions and Evaluation Criteria

In the following, we will present simulation results based on the test conditions that show the influence of the selection of different error-resilience features for the quality of the decoded images. For all following tests the H.264/AVC test model software version JM1.7 is used. Note that in the final standard [1], the zig-zag scanning and run-length coding is replaced by context-adaptive variable length codes (CVLC). All UVLCs are replaced by CVLC, which are adapted to the statistics of different syntax elements. In addition, quantizer values have been shifted. However, all changes from JM1.7 to the final standard are of little relevance to the presented results and conclusions in this paper.

The reported PSNR is the arithmetic mean over the decoded luminance PSNR over all frames of the encoded sequence and over 100 transmission and decoding runs. The 100 starting positions for the error patterns have been selected such that they are almost equally distributed over the error pattern. For all comparable results, the same starting positions and, therefore, the same channel statistics have been applied. In addition, we present results for the cumulative distribution of the decoded luminance PSNR for each frame, i.e., the likelihood that the PSNR of the frames of the sequence is smaller than the value on the x axis. This shows the variance in the decoded quality. It is assumed that the high-level syntax parameters have been transmitted in advance and out-of-band applying a reliable setup protocol. The NAL overhead, the RTP/UDP/IP overhead after RoHC, and the link layer overhead is taken into account in the bit-rate constraints according to [25].

For the following simulations, we concentrate on test case 5 from [25], which includes the QCIF test sequence "Foreman" (30 Hz, 300 frames) at a constant frame rate of 7.5 Hz at bit-error pattern 3 according to Table II, i.e., a mobile user at 3 km/h and maximum bit-rate 64 kbit/s. This is the most critical case in terms of error probability, additional test results will be made available online[2]. For the following tests, entropy coding based on the UVLC and only one reference frame has been applied, if not stated otherwise. The encoded sequences are I-P-P-P… sequences; B-pictures are excluded due to the unacceptable delay involved in the encoding process. Note that due to the repeated decoding of an encoded file, every 75th frame is an I-frame, i.e., an I-frame occurs every 10 s. Constrained intra has been used to avoid error propagation from inter MBs to intra MBs. For all encoding runs, R-D optimized mode selection and motion vector selection according to (1) have been used. The distortion $D(o)$ and the set of coding modes $\mathcal{O}$ is appropriately selected according to the applied features. In the case of using the expected decoder distortion, the number of decoders operated in the encoder has been fixed to $K = 100$. Unless stated otherwise, the error concealment in the decoder is based on the advanced error concealment as presented in Section IV-B, whereas the multiple decoders in the encoder always apply previous frame concealment. This reduces encoding complexity significantly and results only in negligible performance losses for the investigated cases.

B. Low-Delay Rate Control

Version JM1.7 of the H.264/AVC test model encoder which is used in the experiments does not include a rate control to achieve a constant bit rate for wireless conversational services. Moreover, the rate control introduced in later versions of H.264/AVC test model encoder is not suitable for constant rate encoding. For this reason we have added a rate control which provides an almost constant bit-rate encoding for each frame by adapting the QP for each frame appropriately. Therefore, before we investigate the error-resilience features in H.264/AVC, we will first focus on the effect of bit-rate control that is necessary for constant bit-rate conversational applications. Fig. 9 shows the cumulative distribution of the encoding PSNR for the applied rate control and a fixed QP such that both encoded files result in the

[2]For additional simulation results, we refer to http://www.lnt.ei.tum.de/~stockhammer.

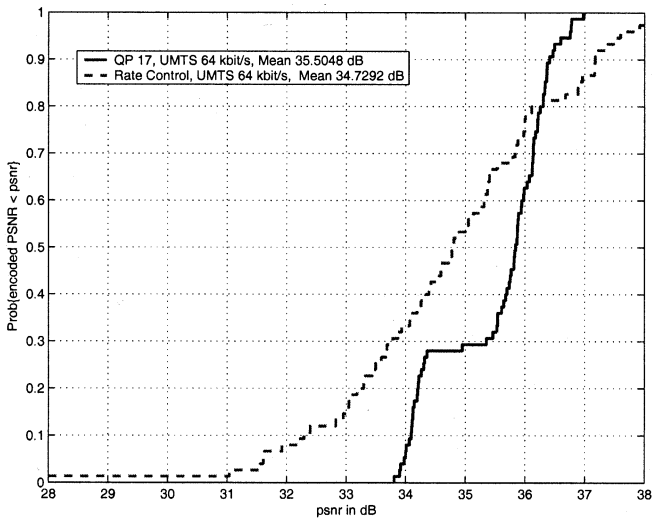


Fig. 9. Cumulative distribution of encoding luminance PSNR for each frame for constant QP 17 and constant bit rate such that rate constraint of 64 kbit/s for UMTS is fulfilled.

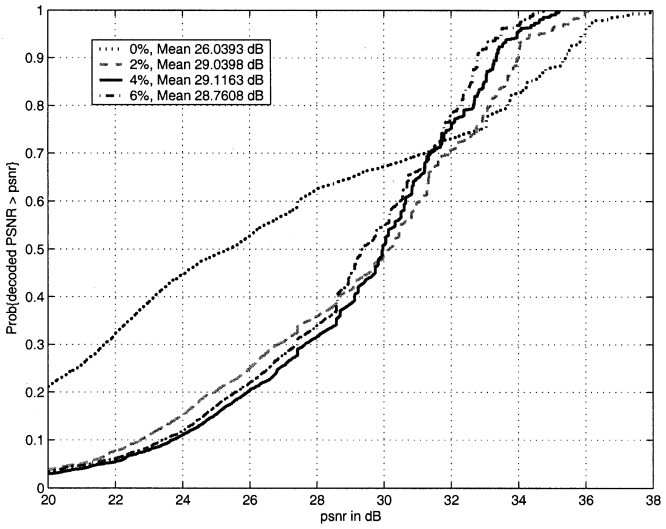


Fig. 10. Cumulative distribution of decoded PSNR for different NAL unit erasure rates for the estimation of the expected distortion in the encoder.

same total bit-rate, and, additionally the file can be transmitted over the 64 kbit/s link including NAL unit, packet, and link layer overhead. However, the fixed QP results in an extremely varying data rate and, therefore, the introduced delay jitter is not acceptable. In the encoding process, no error-resilience tools have been used, i.e., one frame is encapsulated into one NAL unit, for $D(o)$ the encoding distortion is applied, and the set of coding modes $\mathcal{O}$ is unrestricted.

The average PSNR for the rate control is about 0.8 dB below the PSNR for the fixed QP. In addition, the probability for low encoded PSNR is significantly higher as for complex scenes the QP has to be adapted appropriately. Therefore, it can be seen that the rate control necessary for conversational applications involves an inherent decrease in quality if the average PSNR is the measure of interest.

C. $R - E\{D\}$-Optimized MB Mode Selection

In this section we investigate the performance of the system in case that the channel statistics are taken into account into the selection of the coding options in the encoder. For this purpose we replace the encoding distortion $D(o)$ in (1) by the expected decoder distortion assuming a channel producing independent packet losses with probability p. As we use the mapping of one frame to one transport packet and apply the strict rate control producing almost a constant number of bytes for each encoded frame, the size of each frame results in roughly 1000 bytes. The corresponding loss rate for packets of this size for bit-error pattern 3 is approximately 4%–5% according to Fig. 4. Fig. 10 shows the cumulative distribution of decoded PSNR for different NAL unit erasure rates $p = \{0, 0.02, 0.04, 0.06\}$ for the estimation of the expected distortion in the encoder. Looking at the results for the average PSNR, it can be seen that the introduction of loss-aware R-D optimization $(p > 0)$ increases the decoded quality significantly compared to the results with pure encoding distortion $(p = 0)$. The average PSNR increases by at least 3 dB when compared to the to pure R-D optimization. The advantage of $R-E\{D\}$ optimization is even more evident when looking at the cumulative distribution of the different strategies. Whereas in the case of no error resilience, the probability of bad frames (frames below 22 dB) is at an unacceptable ratio of about 30%, for loss-aware coding, this is reduced significantly to less than 8%. It is also obvious that if the expected error rate matches the experienced error rate on the channel, the performance is optimal (see $p = 4\%$). However, it can also be seen that a mismatch in the expected error rate in the encoder does not have significant influence. The performance of $p = 2\%$ and $p = 6\%$ is only slightly inferior to the matching expected error rate. Therefore, a rough estimation of the expected decoder distortion at the encoder seems to be good enough for a good mode selection. Note the significant loss in average PSNR for wireless transmission compared to the error-free transmission according to Fig. 9 of more than 5 dB.

D. Slices and Error Concealment

The introduction of slices in the encoding has two beneficial aspects when transmitting over wireless channels, but adversely affects the coding efficiency due to increased packet overhead and reduced prediction within one frame, as e.g., motion vector prediction and spatial intra prediction is not allowed over slice boundaries. The two positive effects with the introduction of slices are the reduced error probability of shorter packets (see Fig. 4) and, the re-synchronization possibility within one frame. The latter technique allows restarting the decoding process at each slice and, in addition, it allows applying advanced error concealment as for example presented in Section IV-B. However, for packet-lossy transmission over the Internet, the introduction of slices does not, in general, provide gains in the decoded quality [69] as long as the slice size is below the MTU size, as the loss of a packet is then independent of its length. This is different for wireless transmission: Fig. 11 shows the average decoded PSNR for different number of packets per frame N_p and MB mode decision with encoding distortion. The experimental conditions are similar to the above section otherwise. For a given number of packets per frame, the size of each packet is selected such that the packets roughly have the same number of bytes. This makes their susceptibility to bit errors almost identical.

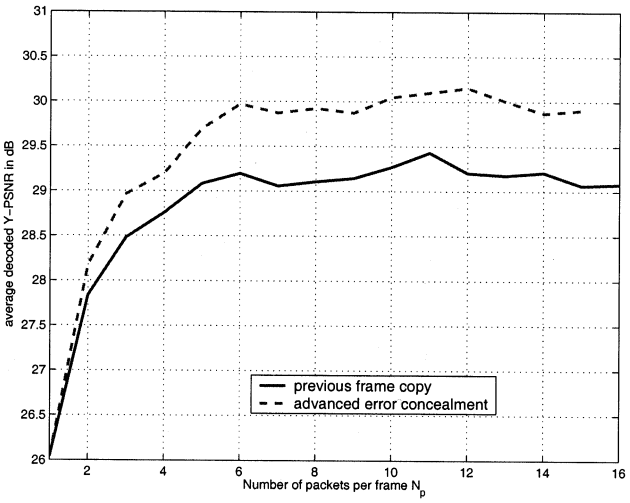


Fig. 11. Average decoded PSNR for different number of packets per frame and different error-concealment schemes; mode selection with encoding distortion.

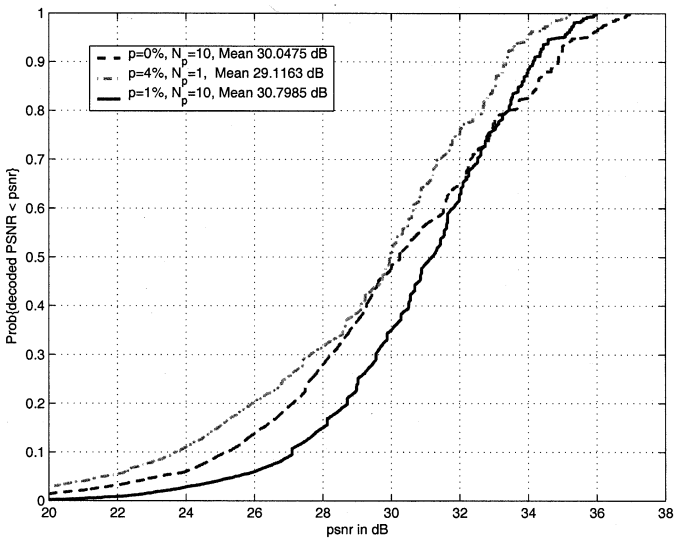


Fig. 12. Cumulative distribution of decoded PSNR for different error-resilience strategies: $R - E\{D\}$-optimized intra updates with and without slice structuring and delay $d = 2$ for different assumed loss probabilities p.

The average decoded PSNR is shown for previous frame copy error concealment and advanced error concealment according to Section IV-B. The benefit of introducing slice structuring in the encoding process is obvious from the results. Compared to the "one frame—one packet" mode indicated by $N_p = 1$ the introduction of shorter packets increases the decoded quality significantly for both error concealment methods. For about $N_p = 6$, the curve flattens out and decreases again for higher $N_p > 12$ due to increasing packet overhead and the reduced compression efficiency. Although a clear maximum cannot be determined, for the wireless environment according to our test conditions, a reasonable number of packets per frame is about 10. The resulting packet size in this case is in the range of 100 bytes. However, note that for the simulations, RoHC was applied, which reduces the typical IP/UDP/RTP overhead from 40 bytes to about 3 bytes and, therefore, the packetization overhead is less significant. The benefits of the advanced error concealment are also obvious from Fig. 11. As can be seen, the gains for advanced error concealment increase with increasing number of packets, as better concealment is possible due to increased number of neighboring MBs in case of losing a single slice.

For the experiments in Fig. 11, no explicit means to reduce the error propagation have been used. However, we can obviously combine the MB mode selection based on the expected decoder distortion with the slice structured coding. As indicated, the loss rate for decreased packet size decreases compared to the single packet for one frame. The loss rate can be estimated by dividing the approximated number of bytes for each frame, roughly 1000 bytes, by the number of packets. The loss probability can then again be estimated with Fig. 4 using the resulting average packet length. The combination of slice-structured coding and adaptive intra MB updates has been investigated and a comparison with the best cases of the previous two system designs is provided in Fig. 12 based on the cumulative distribution of the decoded PSNR. For the slice-structured coding with encoding distortion $(p = 0\%)$ the number of packets is selected as $N_p = 10$. For the expected decoder distortion without slice-structuring $(N_p = 1)$, the adapted loss-rate $p = 4\%$ is chosen. Finally, for the combination of slice-structured coding and channel-adaptive intra updates, the number of packets per frame is selected as $N_p = 10$ and, therefore, the appropriate loss probability to compute the expected decoder distortion according to Fig. 4 is about $p = 1\%$.

The average decoded PSNR indicates that an optimized combination of both error-resilience schemes outperforms each of the presented error-resilience schemes significantly. The result that slice-structured coding is superior to intra updates cannot be generalized for all sequences. The repeated I-frame insertion and the camera pan in the test sequence "Foreman" results in a significant amount of intra information, even if only the encoding distortion is chosen in the mode selection. This might change for different or longer sequences with less intra information. From the cumulative distribution, it can be observed that the probability for bad frames below 22 dB in PSNR is almost vanishing for the combined mode. The presented results indicate that slices in combination with advanced error concealment significantly outperform the single packet for one frame approach. However, the loss compared to error-free transmission is still about 4 dB in average PSNR.

From the results, it can be conjectured that for wireless transmission as investigated in this case, other approaches, which reduce the artifacts within one frame, might provide additional benefits. This includes concepts such as FMO, slice interleaving, or even generic forward-error correction in combination with a NAL unit fragmentation scheme as recently introduced in the draft RTP payload specification for H.264/AVC [19]. Also, data partitioning with appropriate unequal error protection might enhance the quality of the decoded video. In addition, it is conjectured from the results that a better adaptation of the link layer error protection scheme with appropriate interleaving could increase the overall system performance. These features are currently investigated and are subject of future work.

E. Exploiting Feedback in Video Encoding

Finally, we investigate a system which exploits multiple reference frames and network feedback. We restrict our simula-

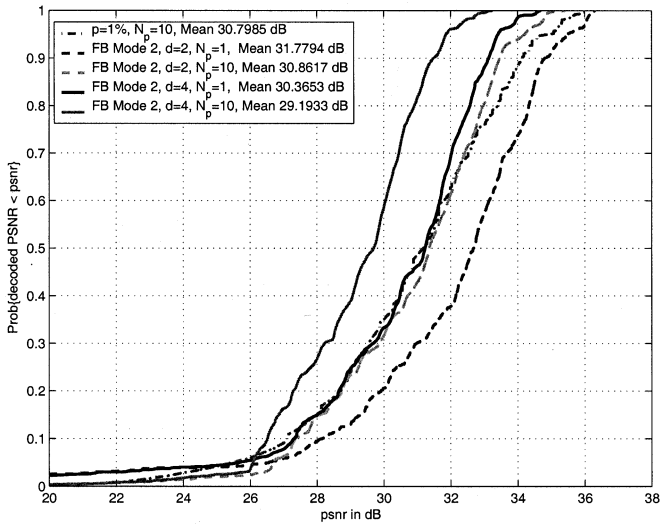


Fig. 13. Cumulative distribution of decoded PSNR for different error-resilience strategies: R-D-optimized intra updates with slice structuring, and feedback mode 2 with and without slice structuring for delay $d = 2$ and $d = 4$.

tion results to feedback mode 2 according to the presentation in Section IV-D with advanced error concealment. There are two reasons for this. On the one hand, the deblocking and mismatch problem present for JM1.7 can be avoided. On the other hand, it has been shown previously that feedback mode 1 and feedback mode 2 result in almost identical performance [70]. Therefore, we chose feedback mode 2 due to the simpler implementation, at least in our simulation environment. Feedback mode 3 is excluded as, on the one hand, the complexity of this mode is rather high and, on the other hand, as we operate with low feedback delays, the expected benefits are only marginal. In contrast to the previous simulations, we use five reference frames for the feedback mode. Fig. 13 shows the cumulative distribution of decoded PSNR for different error-resilience strategies: R-D-optimized intra updates with slice structuring, as well as feedback mode 2 with and without slice structuring for delay $d = 2$ and $d = 4$ frames, which corresponds to a round-trip time of about 250 and 500 ms, respectively.

Let us focus on the delay $d = 2$ case first. The results indicate that the optimized intra mode with slice structuring and MB mode selection with expected decoder distortion and feedback mode 2 perform very similar based on the cumulative distribution and the average decoded PSNR. The feedback mode might still be beneficial in this case as the complex estimation of the decoder distortion is not necessary for the feedback case. However, much more interesting is the case with feedback and no slice structuring. In contrast to the case without feedback (see Fig. 12), the renouncement of slice-structured coding provides a significantly higher average decoded PSNR. Initially, this is obviously surprising, as packet loss rate is still much lower when several slices are used and also the visual effects for a decoded frame when losing a single slice should be lower than in case of losing an entire frame. The first effect can indeed be observed from the cumulative distribution. The probability of bad frames (PSNR below 22 dB) is higher for $N_p = 1$ than for $N_p = 10$. However, in the case of no errors, the increased coding efficiency when not using slices provides many frames with significantly higher PSNR than for slice structuring. As we avoid error propagation, the correctly received frames are really error-free, which is not the case if we use intra updates. Therefore, if feedback information is available and several completely lost frames are tolerable, it is better to use no slice structured coding than harming the compression efficiency. For increased feedback delay $d = 4$, the curves are shifted to the left compared to feedback delay 2. However, the $d = 4$ and $N_p = 1$ performs almost as well as the best case without feedback. Therefore, in the case of available feedback, this very simple system without considering expected decoder distortion and slice structuring and just relying on multiple reference frames outperforms many highly sophisticated error-resilience schemes as long as the delay of the feedback is reasonable. The combination of these methods according to feedback mode 3 is currently investigated and should allow adaptively selecting the best methods, however, with significantly increased encoding complexity.

VI. Conclusions

H.264/AVC promises some significant advances of the state-of-the-art of standardized video coding in mobile applications. In addition to excellent coding efficiency, the design of H.264/AVC also takes into account network adaptation providing large flexibility for its use in wireless applications. The tools provided in H.264/AVC for error resilience do not necessarily differ from the compression efficiency features such as intra MBs or multiple reference frames. However, in the case of error-prone transmission, the selection methods have to be changed by using the expected decoder distortion or by restricting the set of accessible coding options. In experimental results based on common test conditions, it has been shown that in case without any feedback, several slices in combination with channel-adaptive R-D optimized mode selection is a promising approach. In this case, further investigation with advanced error-resilience tools such as flexible MB ordering, data partitioning, and generic forward error correction, might provide benefits. However, in the case of available feedback, the application of multiple reference frames to exclude error propagation without slice structuring provides excellent results.

Acknowledgment

The authors would like to thank T. Oelbaum, D. Kontopodis, Y.-K. Wang, V. Varsa, and A. Hourunranta for implementing and testing parts of the algorithms, V. Varsa and G. Liebl for providing the test conditions and software simulator, S. Wenger, N. Färber, K. Stuhlmüller, E. Steinbach, and B. Girod for useful discussions, and JVT for the collaborative work and the technically outstanding discussions and contributions.

References

[1] "Final committee draft: Editor's proposed revisions," in *Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG*, T. Wiegand, Ed., Feb. 2003, JVT-F100.

[2] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, pp. 560–576, July 2003.

[3] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, pp. 688–703, July 2003.

[4] "Video Coding for Low Bitrate Communication, Version 1," ITU-T, ITU-T Recommendation H.263, 1995.

[5] "Coding of Audio-Visual Objects—Part 2: Visual," ISO/IEC JTC1, ISO/IEC 14 496-2 (MPEG-4 visual version 1), 1999.

[6] "Multimedia Messaging Service (MMS); Media Formats and Codecs," 3GPP Technical Specification 3GPP TR 26.140.

[7] S. Wenger, M. Hannuksela, and T. Stockhammer, "Identified H.26L Applications," ITU-T SG 16, Doc. VCEG-L34, Eibsee, Germany, 2001.

[8] "Codec for Circuit Switched Multimedia Telephony Service; General Description," 3GPP Technical Specification 3GPP TR 26.110.

[9] "Packet Switched Conversational Multimedia Applications; Default Codecs," 3GPP Technical Specification 3GPP TR 26.235.

[10] "Transparent End-to-End Packet Switched Streaming Service (PSS); RTP Usage Model," 3GPP Technical Specification 3GPP TR 26.937.

[11] "Multimedia Messaging Service (MMS); Media Formats and Codecs," 3GPP Technical Specification 3GPP TR 26.140.

[12] "Multimedia Broadcast/Multicast Services," 3GPP Technical Specification 3GPP TR 29.846.

[13] M. Horowitz, A. Joch, F. Kossentini, and A. Hallapuro, "H.264/AVC decoder complexity analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, pp. 704–716, July 2003.

[14] V. Lappalainen, A. Hallapuro, and T. D. Hämäläinen, "Complexity of optimized H.264/AVC video decoder implementation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, pp. 717–725, July 2003.

[15] "High Speed Downlink Packet Access (HSDPA); Overall UTRAN Description," 3GPP Technical Specification 3GPP TR 25.855.

[16] H. Holma and A. Toskala, Eds., *WCDMA for UMTS: Radio Access For Third Generation Mobile Communications*. New York: Wiley, 2000.

[17] *Narrow-Band Visual Telephone Systems and Terminal Equipment, Rev. 4*, ITU-T Recommendation H.320, 1999.

[18] "Generic Coding of Moving Pictures and Associated Audio Information," ISO/IEC International Standard 13 818, 1994.

[19] S. Wenger, T. Stockhammer, and M. M. Hannuksela, "RTP payload format for H.264 video," in *Internet Draft, Work in Progress*, Mar. 2003, Draft-wenger-avt-rtp-h264-01.txt.

[20] "3rd GPP; Technical Specification Group Core Network; IP Multimedia Call Control Protocol Based on SIP and SDP," 3GPP Technical Specification 3GPP TS 24.229.

[21] S. Wenger, "H.264/AVC over IP," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, pp. 645–656, July 2003.

[22] D. Lindberg, "The H.324 multimedia communication standard," *IEEE Commun. Mag.*, vol. 34, pp. 46–51, Dec. 1996.

[23] H. Hannu, L.-E. Jonsson, R. Hakenberg, T. Koren, K. Le, Z. Liu, A. Martensson, A. Miyazaki, K. Svanbro, T. Wiebke, T. Yoshimura, and H. Zheng, "RObust header compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed," in RFC 3095, July 2001.

[24] "Common conditions for video performance evaluation in H.324/M error-prone systems, VCEG (SG16/Q15)," in Ninth Meeting, Redbank, NJ, Oct. 1999, ITU-T Q15-I-60.

[25] G. Roth, R. Sjöberg, G. Liebl, T. Stockhammer, V. Varsa, and M. Karczewicz, "Common Test Conditions for RTP/IP Over 3GPP/3GPP2,", Austin, TX, ITU-T SG16 Doc. VCEG-M77, 2001.

[26] "Radio Link Control (RLC) Protocol Specification," 3GPP Technical Specification 3GPP TS 25.322.

[27] S. S. Hemami, "Robust image transmission using resynchronizing variable-length codes and error concealment," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 927–939, June 2000.

[28] J. Ribas-Corbera, P. A. Chou, and S. Regunathan, "A generalized hypothetical reference decoder for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, pp. 674–687, July 2003.

[29] H. Jenkac, T. Stockhammer, and G. Kuhn, "Streaming media in variable bit-rate environments," presented at the Packet Video Workshop, Nantes, France, Apr..

[30] B. Girod, M. Kalman, Y. J. Liang, and R. Zhang, "Advances in video channel-adaptive streaming," in ICIP 2002, Rochester, NY, Sept. 2002.

[31] E. G. Steinbach, N. Färber, and B. Girod, "Adaptive play-out for law latency video streaming," presented at the ICIP 2001, Thessaloniki, Greece, Oct. 2001.

[32] P. A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *IEEE Trans. Multimedia*, submitted for publication.

[33] Y. J. Liang and B. Girod, "Rate-distortion optimized low—Latency video streaming using channel-adaptive bitstream assembly," presented at the ICME2002, Lausanne, Switzerland, Aug. 2002.

[34] H. Radha, M. van der Schaar, and Y. Chen, "The MPEG-4 fine-grained scalable video coding method for multimedia streaming over IP," *IEEE Trans. Multimedia*, vol. 3, pp. 53–68, Mar. 2001.

[35] M. Karczewicz and R. Kurçeren, "The SP and SI frames design for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, pp. 637–644, July 2003.

[36] V. Varsa, M. M. Hannuksela, and Y. Wang, "Non-Normative Error Concealment Algorithms," ITU-T VCEG-N62, 2001.

[37] S. Wenger and G. Cotê, "Using RFC 2429 and H.263+ at low to medium bit-rates for low-latency applications," presented at the Proc. Packet Video Workshop, New York, NY, Apr. 1999.

[38] V. Varsa and M. Karczewicz, "Slice interleaving in compressed video packetization," presented at the Packet Video Workshop 2000, Forte Village, Italy, May 2000.

[39] S. Wenger and M. Horowitz, "Flexible MB ordering—A new error resilience tool for IP-based video," presented at the IWDC 2002, Capri, Italy, Sept. 2002.

[40] J. Rosenberg and H. Schulzrine, "An RTP payload format for generic forward error correction," in RFC 2733, Dec. 1999.

[41] G. Carle and E. W. Biersack, "Survey of error recovery techniques for IP-based audio-visual multicast applications," *IEEE Network Mag.*, vol. 11, pp. 2–14, Nov. 1997.

[42] U. Horn, K. Stuhlmüller, M. Link, and B. Girod, "Robust internet video transmission based on scalable coding and unequal error protection," *IEEE Trans. Image Processing*, vol. 15, pp. 77–94, Sept. 1999.

[43] Q. F. Zhu and L. Kerofsky, "Joint source coding, transport processing, and error concealment for H.323-based packet video," *Proc. SPIE VCIP*, vol. 3653, pp. 52–62, Jan. 1999.

[44] P. Haskell and D. Messerschmitt, "Resynchronization of motion-compensated video affected by ATM cell loss," *Proc. IEEE ICASSP*, vol. 3, pp. 545–548, 1992.

[45] J. Liao and J. Villasenor, "Adaptive intra update for video coding over noisy channels," *Proc. ICIP*, vol. 3, pp. 763–766, Oct. 1996.

[46] S. Wenger, G. Knorr, J. Ott, and F. Kossentini, "Error resilience support in H.263+," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 867–877, Nov. 1998.

[47] P. Salama, N. B. Shroff, and E. J. Delp, "Error concealment in encoded video," in *Image Recovery Techniques for Image Compression Applications*. Norwell, MA: Kluwer, 1998.

[48] W. M. Lam, A. R. Reibman, and B. Liu, "Recovery of lost or erroneously received motion vectors," in *Proc. ICASSP*, vol. 5, Mar. 1993, pp. 417–420.

[49] V. Varsa, M. M. Hannuksela, and Y.-K. Wang, "Non-Normative Error Concealment Algorithms," ITU-T VCEG-N62, 2001.

[50] Y.-K. Wang, M. M. Hannuksela, V. Varsa, A. Hourunranta, and M. Gabbouj, "The error concealment feature in the H.26L test model," in *Proc. ICIP*, vol. 2, Sept. 2002, pp. 729–732.

[51] T. Wiegand, M. Lightstone, D. Mukherjee, T. G. Campbell, and S. K. Mitra, "Rate-distortion optimized mode selection for very low bit rate video coding and the emerging H.263 standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 182–190, Apr. 1996.

[52] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Mag.*, vol. 15, pp. 74–90, Nov. 1998.

[53] T. Wiegand and B. Girod, "Lagrangian multiplier selection in hybrid video coder control," presented at the Proc. ICIP 2001, Oct. 2001.

[54] T. Stockhammer, D. Kontopodis, and T. Wiegand, "Rate-distortion optimization for H.26L video coding in packet loss environment," presented at the Packet Video Workshop 2002, Pittsburgh, PA, Apr. 2002.

[55] G. Cote, S. Shirani, and F. Kossentini, "Optimal mode selection and synchronization for robust video communications over error-prone networks," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 952–965, Dec. 2000.

[56] T. Wiegand, N. Färber, K. Stuhlmüller, and B. Girod, "Error-resilient video transmission using long-term memory motion-compensated prediction," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 1050-–1050-, Dec. 2000.

[57] C. W. Kim, D. W. Kang, and I. S. Kwang, "High-complexity mode decision for error prone environment," in JVT-C101, May 2002.

[58] R. Zhang, S. L. Regunathan, and K. Rose, "Video coding with optimal inter/intra-mode switching for packet loss resilience," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 966–976, 2000.

[59] Y.-K. Wang and M. M. Hannuksela, "Error-robust video coding using isolated regions," in JVT-C073, May 2002.

[60] Y.-K. Wang, M. M. Hannuksela, and M. Gabbouj, "Error-robust inter/intra mode selection using isolated regions," in Proc. Int. Packet Video Workshop 2003, Nantes, France, Apr. 2003.

[61] E. Steinbach, N. Färber, and B. Girod, "Standard compatible extension of H.263 for robust video transmission in mobile environments," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 872–881, Dec. 1997.
[62] B. Girod and N. Färber, "Feedback-based error control for mobile video transmission," *Proc. IEEE*, vol. 97, pp. 1707–1723, Oct. 1999.
[63] W. Wada, "Selective recovery of video packet losses using error concealment," *IEEE J. Select. Areas Commun.*, vol. 7, pp. 807–814, June 1989.
[64] Y. Wang and Q. Zhu, "Error control and concealment for video communication: A review," *Proc. IEEE*, vol. 86, pp. 974–997, May 1998.
[65] T. Nakai and Y. Tomita, "Core Experiments on Feedback Channel Operation for H.263+," ITU-T SG15 LBC96-308, 1996.
[66] S. Fukunaga, T. Nakai, and H. Inoue, "Error resilient video coding by dynamic replacing of reference pictures," in *Proc. IEEE GLOBECOM*, vol. 3, Nov. 1996, pp. 1503–1508.
[67] Y. Tomita, T. Kimura, and T. Ichikawa, "Error resilient modified inter-frame coding system for limited reference picture memories," presented at the Picture Coding Symposium, Berlin, Germany, Sept. 1997.
[68] "Video Coding for Low Bit-Rate Communication, Version 1," ITU-T Recommendation H.263, 1995.
[69] T. Stockhammer, T. Wiegand, T. Oelbaum, and F. Obermeier, "Video coding and transport layer techniques for H.264-based transmission over packet-lossy networks," presented at the ICIP, Barcelona, Spain, Sept. 2003.
[70] T. Stockhammer and S. Wenger, "Standard-compliant enhancement of JVT coded video for transmission over fixed and wireless IP," presented at the IWDC 2002, Capri, Italy, Sept. 2002.

Thomas Stockhammer received the Diplom.-Ing. degree in electrical engineering in 1996 from the Munich University of Technology (TUM), Munich, Germany, where he is currently working toward the Dr.-Ing. degree in the area of source and video transmission over mobile and packet-lossy channels.

In 1996, he visited Rensselear Polytechnic Institute (RPI), Troy, NY to perform his diploma thesis in the area of combined source channel coding for video and coding theory. There he began research in video transmission and combined source and channel coding. In 2000, he was Visiting Researcher in the Information Coding Laboratory, University of San Diego at California (UCSD). Since then, he has published several conference and journal papers and holds several patents. He regularly participates and contributes to different standardization activities, e.g., ITU-T H.324, H.264, ISO/IEC MPEG, JVT, and IETF. He acts as a member of several technical program committees, as Reviewer for different journals, and as an Evaluator for the European Commission. His research interests include joint source and channel coding, video transmission, system design, rate-distortion optimization, information theory, and mobile communications.

Miska M. Hannuksela received the M.S. degree in engineering from Tampere University of Technology, Tampere, Finland, in 1997.

He is currently a Research Manager in the Visual Communications Laboratory, Nokia Research Center, Tampere, Finland. From 1996 to 1999, he was a Research Engineer with Nokia Research Center in the area of mobile video communications. From 2000 to 2003, he was a Project Team Leader and a specialist in various mobile multimedia research and product projects in Nokia Mobile Phones. He has been an active participant in the ITU-T Video Coding Experts Group since 1999 and in the Joint Video Team of ITU-T and ISO/IEC since its foundation in 2001. He has co-authored more than 80 technical contributions to these standardization groups. His research interests include video error resilience, scalable video coding, and video communication systems.

Thomas Wiegand received the Dr.-Ing. degree from the University of Erlangen-Nuremberg, Erlangen-Nuremberg, Germany, in 2000 and the Dipl.-Ing. degree in electrical engineering from the Technical University of Hamburg-Harburg, Hamburg-Harburg, Germany, in 1995.

He is the Head of the Image Communication Group in the Image Processing Department, Fraunhofer–Institute for Telecommunications – Heinrich Hertz Institute (HHI), Berlin, Germany. During 1997 to 1998, he was a Visiting Researcher at Stanford University, Stanford, CA, and served as a Consultant to 8x8, Inc., Santa Clara, CA. From 1993 to 1994, he was a Visiting Researcher at Kobe University, Kobe, Japan. In 1995, he was a Visiting Scholar at the University of California at Santa Barbara, where he began his research on video compression and transmission. Since then, he has published several conference and journal papers on the subject and has contributed successfully to the ITU-T Video Coding Experts Group (ITU-T SG16 Q.6—VCEG)/ISO/IEC Moving Pictures Experts Group (ISO/IEC JTC1/SC29/WG11—MPEG)/Joint Video Team (JVT) standardization efforts and holds various international patents in this field. He has been appointed as the Associated Rapporteur of the ITU-T VCEG (October 2000), the Associated Rapporteur/Co-Chair of the JVT that has been created by ITU-T VCEG and ISO/IEC MPEG for finalization of the H.264/AVC video coding standard (December 2001), and the Editor of the H.264/AVC video coding standard (February 2002).

[P4] Y.-K. Wang, M. M. Hannuksela, K. Caglar, and M. Gabbouj, “Improved error concealment using scene information,” *Proceedings of the International Workshop VLBV03*, published as *Lecture Notes in Computer Science*, vol. 2849/2003, pp. 283-289, Springer, Sep. 2003.

[P5] M. M. Hannuksela, Y.-K. Wang, and M. Gabbouj, “Isolated regions in video coding,” *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 259-267, Apr. 2004.

Isolated Regions in Video Coding

Miska M. Hannuksela, *Member, IEEE*, Ye-Kui Wang, *Member, IEEE*, and Moncef Gabbouj, *Senior Member, IEEE*

***Abstract*—Different types of prediction are applied in modern video coding. While predictive coding improves compression efficiency, the propagation of transmission errors becomes more likely. In addition, predictive coding brings difficulties to other aspects of video coding, including random access, parallel processing, and scalability. In order to combat the negative effects, video coding schemes introduce mechanisms, such as slices and intracoding, to limit and break the prediction. This paper proposes the use of the isolated regions coding tool that jointly limits in-picture prediction and interprediction on region-of-interest basis. The tool can be used to provide random access points from non-intrapictures and to respond to intrapicture update requests. Furthermore, it can be applied as an error-robust macroblock mode decision method and can be used in combination with unequal error protection. Finally, it enables mixing of scenes, which is useful in coding of masked scene transitions.**

***Index Terms*—Error resilience, isolated regions, random access, video coding.**

I. Introduction

CURRENT video coding standards include ITU-T H.261, ITU-T H.263, ISO/IEC MPEG-1 Part 2, ISO/IEC MPEG-2 Part 2 (a.k.a. ITU-T H.262), and ISO/IEC MPEG-4 Part 2. These standards are based on block-based translational motion compensation and discrete cosine transform (DCT) based residual coding and are herein referred to as conventional video coding standards. The Joint Video Team (JVT) of ITU-T and ISO/IEC recently finalized a new standard based on an earlier ITU-T standardization project called H.26L. The resulting standard is called ITU-T Recommendation H.264 or ISO/IEC International Standard 14 496-10 (MPEG-4 Part 10) [1] and is referred to as the Advanced Video Coding (AVC) standard in this paper.

During transmission, many video communication systems undergo transmission errors. Transmission errors can be categorized into bit errors and packet errors. Bit errors are typically caused by imperfections of physical channels, such as radio interference; while, packet errors are typically due to elements in packet-switched networks. For example, a packet router may become congested; i.e., it may get too many packets as input and cannot output them at the same rate. In this situation, its buffers overflow, and some packets get lost as a result. Packet duplication and packet delivery in different order than transmitted are also possible.

A video communication system includes a transmitter and a receiver. A transmitter includes a source coder and a transport coder. The source coder inputs uncompressed images and outputs coded video stream. The transport coder encapsulates the compressed video according to the transport protocols in use. The receiver performs inverse operations, i.e., transport decoding and source decoding, to obtain a reconstructed video signal. Transmission errors can be controlled in the transport coding layer or in the source coding layer or jointly in both layers. For example, some transport systems enable unequal error protection where part of the transmitted stream is conveyed more reliably than the rest.

Interactive error concealment refers to techniques where the recipient transmits information about corrupted decoded areas and/or transport packets to the transmitter. Many communication systems include a mechanism to convey such feedback information. For example, in ITU-T H.323 and H.324 video conferencing standards, the receiver can request an intra-update of an entire picture or certain macroblocks using the H.245 control protocol. The transmitter typically responds to such a request by coding the requested area in intramode in the next picture to be coded.

Noninteractive error control techniques do not involve interaction between the transmitter and the receiver. Error concealment refers to techniques where the receiver estimates the correct decoded representation of erroneously received data. Forward error control refers to techniques where the transmitter adds such redundant data in the coded stream that helps the receiver conceal transmission errors.

A thorough review of error resilient video coding techniques is given in [2].

Another important aspect in video communication is random access. Random access refers to the ability of the decoder to start decoding a stream at a point other than the beginning of the stream and recover an exact or approximate representation of the decoded pictures. A random access point and a recovery point characterize a random access operation. The random access point is any coded picture where decoding can be initiated. All decoded pictures at or subsequent to a recovery point in output order are correct or approximately correct in content. If the random access point is the same as the recovery point, the random access operation is instantaneous; otherwise, it is gradual.

Random access points enable seek, fast forward, and fast backward operations in locally stored video streams. In video on-demand streaming, servers can respond to seek requests by transmitting data starting from the random access point that is closest to the requested destination of the seek operation.

Manuscript received December 30, 2002; revised August 7, 2003. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Antonio Ortega.

M. M. Hannuksela is with Nokia Research Center, 33721 Tampere, Finland (e-mail: miska.hannuksela@nokia.com).

Y.-K. Wang is with Nokia Mobile Software, 33721 Tampere, Finland (e-mail: ye-kui.wang@nokia.com).

M. Gabbouj is with the Tampere University of Technology, 33101 Tampere, Finland (e-mail: moncef.gabbouj@tut.fi).

Digital Object Identifier 10.1109/TMM.2003.822784

Random access points enable tuning in to a broadcast. In addition, a random access point can be coded as a response to a scene cut in the source sequence or as a response to a fast update intrapicture update request. The proposed isolated regions tool shall prove useful in providing additional random access capability.

This paper is organized as follows. Section II summarizes the types of prediction used in video coding. Applications where prediction needs to be limited or disabled are presented, and a review of methods to limit prediction is given. Section III presents the isolated regions technique, which is based on limiting predictive coding in a specific way. Moreover, the relation of the AVC standard to the isolated region technique is presented in the same section. Section IV demonstrates how isolated regions can be used in random access. Section V applies the isolated regions technique in forward error control; whereas, isolated regions are used in combination with unequal error protection in Section VI. Scene mixing, as presented in Section VII, is yet another application for the isolated regions technique. Finally, Section VIII concludes the paper.

II. Predictive Video Coding

A. Types of Prediction

Video coding is typically a two-stage process. First, a prediction of the video signal is generated based on previous coded data. Second, the residual between the predicted signal and the source signal is coded. Prediction enables efficient compression, but it causes some complications in error-prone environments, in random access, and in parallel decoding. In the following, we categorize the most commonly used types of prediction and in Sections II-B–E, we describe the applications and means for constrained prediction.

Interprediction, which is also referred to as temporal prediction and motion compensation, removes temporal redundancy. In interprediction, the sources of prediction are previously decoded pictures. H.263, MPEG-4 Part 2, and the AVC standard enable storage of multiple reference pictures for interprediction and selection of the used reference picture on picture segment or macroblock basis.

Intraprediction utilizes the fact that adjacent pixels within the same picture are likely to be correlated. Intraprediction can be performed in spatial or transform domain, i.e., either sample values or transform coefficients can be predicted. Intraprediction is typically exploited in intracoding, where no interprediction is applied.

One outcome of the coding procedure is a set of coding parameters, such as motion vectors and quantized transform coefficients. Many parameters can be entropy-coded more efficiently if they are predicted first from spatially or temporally neighboring parameters. For example, a motion vector is typically predicted from spatially adjacent motion vectors. Prediction of coding parameters and intraprediction are collectively referred to as in-picture prediction in this paper.

B. Applications for Constrained Prediction

While prediction brings high compression efficiency, it causes inconveniences in other aspects such as error resiliency, random access, parallel processing, and scalability. To compromise between any of these aspects and compression efficiency, constraining prediction is required.

Error Resiliency: If a piece of coded data is hit by a transmission error, the error is visible not only in the decoded area corresponding to the piece of data, but also in spatially neighboring areas that are predicted from the corrupted area. Moreover, all coding parameters predicted from corrupted parameter values are likely to be incorrect. Furthermore, due to interprediction, the artifacts caused by transmission errors propagate in time. Therefore, constraining prediction in a way that transmission errors are as imperceptible as possible is one of the key features in error-prone video communication systems.

Random Access: Random access refers to the ability to start the decoding at any of the random access points of the stream and recover decoded pictures that are correct in content. Frequent random access points are desirable in many applications. For example, random access points allow new recipients to tune in to a video broadcast, and they allow seeking to a desired position in stored video, such as DVD. In order to code a random access point at a specific picture, typically interprediction has to be broken.

Parallel Processing: Parallel processing refers to the process of encoding/decoding different parts of a picture simultaneously. Parallel processing is a desirable feature in multiprocessor architectures. In practice, parts of a picture being coded simultaneously have to be independent, i.e., no prediction from one part to another is allowed.

Scalability: Scalability refers to the capability of a compressed sequence to be decoded at different bit-rates. In scalable video coding prediction is limited in a way that certain parts of the compressed sequence, such as an enhancement layer in layered scalability or a B picture in conventional video coding standards, can be ignored in the decoding process without affecting the decoding of the rest of the compressed sequence. Scalable coded sequences can be used for many purposes. For example, a streaming server may adjust the bit-rate of a prestored coded sequence according to the prevailing network conditions.

C. Means to Limit In-Picture Prediction

Video coding standards allow dividing a coded picture to coded segments or slices. In-picture prediction is typically disabled across slice boundaries. Thus, slices can be regarded as a way to split a coded picture to independently decodable pieces. Coded segments can be categorized into three classes: raster-scan-order slices, rectangular slices, and flexible slices.

A raster-scan-order-slice is a coded segment that consists of consecutive macroblocks in raster scan order. Video packets of MPEG-4 Part 2 and groups of macroblocks (GOBs) starting with a nonempty GOB header in H.263 are examples of raster-scan-order slices.

A rectangular slice is a coded segment that consists of a rectangular area of macroblocks. A rectangular slice may be higher than one macroblock row and narrower than the entire picture width. H.263 includes an optional rectangular slice submode, and H.261 GOBs can also be considered as rectangular slices.

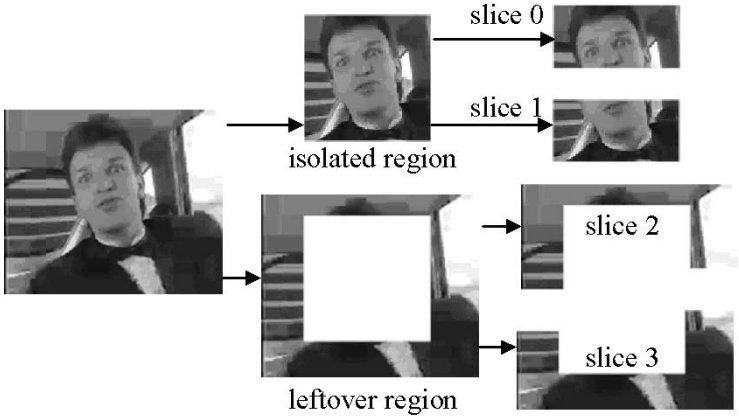


Fig. 1. Example partitioning of a picture to an isolated region and a leftover region and further to slices.

A flexible slice can contain any predefined macroblock locations. The AVC codec allows grouping of macroblocks to more than one slice groups. A slice group can contain any macroblock locations, including nonadjacent macroblock locations. A slice consists of at least one macroblock within a particular slice group in raster scan order.

D. Means to Limit Interprediction

Intracoding of pictures and macroblocks is one way to break interprediction. Reference picture selection can be used to make the chains of interpictures shorter. In addition, interprediction can be limited by restricting the values of motion vectors. A brief review of reference picture selection based methods limiting interprediction has been given in [3].

E. Types and Limitation of Prediction in the AVC Codec

The syntax of a coded AVC sequence consists of Network Abstraction Layer (NAL) units. A NAL unit is an atomic element that can be framed for transport and parsed independently. Each NAL unit has a specific type, which can be a coded slice, a coded data partition, a sequence parameter set, a picture parameter set, or a supplemental enhancement information (SEI) message among other things. The parameter set concept [4] replaces the use of sequence and picture headers. In contrast to redundant coding of sequence and picture headers for improved error resiliency, the AVC codec enables transmission of sequence and picture parameter sets externally from the rest of a coded sequence using another, more reliable transmission channel or protocol.

Some coding parameters in a NAL unit of one type depend on coding parameters of a NAL unit of another type. In particular, the following dependency hierarchy relates to coded slices: A coded slice consists of a slice header and slice data. A slice header refers to a picture parameter set, and a picture parameter set refers to a sequence parameter set. A picture parameter set contains parameters whose values remain unchanged within a coded picture, whereas the parameters in a sequence parameter set remain unchanged during an entire coded video sequence.

A coded picture consists of at least one coded slice. Coded parameters are not predicted across slice boundaries. Many parameter values of a slice are adaptively predicted from earlier coding parameters of the same slice.

The AVC codec includes a number of directional pixel-domain intraprediction modes for 4×4 or 16×16 blocks. The border pixels of the neighboring blocks above and on the left are used as prediction sources. A block is not used as a source for intraprediction if it belongs to a different slice than the block to be coded or decoded. The picture parameter set contains a constrained intraprediction flag that is used to control whether neighboring non-intracoded blocks are used for intraprediction.

Interprediction is based on translational motion of blocks. Motion vectors have the accuracy of 1/4 luma samples. Fractional pixels are interpolated using a two-stage filtering process including a 6-tap and a 2-tap filter. Interprediction can be limited by selecting reference pictures for prediction carefully. Moreover, a particular type of an intrapicture, called an instantaneous decoding refresh (IDR) picture, has been specified. No subsequent picture can refer to pictures that are earlier than the IDR picture in decoding order. Thus each IDR picture forms a random access point.

III. Isolated Regions

A. Fundamentals of Isolated Regions

The proposed technique isolated regions is based on constraining in-picture prediction and interprediction jointly.

An isolated region in a picture can contain any macroblock locations, and a picture can contain zero or more isolated regions that do not overlap. A leftover region is the area of the picture that is not covered by any isolated region of a picture. When coding an isolated region, in-picture prediction is disabled across its boundaries. A leftover region may be predicted from isolated regions of the same picture.

A coded isolated region can be decoded without the presence of any other isolated or leftover region of the same coded picture. It may be necessary to decode all isolated regions of a picture before the leftover region. An isolated region or a leftover region contains at least one slice. Fig. 1 presents an example where the picture contains one isolated region and a leftover region. Both the isolated region and the leftover region contain two slices.

Pictures, whose isolated regions are predicted from each other, are grouped into an isolated-region picture group. An isolated region can be interpredicted from the corresponding isolated region in other pictures within the same isolated-region picture group, whereas interprediction from other isolated regions or outside the isolated-region picture group is disallowed. A leftover region may be interpredicted from any isolated region. The shape, location, and size of coupled isolated regions may evolve from picture to picture in an isolated-region picture group.

B. Coding of Isolated Regions in the AVC Codec

Coding of isolated regions in the AVC codec is based on slice groups introduced in Section II-C. The mapping of macroblock locations to slice groups is specified in the picture parameter set. The AVC syntax includes efficient methods to code certain slice group patterns, which can be categorized into two types, static and evolving. The static slice groups stay unchanged as long as the picture parameter set is valid, whereas the evolving slice groups can change picture by picture according to the corresponding parameters in the picture parameter set and a slice group change cycle parameter in the slice header. The static slice group patterns include interleaved, checkerboard, rectangular oriented, and freeform. The evolving slice group patterns include horizontal wipe, vertical wipe, box-in, and box-out. The rectangular oriented pattern and the evolving patterns are especially suited for coding of isolated regions and are described more carefully in the following.

For a rectangular oriented slice group pattern, a desired number of rectangles are specified within the picture area. A foreground slice group includes the macroblock locations that are within the corresponding rectangle but excludes the macroblock locations that are already allocated by slice groups specified earlier. A leftover slice group contains the macroblocks that are not covered by the foreground slice groups. The left-hand side picture in Fig. 2 includes two rectangular foreground slice groups (indicated by a white rectangle) and the righthand side picture in Fig. 2 includes three foreground slice groups, two of which are rectangular and the third one, i.e., the screen behind the newsreaders, is composed by excluding the first two rectangles from a bounding rectangle.

An evolving slice group is specified by indicating the scan order of macroblock locations and the change rate of the size of the slice group in number of macroblocks per picture. Each coded picture is associated with a slice group change cycle parameter (conveyed in the slice header). The change cycle multiplied by the change rate indicates the number of macroblocks in the first slice group. The second slice group contains the rest of the macroblock locations. Fig. 3 shows an example of the first five change cycles of the first slice group of the box-out type with a change rate of 12 macroblocks.

In-picture prediction is always disabled across slice group boundaries, because slice group boundaries lie in slice boundaries. Therefore, each slice group is an isolated region or leftover region.

Each slice group has a unique identification number within a picture. Encoders can restrict the motion vectors in a way that they only refer to the decoded macroblocks belonging to slice groups having the same identification number as the slice group to be encoded. Encoders should take into account the fact that a range of source samples is needed in fractional pixel interpolation and all the source samples should be within a particular slice group.

Fig. 2. Examples of rectangular oriented isolated regions.

Fig. 3. Example of an evolving isolated region.

The AVC codec includes a deblocking loop filter. Loop filtering is applied to each 4×4 block boundary, but loop filtering can be turned off at slice boundaries. If loop filtering is turned off at slice boundaries, perfect reconstructed pictures can be achieved when performing gradual random access. Otherwise, reconstructed pictures would be imperfect in content even after the recovery point. However, in many applications the mismatch is unperceivable and the picture quality is acceptable without turning off the loop filtering at slice boundaries.

The recovery point SEI message and the motion constrained slice group set SEI message of the AVC standard can be used to indicate that some slice groups are coded as isolated regions with restricted motion vectors. The decoder may utilize the information to achieve faster random access or to save in processing time by ignoring the leftover region.

C. Comparison to Earlier Techniques for Joint In-Picture and Interprediction Limitation

As far as the authors are aware, the closest predecessor of the isolated regions technique is the optional independent segment decoding mode of H.263 (H.263, Annex R). When this optional mode is in use, all slices have to be rectangular. Slice boundaries are treated as picture boundaries, and therefore no spatio–temporal error propagation over slice boundaries occurs. Due to restricted motion prediction, compression efficiency drops compared to normal slice-based operation. The locations of slice boundaries have to remain unchanged within a group of pictures (GOP). This fact hinders the use of the independent segment decoding mode for many of the applications presented in this paper. Furthermore, because the number of macroblocks in a slice is constant within a GOP, the encoder has few means to control the coded size of a slice in bytes. This fact may make the encapsulation of slices to transport packets nonoptimal, because the slice size cannot be adjusted according to an optimal packet size according to prevailing network conditions.

In many applications, such as the case presented in Section VI-C, one rectangular isolated region is sufficient. If

such a scheme were coded with H.263 rectangular slices, five rectangular slices would be needed in contrast to one isolated region and one leftover region. Consequently, both in-picture and interprediction falling into the area of the leftover region would be disallowed unnecessarily across the boundaries of the rectangular slices.

IV. Random Access

A. Gradual Decoding Refresh

Conventionally each intrapicture has been a random access point in a coded sequence. The introduction of multiple reference pictures for interprediction caused that an intrapicture may not be sufficient for random access. For example, a decoded picture before an intrapicture in decoding order may be used as a reference picture for interprediction after the intrapicture in decoding order. Therefore, an IDR picture as specified in the AVC standard or an intrapicture having similar properties to an IDR picture has to be used as a random access point. In this section term IDR picture is not exclusively specific to the AVC standard.

Gradual decoding refresh (GDR) refers to the ability to start the decoding at a non-IDR picture and recover decoded pictures that are correct in content after decoding a certain amount of pictures. That is, GDR can be used to achieve random access from non-intrapictures. Some reference pictures for interprediction may not be available between the random access point and the recovery point, and therefore some parts of decoded pictures in the gradual decoding refresh period cannot be reconstructed correctly. However, these parts are not used for prediction at or after the recovery point, which results into error-free decoded pictures starting from the recovery point.

It is obvious that gradual decoding refresh is more cumbersome both for encoders and decoders compared to instantaneous decoding refresh. However, gradual decoding refresh is desirable in error-prone environments thanks to two facts: First, a coded intrapicture is generally considerably larger than a coded non-intrapicture. This makes intrapictures more susceptible to errors than non-intrapictures, and the errors are likely to propagate in time until the corrupted macroblock locations are intracoded. Second, intracoded macroblocks are used in error-prone environments to stop error propagation (see Section V-A for more details). Thus, it makes sense to combine the intramacroblock coding for random access and for error propagation prevention, for example, in video conferencing and broadcast video applications that operate on error-prone transmission channels. This conclusion is utilized in gradual decoding refresh.

An evolving isolated region can be used to provide gradual decoding refresh. A new evolving isolated region is established in the picture at the random access point, and the macroblocks in the isolated region are intracoded. The shape, size, and location of the isolated region evolve from picture to picture. The isolated region can be interpredicted from the corresponding isolated region in earlier pictures in the gradual decoding refresh period. When the isolated region covers the whole picture area, a picture completely correct in content is obtained when decoding started from the random access point. This process can also be generalized to include more than one evolving isolated region that eventually cover the entire picture area.

There may be tailored in-band signaling, such as the recovery point SEI message of the AVC standard, to indicate the gradual random access point and the recovery point for the decoder. Furthermore, the recovery point SEI message includes an indication whether an evolving isolated region is used between the random access point and the recovery point to provide gradual decoding refresh.

Gradual decoding refresh using isolated regions can also be applied as a response to intrapicture update request. In applications with a feedback channel, a receiving terminal may request the far-end encoder for an intrapicture refresh if the received pictures are too corrupted. There is another use of an intrapicture refresh request in multipoint video conferencing, in which the multipoint control unit orchestrates a switch of source sequences delivered to recipients by issuing an intrapicture refresh request to a desired source terminal. Conventionally, an encoder responds to an intrapicture refresh request by coding and transmitting an intracoded picture. Due to avoiding of intrapicture coding, improved error resiliency can be achieved by using isolated regions.

B. Simulations

Two sets of simulations were done using the AVC codec.

1) *Coding efficiency simulations*. Gradual decoding refresh based on isolated regions was compared to periodic IDR picture coding at a 1-s random access period. Error-free application environment, such as local storage, was assumed, and therefore the coding options yielding the best coding efficiency were selected. The simulations abided the coding efficiency simulation common conditions specified by ITU-T Video Coding Experts Group [5]. A number of QCIF and CIF sequences were coded, and the average bitrate loss of gradual decoding refresh compared to periodic IDR was between 11% and 17%. More results can be obtained from [6].
2) Error resiliency simulations. The error resiliency performance of gradual decoding refresh was compared with the periodic IDR picture coding. The target was to simulate IP multicast streaming where random access points allow new receivers to start decoding. Random access period of about 1 second was used. Packet loss simulations under loss rates of 0, 3, 5, 10, and 20% were performed according to the conditions specified by ITU-T Video Coding Experts Group [7] with minor modifications as listed in [6]. One set of results is presented in Fig. 4 and more results can be obtained from [6]. It can be seen that gradual decoding refresh performs consistently better compared to periodic IDR in all loss rates. Moreover, the PSNR difference between the cases grows as a function of loss rate. From the simulation results, it can also be seen that using gradual decoding refresh based on isolated regions to respond intraupdate requests has better error resiliency performance than coding intrapictures.

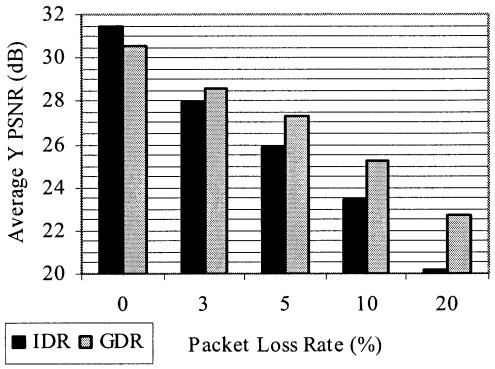


Fig. 4. Comparison of periodic IDR and GDR in terms of average luminance PSNR at different packet loss rates. Sequence: Paris at 384 kbits/s.

V. Error-Robust Inter/Intra-Mode Decision

A. Error-Robust Macroblock Mode Decision

Video encoders have numerous ways to reduce the spatial and temporal propagation of transmission errors and to help decoders concealing transmission errors. One of these methods is to stop temporal error propagation by intramacroblock coding. In applications, where the content is encoded before transmission (e.g., on-demand streaming) or where no feedback about the error or loss locations from the recipients is possible (e.g., live multicast with a huge number of receivers), the encoder has to conclude the rate and locations of intramacroblocks based on expected or measured transmission error or loss rate.

The macroblock mode selection algorithms can be categorized into nonadaptive and adaptive algorithms, and adaptive methods can be further classified to cost-function-based and rate-distortion optimized ones. The family of nonadaptive intrarefresh algorithms includes the circular intrarefresh algorithm that scans the picture area in a predefined order and codes a certain number of intramacroblocks per picture in the predefined scan order. Another example of a nonadaptive algorithm is to code a certain number of macroblocks in intramode at randomly selected macroblock locations. Adaptive macroblock mode decision methods select the intracoded macroblock locations in a way that the content of the pictures is taken into account. For example, a static background area needs not be refreshed in intramode as often as moving objects. Cost-function-based methods, such as [8] and [9], calculate a cost for each macroblock with a certain function that may take into account the amount of prediction error data after motion compensation, for example. A certain number of macroblocks having the highest cost are coded in intramode. Rate-distortion optimized macroblock mode selection algorithms use a Lagrangian cost function that linearly combines terms "rate" and "distortion." The mode selection of each macroblock is such that the cost is minimized. An estimate of the expected distortion caused by transmission errors and losses is taken into account in the cost function. A number of distortion estimation algorithms have been proposed and one of them, herein referred to as the loss-aware rate-distortion-optimized (LA-RDO) macroblock mode selection algorithm, has been selected into the reference implementation of the AVC codec [10]. The computational complexity of rate-distortion optimized macroblock mode selection algorithms is typically multifold compared to nonadaptive and cost-function-based algorithms.

B. Isolated Regions in Macroblock Mode Decision

Evolving isolated regions can be used as a nonadaptive macroblock mode selection algorithm. A new evolving isolated region is established at the beginning of an intrarefresh period, i.e., the period of the isolated-region picture group. The intrarefresh period is completed when the isolated region covers the entire picture area. The macroblocks in the isolated region of the first picture in the intrarefresh period are intracoded. The newly added macroblocks in the isolated region of later pictures are intracoded, whereas the other macroblocks in the isolated region can be interpredicted from the corresponding isolated region within the same intrarefresh period.

If the above algorithm has an adaptive change rate for isolated regions or the following modification is applied, the algorithm falls into the category of adaptive macroblock mode selection algorithms: In contrast to coding newly added macroblocks in intramode, the encoder can apply a normal macroblock mode selection algorithm for them. As a result, the newly added macroblocks may be interpredicted from the corresponding isolated region in the same isolated-region picture group or they may be intracoded.

The encoder can select a proper change rate of the isolated region according to the picture size and the assumed transmission error rate. Generally, a good change rate is equivalent to the expected loss rate of macroblocks. For example, for a CIF sequence, if the packet loss rate is 20%, a change rate of about 80 macroblocks per picture is appropriate. However, due to the possible large differences in sequence characteristics and different coding options, a content-adaptive change rate may perform better and is under investigation.

C. Simulations

Four intrarefresh algorithms were compared: conventional circular intrarefresh at a rate of one macroblock row per picture (CIR), the loss-aware rate-distortion-optimized macroblock mode selection of the AVC reference codec (LA-RDO), isolated regions based circular intrarefresh (IREG-CIR), and a combination of LA-RDO and IREG-CIR. Real-time multicast/broadcast to users with different network conditions was assumed. Therefore, the coding options were selected in a way that the strongest error resiliency performance suitable for the worst expected network condition, 20% packet loss rate, was targeted. The coded bitstreams were decoded after packet loss simulation under different loss rates 0, 3, 5, 10, and 20%. Six coded sequences for each intrarefresh algorithm were generated: Foreman QCIF at 64 kbits/s, Foreman QCIF at 144 kbits/s, Hall Monitor QCIF at 32 kbits/s, Irene CIF at 384 kbits/s, Paris CIF at 144 kbits/s, and Paris CIF at 384 kbits/s, referred herein to as sequences 1 to 6, respectively. More details on the simulation conditions can be obtained from [11].

Fig. 5 presents the average luma PSNR of all the test sequences for each intrarefresh algorithm and each packet loss rate. The simulation results show that the difference in average luma PSNR between IREG-CIR and LA-RDO is

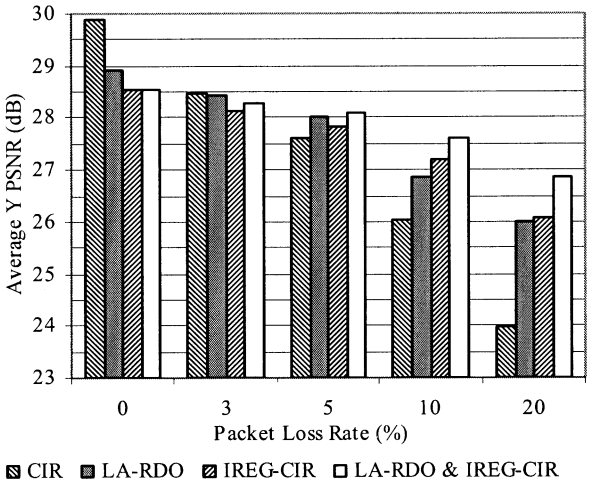


Fig. 5. Comparison of macroblock mode selection algorithms at different packet loss rates. Vertical axis indicates the average luma PSNR of all the test sequences.

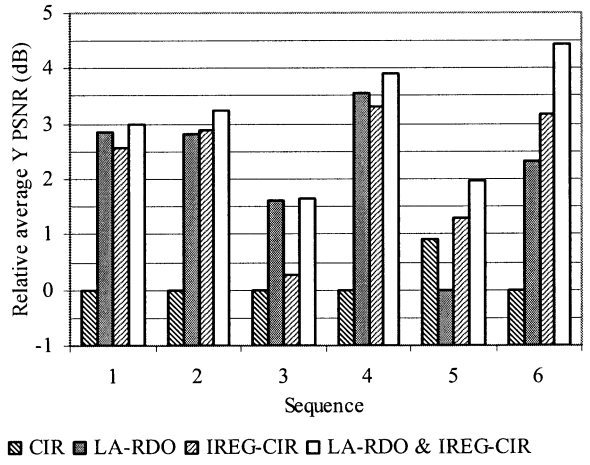


Fig. 6. Comparison of macroblock mode selection algorithms at 20% packet loss rate. Vertical axis indicates the average luma PSNR for a particular sequence and algorithm relative to the worst average luma PSNR for that sequence.

within 0.5 dB regardless of the packet loss rate. In packet loss rates greater than or equal to 5%, the combination of LA-RDO and IREG-CIR outperforms other algorithms, the difference being more than 0.5 dB in the 20% packet loss rate case, to which the bitstreams were optimized. Fig. 6 shows the average luma PSNR for each test sequence in the 20% packet loss rate case. It can be observed that the combination of LA-RDO and IREG-CIR outperforms other algorithms consistently. More detailed simulation results are available in [11].

VI. Unequal Error Protection

A. Conventional Coding Tools for Unequal Error Protection

In order to apply unequal error protection, coded video sequences have to be organized in portions of different importance in terms of visual quality. Techniques achieving this goal include data partitioning, scalable coding, and object-based coding.

Data partitioning refers to a technique where subjectively equally important codewords of all macroblocks in a slice are partitioned into a continuous block of data. Typically, macroblock headers and motion information form one partition and coded prediction error blocks form another partition.

Data partitioning and scalable coding techniques generally treat an entire image equally in spatial domain. However, many images have distinct spatial regions of interest. These regions could have better error protection than other areas in order to obtain a better subjective quality compared to coding and transport schemes that treat all regions equally. Arbitrarily shaped objects [12], as defined in the MPEG-4 Part 2, can be used to extract the regions of interest. However, its high complexity limits its use in real-time encoding.

B. Isolated Regions for Unequal Error Protection

Isolated regions can be used for unequal error protection. The encoder first selects at least one region of interest from the first picture to be encoded using face detection or image analysis techniques, for example. Each region of interest is an isolated region, and the rest of the macroblocks form the leftover region. In the next picture to be encoded, the encoder tracks the same regions of interest as in the previous picture. Each region of interest is coded as an isolated region that is interpredicted only from the corresponding isolated region in the previous reference pictures.

The isolated regions technique allows partitioning pictures spatially and temporally to regions of interest. Each coded isolated region can be further divided into slices and data partitions. Furthermore, the quality of an isolated region can be improved in an enhancement layer, whereas the layer may not provide any quality improvement to the leftover region. Thus, isolated region coding complements data partitioning and scalable coding, and it is an alternative to object-based coding.

C. Simulations

We selected multicast Internet streaming as a target application. A constant rectangular region of interest was selected for each sequence, and smaller quantization steps were used within the region of interest. In one set of sequences the region of interest was coded as an isolated region, and another set of sequences was coded conventionally. The scheme was compared to the conventional codec (version TML8.6 of the AVC public reference software [13]) with and without region-of-interest quantization (abbreviated as Conv + ROI and Conv, respectively). The selection of the quantization step size based on the region of interest was the same in the proposed coding scheme and the Conv + ROI coding scheme.

As interactive error concealment cannot be used in large scale with IP multicast, transport coding level forward error correction (FEC) according to RFC 2733 [14] was used. To be more detailed, we used the so-called parity FEC, where one FEC packet is associated with two media packets and is able to correct the loss of either media packet. Other FEC strengths were not experimented, because we targeted to minimize the delay associated with FEC coding and decoding.

Encapsulation into RTP packets was done as follows. In the proposed coding scheme, intrapictures were encapsulated into five packets. There were two packets for the isolated region: one packet contained odd macroblock rows and another packet

Fig. 7. Results of the unequal error protection simulations: example snapshots of 20% packet loss rate. From left to right, the used codecs are Conv, Conv + ROI, and the proposed codec.

contained even macroblock rows. This slice interleaving mechanism, introduced in [15], was used to obtain a better error concealment result. One parity FEC packet was generated for the two foreground packets according to RFC 2733. The leftover region was packetized into another two packets using slice interleaving method. Two consecutive interpictures consisted a group, and for each such group there were two isolated region packets, one parity FEC packet for the isolated region packets, and two leftover region packets. An isolated region packet contained data from two pictures: macroblocks from even rows of a certain frame and macroblocks from odd rows of the next frame or vice versa. When subpicture coding was not in use, there were three packets for each intra- and interframe: two packets for the entire picture (slice interleaving applied), and one parity FEC packet for the two packets.

Intramacroblock refresh was tailored for the worst expected case (20% packet loss rate), and packet losses were simulated with the obtained packet stream at 0, 3, 5, 10, and 20% packet loss rates. See [16] and [17] for further details on the simulation conditions.

The experiments were done using the Carphone, Hall, Coastguard, Foreman, News, and Irene sequences, with different frame rates and bit-rates. We present only part of the results due to lack of space, more results can be obtained from [16] and [17].

Fig. 7 shows some example snapshots of Foreman at 64 kbps and Carphone at 64 kbps in 20% packet loss rate. It can be seen that in both sequences the proposed subpicture coding scheme with gradual bit allocation maintains the best subjective image quality. In fact, the overall PSNR in the proposed coding drops a little compared to conventional coding cases. However, since errors in the background are far less noticeable than errors in the foreground, the overall subjective quality is improved.

VII. Scene Mixing

A. Applications

There are a couple of situations where mixing of multiple source pictures into the same coded picture, termed as scene mixing herein, is necessary. The cases can be roughly categorized into masked scene transitions and constant scene mixing. Masked scene transitions are such that one scene spatially uncovers from the other scene or from black in a gradual manner, and all pictures are mixed and displayed at full intensity. Examples of constant scene mixing include the so-called picture-in-picture scheme, where a picture from one source is included in the picture area originating from another source. For instance, a news broadcast may include a newsreader and a small screen besides her showing video material of a news topic. Furthermore, in video conferencing or surveillance, pictures from multiple cameras may have to be tiled to the same coded picture.

B. Problems

Conventionally, scene mixing is done as follows. First, source pictures are composed from the original pictures of different scenes. Then, the source pictures are coded as if they were normal pictures. The conventional coding approach is not optimal at least due to the following reasons.

- Boundaries of slices do not follow the original source picture boundaries. Thus, in-picture prediction is not likely to succeed well if the source for prediction is from a different scene than the block to be coded.
- It is likely that there is a sharp edge between the original source pictures. If a loop filter is applied, it smoothes the edge unnecessarily.

C. Scene Mixing Based on Isolated Regions

A masked scene transition can be coded with an evolving isolated region. Picture content from one scene is covered by one region and picture content from another scene of the transition is covered by another region. The boundary between the regions moves from picture to picture according to the transition effect.

Constant scene mixing can be implemented as follows: An isolated region covers each original source picture, and the entire picture area excluding the isolated regions forms the leftover region.

As a result of covering each original source picture by an isolated region, each slice contains data from one original picture only. Consequently, in-picture prediction within a slice is likely to succeed well, whereas in-picture prediction and loop filtering in particular is disallowed across the boundaries of source pictures. The disadvantage of the technique compared to conventional coding is that scenes can be mixed along macroblock boundaries only. However, in most cases, especially when the picture sizes are large, the disadvantage does not cause perceivable quality degradations compared to conventional coding.

VIII. Conclusion

A novel technique called isolated regions is proposed in this paper. The technique is based on constraining in-picture and interprediction jointly. It provides an elegant solution for many applications, such as gradual decoding refresh, error resiliency and recovery, region-of-interest coding and unequal error protection, picture in picture functionality, and coding of masked video scene transitions. With gradual decoding refresh based on the technique, random access, media channel switching for receivers, and allowing newcomers for multicast streaming is as easy as conventional intrapicture coding with smoother bit-rate

and high error resiliency. Future research directions include investigating proper ways to apply the isolated regions technique in other video coding standards than the AVC standard and investigating adaptive region evolution algorithms for further improved error resilience.

REFERENCES

[1] Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264 \ ISO/IEC 14 496-10 AVC). presented at Joint Video Team Doc. JVT-G050r1. [Online]. Available: ftp://ftp.imtc-files.org/jvt-experts/2003_05_Geneva/JVT-G050r1.zip

[2] Y. Wang, S. Wenger, J. Wen, and A. K. Katsaggelos, "Error resilient video coding techniques," *IEEE Signal Processing Mag.*, vol. 17, pp. 61–82, Jul. 2000.

[3] M. M. Hannuksela, "Simple packet loss recovery method for video streaming," in *Proc. Int. Packet Video Workshop PV2001*, Apr. 2001.

[4] T. Stockhammer, M. M. Hannuksela, and S. Wenger, "H.26L/JVT coding network abstraction layer and IP-based transport," in *Proc. IEEE Int. Conf. Image Processing*, Sept. 2002.

[5] G. Sullivan and G. Bjontegaard. Recommended simulation common conditions for H.26L coding efficiency experiments on low-resolution progressive-scan source material. presented at ITU-T Video Coding Experts Group Doc. VCEG-N81. [Online]. Available: ftp://standard.pictel.com/video-site/0109_San/VCEG-N81.doc

[6] Y.-K. Wang and M. M. Hannuksela. Gradual decoder refresh using isolated regions. presented at Joint Video Team Doc. JVT-C074. [Online]. Available: ftp://ftp.imtc-files.org/jvt-experts/2002_05_Fairfax/JVT-C074.doc

[7] S. Wenger. Common conditions for wire-line, low delay IP/UDP/RTP packet loss resilient testing. presented at ITU-T Video Coding Experts Group Doc. VCEG-N79. [Online]. Available: ftp://standard.pictel.com/video-site/0109_San/VCEG-N79r1.doc

[8] *Annex E, Features Supported, by the Algorithm*, ISO/IEC Int. Std. 14 496-2:2001.

[9] J. Y. Liao and J. D. Villasenor, "Adaptive intra update for video coding over noisy channels," in *Proc. IEEE Int. Conf. Image Processing*, Oct. 1996.

[10] T. Stockhammer and S. Wenger, "Standard-compliant enhancements of JVT coded video," in *Proc. 2002 Tyrrhenian Int. Workshop on Digital Communications (IWDC 2002)*, Sept. 2002.

[11] Y.-K. Wang and M. M. Hannuksela. Error-robust video coding using isolated regions. presented at Joint Video Team Doc. JVT-C073. [Online]. Available: ftp://ftp.imtc-files.org/jvt-experts/2002_05_Fairfax/JVT-C073.doc

[12] N. Brady, "MPEG-4 standardized methods for the compression of arbitrarily shaped video objects," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 1170–1189, Dec. 1999.

[13] K. Sühring. H.264/AVC Ref. Software. [Online]. Available: http://bs.hhi.de/~suehring/tml/

[14] J. Rosenberg and H. Schulzrinne. An RTP payload format for generic forward error correction. presented at IETF Internet Draft RFC 2733. [Online]. Available: ftp://ftp.ietf.org/rfc/rfc2733.txt

[15] S. Wenger and G. Côté, "Using RFC2429 and H.263 + at low to medium bit-rates for low-latency applications," in *Proc. Int. Packet Video Workshop*, Apr. 1999.

[16] Y.-K. Wang and M. M. Hannuksela. Results of the core experiment for sub-picture coding. presented at Joint Video Team Doc. JVT-B040. [Online]. Available: ftp://standard.pictel.com/video-site/0201_Gen/JVT-B040.doc

[17] M. M. Hannuksela, Y.-K. Wang, and M. Gabbouj, "Sub-picture: ROI coding and unequal error protection," in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, Sept. 2002, pp. 537–540.

Miska M. Hannuksela (M'03) received the M.S. degree in engineering from Tampere University of Technology, Tampere, Finland, in 1997.

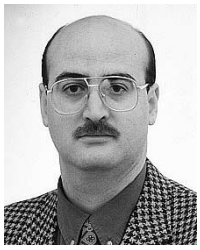
He is currently a Research Manager in the Visual Communications Laboratory, of Nokia Research Center, Tampere. From 1996 to 1999, he was a Research Engineer in the area of mobile video communications at the Nokia Research Center. From 2000 to 2003, he was a Project Team Leader and a specialist in various mobile multimedia research and product projects at Nokia Mobile Phones. He has co-authored more than 80 technical contributions to these standardization groups. His research interests include video error resilience, scalable video coding, and video communication systems.

Mr. Hannuksela has been an active participant in the ITU-T Video Coding Experts Group since 1999 and in the Joint Video Team of ITU-T and ISO/IEC since its foundation in 2001.

Ye-Kui Wang (M'02) received the B.S. degree in industrial automation in 1995 from the Beijing Institute of Technology, Beijing, China, and the Ph.D. degree in electrical engineering in 2001 from the Graduate School at Beijing, University of Science and Technology of China.

He is currently a Senior Design Engineer with Nokia Mobile Software, Tampere, Finland. From 2001 to 2002, he was a Senior Researcher with the Tampere International Center for Signal Processing, Tampere University of Technology. He has co-authored over 40 technical contributions to JVT, VCEG, and MPEG, and 18 academic papers. His research interests mainly focus on video coding and communications.

Dr. Wang has been an active participant in the Joint Video Team of ITU-T VCEG and ISO/IEC MPEG.



Moncef Gabbouj (M'85–SM'95) received the B.S. degree in electrical engineering in 1985 from Oklahoma State University, Stillwater, and the M.S. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, in 1986 and 1989, respectively.

He is currently a Professor and Head of the Institute of Signal Processing, Tampere University of Technology, Tampere, Finland. From 1995 to 1998, he was a Professor with the Department of Information Technology, Pori School of Technology and Economics, Pori, Finland, and, during 1997 and 1998, he was on sabbatical leave with the Academy of Finland. His research interests include nonlinear signal and image processing and analysis, content-based analysis and retrieval and video coding. He was co-guest editor of the *European Journal of Applied Signal Processing*, special issues on Multimedia Interactive Services (April and June 2002) and *Signal Processing*, special issue on nonlinear digital signal processing (August 1994). He is co-author of over 200 publications.

Dr. Gabbouj is the Chairman of the IEEE-EURASIP NSIP (Nonlinear Signal and Image Processing) Board. He is currently the Technical Committee Chairman of the EC COST 211quat. He served as associate editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING. He is the chairman of the IEEE Finland Section and past chair of the IEEE Circuits and Systems (CAS) Society, TC DSP, and the IEEE Signal Processing/CAS Finland Chapter. He was also the TPC Chair of EUSIPCO 2000 and the DSP track chair of the 1996 IEEE ISCAS and the program chair of NORSIG'96. He is also member of EURASIP AdCom. He was co-recipient of the Myril B. Reed Best Paper Award from the 32nd Midwest Symposium on Circuits and Systems and co-recipient of the NORSIG 94 Best Paper Award from the 1994 Nordic Signal Processing Symposium. He was the prime investigator in several EU research and educational projects and Auditor of a number of ACTS and IST projects on multimedia security, augmented and virtual reality, image and video signal processing.

[P6] D. Tian, M. M. Hannuksela, and M. Gabbouj, “Sub-sequence video coding for improved temporal scalability,” *Proceedings of IEEE International Symposium on Circuits and Systems*, vol. 6, pp. 6074-6077, May 2005.

Sub-Sequence Video Coding For Improved Temporal Scalability

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

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

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

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

I. INTRODUCTION

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

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

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

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

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

II. CONVENTIONAL TEMPORAL SCALABILITY

A. Individually Disposable Pictures

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

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

0-7803-8834-8/05/$20.00 ©2005 IEEE.

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

B. Disposal of Picture Chains

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

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

III. SUB-SEQUENCES AND H.264

A. Sub-Sequence and Sub-Sequence Layer

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

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

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

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

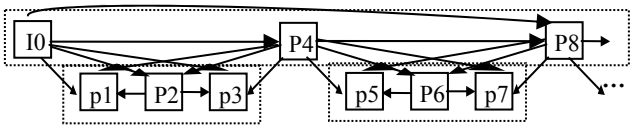


Figure 1. Example of sub-sequences: coding pattern "IpPpP" (*The numbers in the figure indicates the output order and the number of reference frames is 3.*)

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

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

B. Use of Sub-Sequences

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

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

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

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

C. Sub-Sequences in H.264

1) Overview

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

2) Gaps in frame number

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

3) Sub-sequences related SEI messages

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

4) File format

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

IV. Simulations

A. Simulation Environment

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

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

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

B. Marking Reference Pictures

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

C. Simulation Results

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

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

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

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

V. CONCLUSIONS

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

REFERENCES

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

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

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

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

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

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

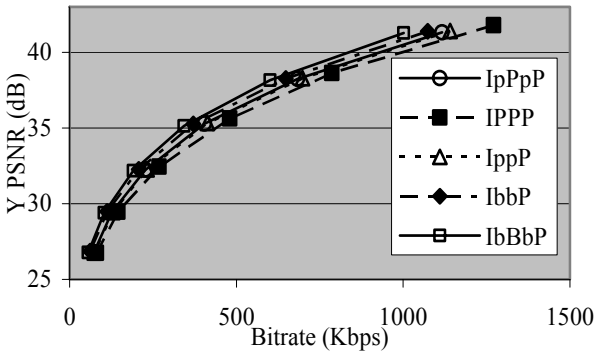


Figure 2. Rate-distortion curves for Pairs (*CIF @ 30 Hz*)

TABLE I. NUMBER OF REFERENCE FRAMES

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

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

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

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

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

[P7] D. Tian, V. K. Malamal Vadakital, M. M. Hannuksela, S. Wenger, and M. Gabbouj, "Improved H.264/AVC video broadcast/multicast," *Proceedings of Visual Communications and Image Processing 2005*, published as *Proceedings of SPIE*, vol. 5960, pp. 71-82, Jul. 2005.

Improved H.264 /AVC video broadcast /multicast

Dong Tian[*a], Vinod Kumar MV[a], Miska Hannuksela[b], Stephan Wenger[b], Moncef Gabbouj[c]

[a]Tampere International Center for Signal Processing, Tampere, Finland
[b]Nokia Research Center, Tampere, Finland
[c] Tampere University of Technology, Tampere, Finland

ABSTRACT

This paper investigates the transmission of H.264 /AVC video in the 3GPP Multimedia Broadcast /Multicast Streaming service (MBMS). Application-layer forward error correction (FEC) codes are used to combat transmission errors in the radio access network. In this FEC protection scheme, the media RTP stream is organized into source blocks spanning many RTP packets, over which FEC repair packets are generated. This paper proposes a novel method for unequal error protection that is applicable in MBMS. The method reduces the expected tune-in delay when a new user joins into a broadcast. It is based on four steps. First, temporally scalable H.264 /AVC streams are coded including reference and non-reference pictures or sub-sequences. Second, the constituent pictures of a group of pictures (GOP) are grouped according to their temporal scalability layer. Third, the interleaved packetization mode of RFC3984 is used to transmit the groups in ascending order of relevance for decoding. As an example, the non-reference pictures of a GOP are sent earlier than the reference pictures of the GOP. Fourth, each group is considered a source block for FEC coding and the strength of the FEC is selected according to its importance. Simulations show that the proposed method improves the quality of the received video stream and decreases the expected tune-in delay.

Keywords: Video streaming, H.264 /AVC, MBMS, 3GPP, FEC

1. INTRODUCTION

Video coding and transmission have been widely studied in recent decades and a several successful standards have been developed. The latest video coding standard, H.264 /AVC, was jointly developed by the ITU-T and MPEG community and its first version was ratified in 2003[2]. It has proven its superiority over its predecessors in terms of coding efficiency and error resiliency. H.264 /AVC continues to be based on the traditional technique of motion compensation and transform coding of the residual signal. However, a number of advanced features have been added, such as the possible use of multiple reference pictures, and variable block sizes for the motion prediction. Excellent compression efficiency and network-friendliness make H.264 /AVC a competitive candidate for future applications such as 3GPP's Multimedia Broadcast /Multicast Streaming service (MBMS). Due to the anticipated high demands for the video streaming over the mobile network, video streaming based on H.264 /AVC has been one of the focuses in the 3GPP standardization community.

MBMS is a point-to-multipoint service in which data is transmitted from a single source entity to multiple recipients. It has been standardized in 3GPP release 6. A general description of MBMS systems can be found in the technical specification [1]. As depicted in Figure 1, the content delivery of MBMS is conceptually divided into three layers: bearer, delivery method, and user service.

The MBMS bearer defines the architecture to transport data from a single source to multiple receivers. Two delivery methods have been specified: download and streaming. Software update is an example application of the download delivery method, whereas live video is an example using the streaming delivery method. This paper is concerned only with the streaming delivery method.

MBMS uses IP/UDP/RTP transport without underlying guarantied delivery. The packet loss rates perceived at the receiver are highly variable, and depend primarily on the signal quality of the wireless link. This quality is influenced by factors beyond the network's control, such as the physical location and the speed of the receiver relative to the base station. Since the broadcast nature of MBMS does not allow for ARQ-type repair techniques, and since the expected error rates are too high to depend solely on source-coding based tools, a packet-based forward error correction (FEC)

[*] Email: dong.tian@tut.fi, Phone: +358-40-8282128

Visual Communications and Image Processing 2005, edited by Shipeng Li, Fernando Pereira, Heung-Yeung Shum, Andrew G. Tescher, Proc. of SPIE Vol. 5960 (SPIE, Bellingham, WA, 2005) · 0277-786X/05/$15 · doi: 10.1117/12.631439

scheme has been introduced. Utilizing FEC allows reducing the packet loss rate as perceived by the media decoder to zero in virtually all cases. Only at extreme error rates, or when an insufficient FEC strength has been chosen, a packet loss rate above zero may be observable at the media receiver. Assuming FEC at an appropriate strength has the distinct advantage of allowing encoded media without any bits being spent for source-coding based error resilience, which in turn leads to higher coding efficiency and an overall better quality of experience.

In order to be efficient in a highly bursty packet lossy scenario, a FEC block must be as large as possible. Under consideration are FEC block sizes of several dozen packets, which require several seconds for the transmission over the (comparatively slow) links. While efficient from an error recovery point-of-view, such large FEC blocks have negative properties from a user experience point-of-view: since a whole FEC block needs to be received before repair can commence, the tune-in delay is at least as long as the duration of the FEC block – unless FEC repair is not used during the tune-in phase.

This paper proposes a smart delivery order of packets to reduce the tune-in delay and a novel method for unequal error protection to improve error resilience.

This paper is organized as follows. Section 2 analyzes the problem with the use of FEC, including the abrupt degradation in quality and the tune-in delay introduced. In section 3 we propose solutions along with examples to protect H.264 /AVC bitstreams unequally and a novel transmission order of coded video data to reduce the tune-in delay. Section 4 provides the details about the simulations. Conclusions are presented in section 5.

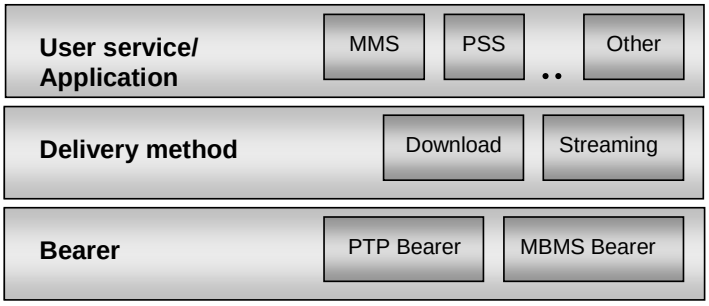


Figure 1. The three layers in MBMS

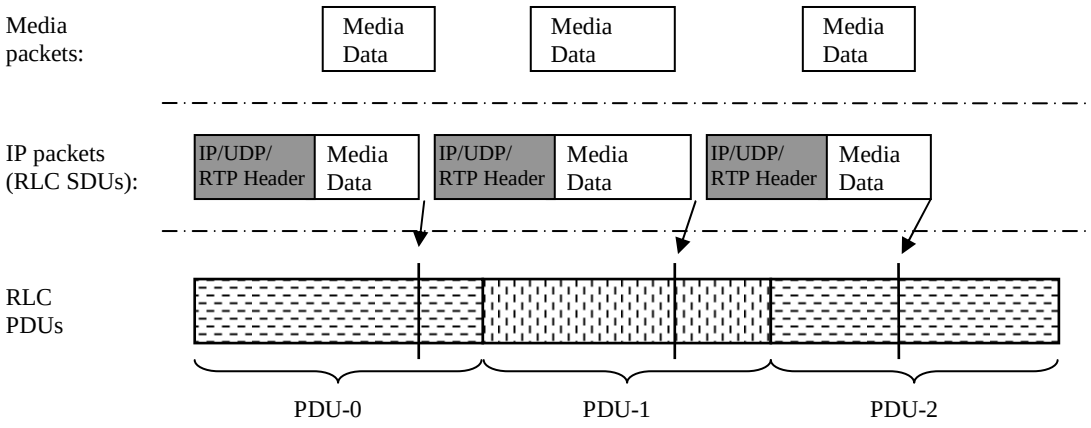


Figure 2. Illustration of the formation of PDUs from media data

2. PROBLEMS

2.1. Background

H.264 /AVC provides a friendly interface between the video coding layer (VCL) and network transmission layer. Each picture is segmented into slices, whereby a slice encompasses an integer number of macroblocks, between one and all macroblocks of a picture. Each slice is encapsulated into one NAL unit which can be considered as the smallest independently decodable unit. All data referring to more than one slice is part of a parameter set, and may be transported by means different from those used to transport slices.

The NAL units are encapsulated into RTP packets. RFC3984[5] defines the RTP payload format for H.264 /AVC. It specifies three packetization modes: single NAL unit mode, non-interleaved mode and interleaved mode. The single NAL unit mode and non-interleaved mode are targeted towards conversational systems, in which NAL units are transmitted in NAL unit decoding order, thereby minimizing delay. The interleaved mode is targeted towards systems with relaxed end-to-end latency demands, e.g. broadcast /multicast systems like MBMS. The interleaved mode allows transmission of NAL units out of NAL unit decoding order. A decoding order number (DON) - a field in the RTP payload header or a derived variable - is employed to re-establish the decoding order. In this paper, only the interleaved mode is considered.

RTP packets are transported over UDP/IP. On the sub-IP layers, we assume that one Radio Link Layer (RLC) SDU consists of a single IP packet including its uncompressed header. RLC SDUs are framed and mapped into RLC PDUs (radio data blocks) for the delivery over the MBMS bearer service. It should be noted that the PDU size are constant across the whole transmission session while the SDU sizes are varying due to the variable sized slices in video, and thus the PDUs are not aligned to the SDUs (IP packets) as shown in Figure 2. Loss of a single RLC PDU would cause destruction of all the involved SDUs.

The protocol overhead can be assumed as follows: 12 bytes for RTP, 8 bytes for UDP, and 20 bytes for IPv4 or 40 bytes for IPv6. In addition there is a small overhead per NAL unit due to the use of the sophisticated payload header of RFC3984's interleaved mode. Header compression may be employed to reduce the size of the IP/UDP/RTP headers, but is not further considered here.

In the 3GPP technical specification related to MBMS[1], as illustrated in Figure 3, the FEC is implemented as a meta-payload hierarchically located between RTP and the media payload. The processing can be outlined as follows: an RTP packet, generated by the media encoder, is modified by inserting a FEC payload ID (in the form of a payload header) that indicates the position of the bits of the packet in the to-be-formed FEC block. Furthermore, the RTP payload type is modified so to indicate the presence of the FEC payload ID. The modified RTP packet is sent using the normal RTP mechanisms. In addition, the original RTP packet is also copied into a data structure over which the FEC encoding is run. Once a sufficient amount of data is collected (the FEC block is filled up with variable length RTP packets), the FEC algorithm is applied to calculate a number of repair packets. Those repair packets are also being sent using RTP, and SSRC multiplexing is employed to identify the two different streams.

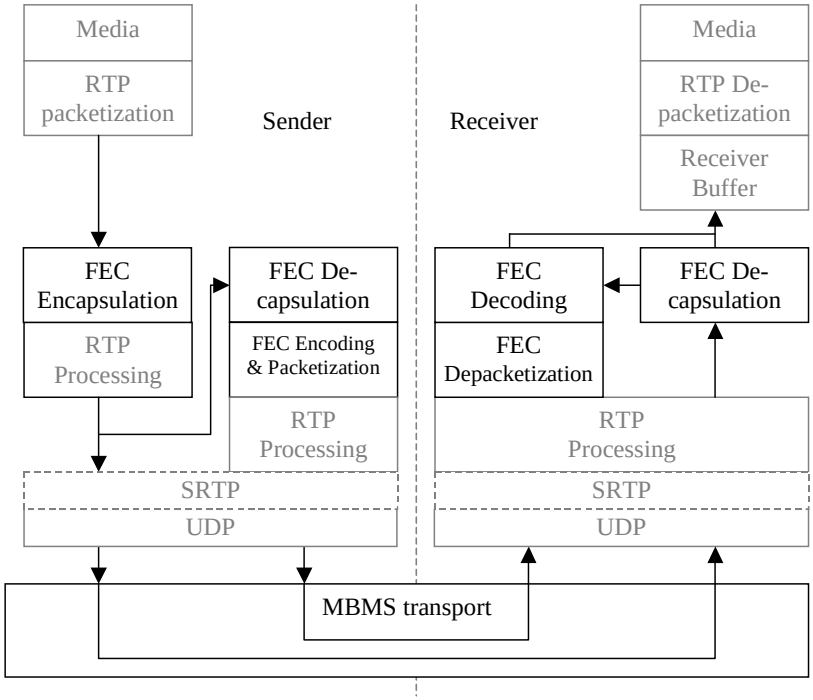


Figure 3. The MBMS system with FEC

At the receiver, first, all media packets and all repair packets belonging to the same FEC block are collected. This is possible through the information of the repair packet payload header and the FEC payload ID. Then, the FEC decoding is applied, resulting in the reconstruction of any missing packets. The special case of insufficient FEC strength, in which not all media packets can be recreated, does obviously not allow to reconstruct missing packets - however, the correctly received media packets are still available and could, after removal of the FEC payload ID, be used for media reconstruction.. The received media packets are transformed into their original state by removing the FEC payload ID and changing the Payload Type to its original value. Finally, the received and the reconstructed media packets are re-sequenced utilizing the RTP sequence number.

2.2. Delay in FEC protected MBMS system

2.2.1. Initial Delay

Hypothetical decoder (HD) models are defined to set up the minimum requirements for the bitstream flows. They are typically composed of a hypothetical buffer and an instantaneous decoder. Such models can be used by the sender to verify the transmitted bitstream does not cause underflow or overflow in the receiver's buffer. A H.264 /AVC HD is defined in Annex C of [2] and a detailed description of an MBMS FEC HD can be found in [7]. In the MBMS system with FEC, since two such HDs are cascaded, additional requirements for the FEC HD are to be met to guarantee the H.264 /AVC HD does not underflow or overflow.

The FEC duration, T_{FEC}, includes not only the actual transmission time of the FEC block, but certain amount of initial buffering delay, D_H, to keep the buffer of media decoder not to underflow or overflow according to the hypothetical decoder models. The value of T_{FEC} may vary from one FEC block to another, and therefore, a variable delay D_C is proposed to be present for each FEC block, allowing for optimization of the initial buffering delay in receivers (see [7] for details). Alternatively, the receiver has to delay the decoding of the media source packets for such a period of the maximum FEC duration, $\max(T_{FEC})$, across the streaming session so as to maximize the probability of correct reception of media samples and to maintain a regular presentation rate of the media samples at the same time [1]. Another factor in the initial delay is the maximum FEC decoding time for a FEC block within the whole streaming session, which is denoted as D_F in Figure 4. The so-called initial delay D_I is therefore can be expressed as,

$$D_I = \max(T_{FEC}) + D_F.$$

It should be noted that the initial delay is until the decoding of the media source packets and further delay is needed for the rendering of the decoded media samples. For H.264 /AVC, the additional delay for rendering may be signaled in the bitstream, e.g. by the value of num_reorder_frames in the VUI structure or picture timing SEI messages [2]. In this paper we exclude the rendering delay from the consideration.

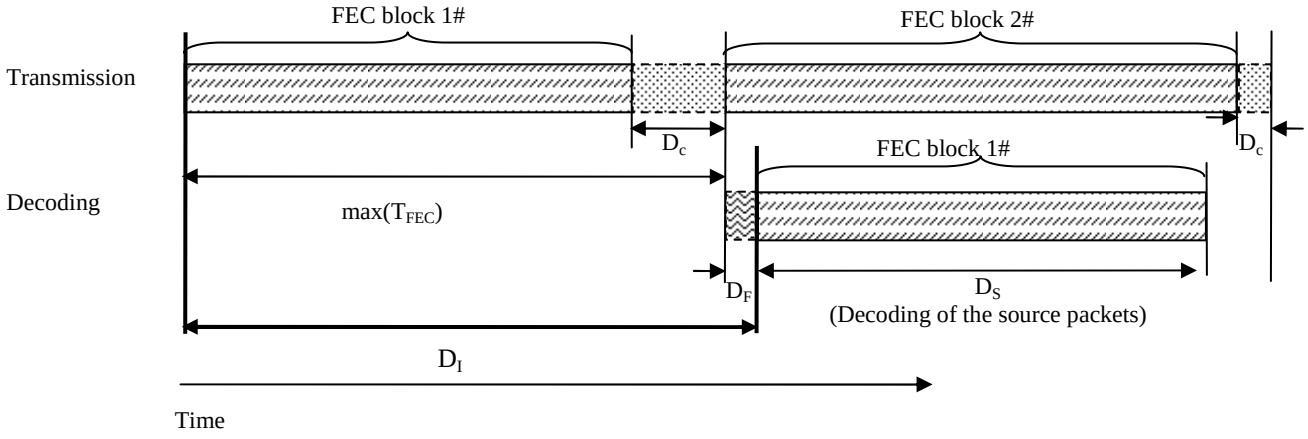


Figure 4. Initial buffering delay

2.2.2. Tune-in Delay

Tune-in delay is defined as the duration from the start reception of packets to the start of correct decoding of packets. It is experienced by a new user who joins the ongoing broadcast, and the tune-in point (the first received packet) is anywhere but at the very start of a FEC block. To successfully tune in, packets representing a random access point (e.g. in the form of an IDR picture, which is assumed henceforth) have to be available. Ideally, all packets after the random access point also have to be available - only this allows for the full user experience. However, even if some packets are missing and the resulting picture degrades, the resulting user experience is perhaps still higher compared to displaying no picture at all.

Frequent random access points are desired to facilitate a shorter tune-in delay and enhanced error resilience, but not wanted regarding to the coding efficiency. As a compromise, a consensus within the 3GPP community is that the FEC block boundaries are aligned at the IDR pictures [8].

Based on the thoughts above, we can imagine two different tune-in strategies. In the first strategy, the receiver first synchronizes to the FEC block structure, i.e. waits for the reception and successful processing of one complete FEC block, before attempting the media decoding. In the second strategy, the receiver searches the media packets as received, disregarding FEC block boundaries, for an random access point. Once found, it starts decoding the random access point and any following pictures regardless of the status of the FEC repair engine. The latter approach obviously allows for a shorter tune-in time, but at the expense of the chance of a seriously degraded picture quality due to losses.

In this paper, we do not expect to rely on the FEC decoder to recover the source packets prior to the tune-in point in the tune-in FEC block. Therefore, the second tune-in strategy is used.

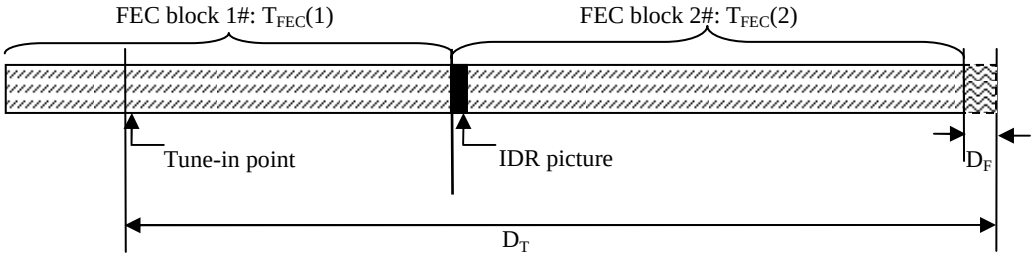


Figure 5. Tune-in delay in MBMS

Conventionally, the packets are sent in the same order of their decoding, in which the IDR picture is sent at the beginning of the FEC block. As a consequence, nothing can be reproduced from the tune-in FEC block and we have to wait until the following FEC block is received. As shown in Figure 5, the delay of D_T can be expressed as,

$$D_T = r * T_{FEC}(1) + T_{FEC}(2) + D_F \quad (1)$$

where r is the percentage of packets being received in the tune-in FEC block, $T_{FEC}(1)$ and $T_{FEC}(2)$ are the durations of the tune-in FEC block and its succeeding FEC block, respectively. Suppose r to be an evenly distributed random variable between 0% and 100%, exclusively. The duration of a FEC block is assumed to be much longer than the hypothetical transmission time of one packet, then r and T_{FEC} can be treated to be statistically independent, and hence we have,

$$E(D_T) = 1.5 * E(T_{FEC}) + D_F, \quad (2)$$

Note that the tune-in delay D_T is not sufficient for a regular decoding rate, but a delay to have a correct decoding of pictures. For a new user, the delay D_P until a regular decoding rate, is called as tune-in period hereinafter,

$$D_P = \max(D_I, D_T). \quad (3)$$

And its expectation is,

$$\begin{aligned} E(D_P) &= \max(E(D_T), D_I). \\ &= \max(1.5 * E(T_{FEC}) + D_F, \max(T_{FEC}) + D_F) \\ &= \max(1.5 * E(T_{FEC}), \max(T_{FEC})) + D_F. \end{aligned} \quad (4)$$

For example, if the T_{FEC} is always of 5 seconds, the expectation of the tune-in delay D_T is then 7.5 seconds, even larger than the initial buffering delay of 5 seconds (with the assumption of an instantaneous FEC decoder, $D_F = 0$). It is irritating for a new user to endure such a long tune-in delay without anything to be decoded and hence to be presented.

In addition it shall be noted that a user may suffer a delay equivalent to the tune-in delay with the FEC synchronization point being lost.

3. GOP STRUCTURE AND PACKET TRANSMISSION ORDER

We propose the combination of two mechanisms to enhance reproduced quality and, simultaneously, reduce the tune-in delay: First, we encode the video utilizing a GOP structure employing sub-sequences and apply unequal error protection for different sub-sequence layers. Second, we remove the requirement of receiving the whole FEC block, by re-ordering the packets in an ascending order of decoding relevance.

3.1. Coding in sub-sequences and unequal error protection

3.1.1. Sub-sequence in H.264 /AVC

Sub-sequences are a form of temporal scalability coding that has been made possible by the introduction of reference picture selection. They were first proposed for H.263+ [6], but have gained popularity only in conjunction with H.264 /AVC. The sub-sequence concept can perhaps best be introduced by an example. Consider Figure 6. A “base layer” and a single “enhancement layer”, are depicted. The base layer consist of the IDR picture I0, and the predicted pictures P4 and P8. The IDR picture, by definition, is not predicted from any previous or future picture. P4 is predicted only from I0, and P8 is primarily predicted from P4; however, since H.264 employs reference picture selection on a macroblock level for coding efficiency reasons, some macroblocks may also be predicted from I0 if the encoder determines that this saves bits. This base layer operates with a frame skip of 3 source pictures.

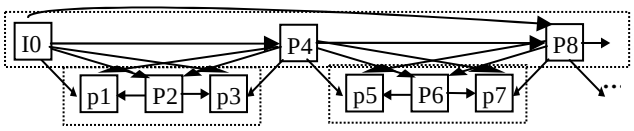


Figure 6. Example of sub-sequences: (*The numbers in the figure indicates the output order*)

Two sub-sequences are further depicted as part of a temporal enhancement layer. In the first of these sub-sequences, picture P2 (note the capital letter P) is predicted only from I0 and P4, but not from p1 and p3. Hence this picture is independent from p1 and p3. However, P2 is used as a reference picture for p1 and p3. Finally, p1 and p3 are predicted from all surrounding pictures including P2. A similar prediction relation consists between p5, P6, and p7. None of the latter pictures have any explicit or implicit prediction relationship with p1, P2, or p3.

Generalized, one can make the following statement: A sub-sequence represents those predictively coded pictures that can be disposed without affecting the decoding of any other sub-sequence in the same sub-sequence layer, or any sub-sequence in any lower sub-sequence layer. The sub-sequence technique enables easy identification of disposable chains of pictures when processing pre-coded bitstreams.

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

While the larger temporal difference between pictures in the base layer does yield a less efficient encoding and hence more bits, it is possible to adjust the QPs of the higher layers to values offering a lower fidelity, very similar to what is commonly done in MPEG-2 B pictures. It was found in [4] that the temporal scalability can be achieved with no significant deterioration in coding efficiency.

3.1.2. Sub-sequence and unequal error protection in MBMS

When employing sub-sequences, the video stream is arranged into so-called super FEC blocks, each of which contains an integer number of consecutive FEC blocks (i.e. source blocks and associated repair packets) in transmission order. All slices in a super FEC block must succeed in decoding order any slice in previous super FEC blocks and must precede in decoding order any slice in succeeding super FEC blocks. In other words, super FEC blocks form self-contained sequences of coded pictures.

To improve the error resilience, we employ unequal error protection (UEP) for different sub-sequence layers. In MBMS, the network conditions as perceived by the individual receiver vary widely. Therefore, the media transmission has to be tailored assuming relatively bad network conditions. It is up to the operator to determine the best operation point according to its business model, i.e. what a failure rate he is willing to accept. Nevertheless, we follow assumptions as used in 3GPP, where a 10% RLC PDU loss rate is considered a worst case for UTRAN streaming [11].

The loss rate, as perceived by the RTP receiver, can be significantly higher than the RLC PDU loss rate, depending on the size and the alignment of the RTP packets to the RLC PDUs. In order to simplify the discussion, we make worst case assumption: each PDU contains parts of three RTP packets - some last bytes of a first RTP packet, a complete second RTP packet, and some bytes of a third RTP packet. Therefore, in a bad case, SDU loss rate is approximately three times of PDU loss rate, i.e. 30%. Let m be the number of RTP packets in a FEC block. The expected number of received video packets is $0.7*m$, and the expected number of video packets to be corrected is $0.3*m$. Consequently, a minimum $0.3*m$ repair packets should be received for the FEC block. When the same loss rate is applied for repair packets, the transmitted number of repair packets n should be such that $0.7*n=0.3*m <=> n=(0.3*m)/0.7 <=> n=(3/7)*m$. To make it an integer number, $n = ceil(3/7*m)$.

In the simulations discussed later, the FEC bitrate for reference pictures is selected as discussed above, so to allow virtually all RTP packets belonging to reference pictures to be repaired. For RTP packets belonging to non-reference pictures we arbitrarily select a FEC strength that is only $1/10^{th}$ of the FEC strength for reference pictures.

3.2. Transmission order and reduced tune-in delay

The algorithm in this subsection helps to minimize the tune-in delay by avoiding the reception delay of a complete FEC block before beginning decoding. Furthermore, by applying this algorithm, a reduced frame rate may be possible even when only parts of the FEC block are received. We assume here to start decoding as soon as the moment of tune in, without initially relying on FEC-protection. When packetizing pictures in decoding order and assuming a low random access point frequency (e.g. one per FEC block), doing so does not yield meaningful results, since decoding would start

somewhere in the middle of a sequence. More than one random access point per FEC block has negative impact on the compression efficiency and is useless except for tune-in, and hence should be avoided.

In order to reduce the tune-in delay, we employ RFC3984's interleaved packetization mode to place all NAL units belonging to the more important pictures (e.g. the IDR picture, and a few P pictures) towards the end of the FEC block. Even without FEC correction, we assume that in many cases at least the IDR picture (and perhaps a few of the P pictures) can be successfully taken from the packet stream and be reconstructed. This results in having a first visible signal available after a very short tune-in delay (perhaps as short as a few hundred milliseconds). The algorithm, discussed in more detail later, applies to the super FEC block layer and FEC block layer respectively.

3.2.1. Super FEC block layer

In a super FEC block, the media samples are organized into more than one group according to the layers of the prediction hierarchy. Within each layer, any group can be decoded independently from the other groups, as long as the hierarchically higher layers are available.

To reduce tune-in delay, we arrange the groups into FEC blocks in the order of importance for reproduction - the most important groups are placed into FEC blocks that are transmitted last in the super FEC block. For a two-layered system, as discussed before in section 3.1, the super FEC block consists of FEC blocks of two classes: those which carry the base layer information and those which carry the enhancement layer information. In this example, the FEC blocks with the oldest enhancement layer information would be placed first in the super FEC block, followed by newer enhancement layer information, older base layer information, and newer base layer information. The scheme could be easily expanded to more than two layers following the same rationale.

When tuning in to a stream somewhere two thirds in a super FEC block, this would result in the loss of the enhancement layer and some older parts of the base layer. However, the more recent base layer data (in the form of complete FEC blocks, hence including a random access point) would be still available, allowing displaying a video sequence with reduced frame rate.

3.2.2. FEC block layer

In many cases packets from pictures can be ranked beyond static layering according to their relevance for the decoding process. Referring back to subsection 3.1, it should be clear that even in the base layer, the picture I0 is more important than the picture P4 and P8. P4, again, is more important than P8, because P4 is required to reconstruct P8 but not vice versa. The ordering criteria is the inverse of the decoding order - pictures earlier in decoding order are more important than pictures later in decoding order.

Utilizing RFC3984's interleaved mode, it is possible to put data belonging to less important pictures towards the beginning of a FEC block, and pictures with higher importance towards the end of the FEC block.

The FEC repair packets follow the source packets. Let m be the number of media source packet and n be the number of FEC repair packets.

If only some FEC repair packets are received in the tune-in FEC block, the tune-in delay cannot be reduced compared to traditional transmission order. We discuss how the tune-in delay is reduced with at least one source packet available.

As in subsection 2.2, the first presented picture of a FEC super block is expected to be IDR coded in H.264 /AVC, which is sent after packets from all other pictures. Supposing at least the IDR picture is received, the tune-in delay is,

$$D_T = r * T_{FEC} - D_h, \tag{5}$$

Since the decoding of the FEC code is unnecessary for the tune-in FEC block, we need not wait for the reception of the FEC packets and thus shall reduce the corresponding part in the delay of D_H:

$$D_h = n * D_H / (n+m), \tag{6}$$

where n is the number of FEC repair packets and m is the number of media source packets.

And the expectation of D_T is,

$$E(D_T) = 0.5\ E(T_{FEC}) - E(D_h), \tag{7}$$

with at least 2/3 reduction in the tune-in delay compared to 1.5 $E(T_{FEC})$ in (2). With the reduced tune-in delay, some pictures can be displayed before the regular presentation rate can be achieved and a better user experience can be expected.

In the example with the FEC duration fixed to be 5 seconds, the expected tune-in delay will be reduced from 7.5 seconds to 2.5 seconds. In the first 2.5 seconds, nothing can be presented and in the subsequent 5 seconds, some pictures can be rendered, and finally (7.5 seconds since the tune-in point) the pictures can be presented at the regular rate.

Additionally, for an old user that loses the FEC synchronization point, more pictures can be rescued with the proposed transmission order compared with the conventional transmission order.

3.3. Examples of transmission order

Let's explore some examples to gain an understanding how the tune-in delay is reduced.

3.3.1. Example 1 with single sub-sequence layer

In this first example, the video is coded in IPP with the IDR frequency to be 16. Each picture is coded into one slice. Traditionally, each FEC block would consist of 15 video packets and 5 repair packets, and they are sent in the following order,

… **[** I_0 **][** P_1 **][** P_2 **][** P_3 **][** P_4 **][** P_5 **][** P_6 **][** P_7 **][** P_8 **][** P_9 **][** P_{10} **][** P_{11} **][** P_{12} **][** P_{13} **][** P_{14} **][** FEC **]** …

Where I_0 is an IDR picture, P_x stand for P pictures, **[** FEC **]** stand for the five FEC repair packets. The conventional transmission order will be the same as the coding order.

However, employing our algorithm, the transmission order looks as follows:

… **[** P_{14} **][** P_{13} **][** P_{12} **][** P_{11} **][** P_{10} **][** P_9 **][** P_8 **][** P_7 **][** P_6 **][** P_5 **][** P_4 **][** P_3 **][** P_2 **][** P_1 **][** I_0 **][** FEC **]** …

To simplify the following discussions, let's assume the transmission time for each packet to be constant. As an example, assume the tune-in point to be the 11^{th} packet in the FEC block. In the conventional transmission order, the tune-in point is at the P_{10}. The decoder cannot decode anything meaningful until it reaches the next IDR in the following FEC block; hence the tune-in delay is 10+20=30 packets of (hypothetical) transmission time.

With our modified transmission order, the tune-in point would be at P_4. After re-ordering, the decoder can decode the packets from I_0 to P_4, and the tune-in delay is the (hypothetical) transmission time of 5 packets. During the reception of the subsequent FEC block, 5 video packets can be decoded and displayed. See Figure 7 for the illustration, where the FEC decoding time is ignored.

3.3.2. Example 2 with two sub-sequence layers

The second example shows how sub-sequence coding, UEP and transmission reordering can be designed jointly. Assume that each picture is coded into one slice and the IDR refresh rate is set to 15. Two non-reference pictures (marked as "p") are coded between two successive reference pictures (either IDR picture, marked as "I", or reference inter picture, marked as "P").

Presentation order is,

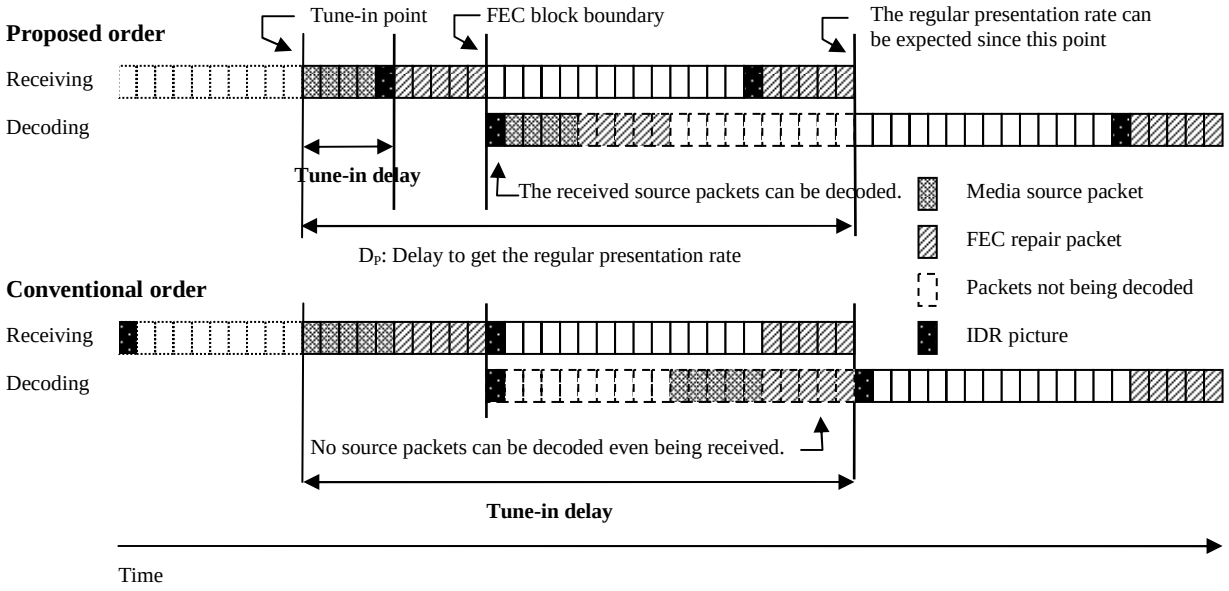


Figure 7. How the tune-in delay is reduced in example 1

… [I_0][p_2][P_1][p_4][P_3][p_6][P_5][p_8][P_7][p_{10}][P_9][p_{12}][P_{11}][p_{14}][P_{13}], [I_{15}] … .

Conventional transmission order is,

… [I_0][P_1][p_2][P_3][P_4][P_5][p_6][P_7][p_8][P_9][p_{10}][P_{11}][p_{12}][P_{13}][p_{14}][FEC_1][FEC_2] … .

And the proposed transmission order is,

… [FEC_1][p_2][p_4][p_6][p_8][p_{10}][p_{12}][p_{14}][FEC_2][P_{14}][P_{12}][P_{10}][P_8][P_6][P_4][P_2][I_0], …

"p" stands for a non-ref picture, "P" /"I" stands for a predictive reference picture /IDR picture. In case of all the non-ref pictures are not received, the receiver can still have a 1/2 of the full frame rate. In the worst case, that only I_0 is received, the receiver can display the first picture while waiting for the next complete FEC block during the reception of the subsequent FEC super block, which is better than nothing to be displayed.

For the FEC block of the non-reference pictures, the packets can be transmitted in any order, because every packet does not use any other packet in the block.

4. SIMULATIONS

4.1. Common Simulation Conditions

Simulations were performed following the draft video simulation conditions for 3GPP services[10] as closely as possible. We had to diverge from the condition in [10] in the following areas:

- Due to copyright problems, the Nasa sequence could not be employed. Hence we used the Tour of Glasgow sequence.
- In [10], it is suggested to perform a single simulation run of a 60-second sequence. In order to gather statistically relevant results, we used 50 simulation runs of a 50-sec sequence.

4.2. Video encoding

The picture rate of the Glasgow Tour sequence is originally 12.5 Hz but is considered herein to be 15 Hz (or 30000/2002 Hz, to be exact). Consequently, the duration of the clip is 50 seconds. A constant picture rate of 7.5 Hz was used in all the coded streams. We always code the first picture of a new scene with IDR in H.264 /AVC. There are totally 23 scenes in the Glasgow Tour sequence.

The target bitrate for video and its FEC were calculated by subtracting the FEC bitrate from the channel bitrate. The remaining bitrate was set as the target bitrate for video encoding.

In order to produce the bitstream with the required bitrate, we applied a simple rate control method as follows. A constant quantization parameter (QP) value, herein QP_1, resulting into closest bitrate larger than the target bitrate was first found by trial and error. In order to achieve the target bitrate more accurately, the bitstream was then coded with two QP values, QP_1 for the first pictures in the stream and QP_1+1 for the remaining pictures in the stream. An optimal "change-point picture" (the picture in which the change of QPs happens), resulting into closest bitrate compared to the target bitrate, was searched by trial and error.

Two codec configurations were tested:

1. H.264 /AVC Baseline with constraint_set1_flag = 1. All coded pictures are reference pictures. This codec configuration is referred to as H.264 /AVC IPP hereinafter.
2. H.264 /AVC Baseline with constraint_set1_flag = 1. Every other coded picture is a non-reference picture coded similarly to a B picture in conventional video coding. For more details on the use and benefits of non-reference pictures in H.264 /AVC Baseline, please refer to [4] and [9]. This codec configuration is referred to as H.264 /AVC IpP hereinafter.

The maximum slice size of H.264 /AVC was set to 500 bytes.

4.3. FEC coding

We used Reed-Solomon FEC coding and simple source block generation (one media RTP packet to one column of the source block). To minimize initial buffering delay, source block boundaries were made to match scene boundaries, i.e. the first picture of a source block was an IDR picture. We believe that the results are applicable to more complex Reed-Solomon schemes and other FEC schemes as well.

It was assumed that at the time of encoding, the media encoder and FEC encoder do not have knowledge of the prevailing channel conditions but have to tailor the stream according to expected worst case, i.e. 10% PDU loss rate.

To achieve optimal quality for 10% PDU loss rate, we used a small number of trials and errors and some reasoning as follows:

- We tried suggested 1:3 share of FEC and media bitrate in [11]. However, this FEC code rate turned out to be too low for UTRAN 10% resulting into several dBs of quality drop in average luma PSNR.
- We tried using an adaptive intra macroblock refresh (AIR) algorithm in video encoding without any FEC coding. This resulted into inferior performance compared to coded sequences without AIR, protected with FEC coding.
- We made the reasoning in subsection 3.1.2 to find out the number of FEC repair packets.

For H.264 /AVC IpP codec configuration, non-reference and reference pictures for each group of pictures (from an IDR picture, inclusive, to the next IDR picture, exclusive) were arranged such that non-reference pictures were transmitted earlier than reference pictures. This arrangement minimizes the expected tune-in delay as explained in subsection 3.3.2.

4.4. Packet Loss Simulation

At the time of running the simulations, only the PDU loss patterns under UTRAN for 1% loss rate were available in 3GPP SA4. Therefore, we produced the 10% loss rate pattern ourselves (a random error pattern was used).

We ran 50 simulations of the 50-sec coded stream to get statistically reliable results. A random error pattern starting position was generated for each run, and the same starting positions were used for all codec configurations.

4.5. Decoding

Corrupted SDUs were identified and were discarded in the receiver. FEC decoding is applied to recover missing media packets, whenever possible. An error concealment algorithm similar to TCON (of H.263 Test Model Reference) is applied in both cases.

Table 1. Simulation results

	Total bitrate (video+FEC) (kbps)	FEC bitrate share (%)	Average luma PSNR (dB)		
			Error Free	PLR 1%	PLR 10%
H.264 (IPP)	44.19	42.6%	27.97	27.97	27.38
H.264 (IpP)	44.18	19.0%	28.43	28.43	27.55

4.6. Simulation Results

The simulation results are summarized in the Table 1. It can be seen that the H.264 /AVC IpP codec configuration accompanied by the unequal protection of reference and non-reference pictures improves the performance especially in error-free transmission and packet loss rate (PLR) 1% case. Furthermore, the H.264 /AVC IpP codec configuration enables lower tune-in delay.

5. CONCLUSIONS

In this paper, we studied the video transmission of the H.264 /AVC in MBMS over 3GPP. We proposed a novel method for unequal error protection to achieve better error resilience, which requires scalable coding of H.264/AVC and arranging of transmission order of the pictures based on their importance. We also analyzed the delays in the MBMS streaming and introduced a unconventional transmission order of the packets so to a shorter tune-in delay and a better user experience.

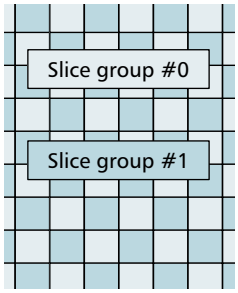
REFERENCES

1. 3GPP TS 26.346 V1.7.0, “Multimedia broadcast /multicast service protocols and codecs (Release 6)”, Feb 2005
2. T. Wiegand, G. Sullivan (editors), “Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC)”, document JVT-G050, 2003
3. Tdoc S4-040671, “Video simulations for MBMS streaming”, Nov 2004
4. D. Tian, M. M. Hannuksela, M. Gabbouj, “Sub-sequence video coding for improved temporal scalability”, ISCAS 2005, Kobe, Japan, May 2005
5. S. Wenger, M. M. Hannuksela, et. al. “RFC3984 - RTP payload format for H.264 video”, Feb 2005
6. S. Wenger, “Temporal scalability using P-pictures for low-latency applications”, Multimedia Signal Processing Workshop 1998
7. Tdoc S4-040672, “FEC buffering for MBMS streaming delivery method”, Nov 2004
8. Tdoc S4-050068, “Media alignment to FEC structures in MBMS streaming”, Feb 2005
9. Tdoc S4-040743, “Reduction of tune-in delay in MBMS streaming”, Nov 2004
10. Tdoc S4-040582, “Draft video simulation conditions for 3GPP services”, Aug 2004
11. Tdoc S4-040348, “Simulation guidelines for the evaluation of FEC methods for MBMS download and streaming services”, May 2004

[P8] T. Stockhammer and M. M. Hannuksela, “H.264/AVC video for wireless transmission,” *IEEE Wireless Communications*, vol. 12, no. 4, pp. 6-13, Aug. 2005.

H.264/AVC VIDEO FOR WIRELESS TRANSMISSION

THOMAS STOCKHAMMER, NOMOR RESEARCH
MISKA M. HANNUKSELA, NOKIA RESEARCH CENTER



The authors introduce the features of the H.264/AVC coding standard that make it suitable for wireless video applications, including features for error resilience, bit rate adaptation, integration into packet networks, interoperability, and buffering considerations.

ABSTRACT

H.264/AVC will be an essential component in emerging wireless video applications thanks to its excellent compression efficiency and network-friendly design. However, a video coding standard itself is only one component within the application and transmission environment. Its effectiveness strongly depends on the selection of appropriate modes and parameters at the encoder and decoder, as well as in the network. In this article we introduce the features of the H.264/AVC coding standard that make it suitable for wireless video applications, including features for error resilience, bit rate adaptation, integration into packet networks, interoperability, and buffering considerations. Modern wireless networks provide many different means to adapt quality of service, such as forward error correction methods on different layers and end-to-end or link layer retransmission protocols. The applicability of all these encoding and network features depends on application constraints such as the maximum tolerable delay, possibility of online encoding, and availability of feedback and cross-layer information. We discuss the use of different coding and transport related features for different applications: video telephony and conferencing, video streaming, download-and-play, and video broadcasting. Guidelines for the selection of appropriate video coding tools, video encoder and decoder settings, and transport and network parameters are provided and justified. References to relevant research publications and standards contributions are given.

INTRODUCTION

Most emerging and future mobile client devices will significantly differ from those used for speech communications only: handheld devices will be equipped with a color display and a camera, and have sufficient processing power to allows presentation, recording, and encoding/decoding of video sequences. In addition, emerging and future wireless systems will provide sufficient bit rates to support video communication applications. Nevertheless, bit rates will always be scarce in wireless transmission environments due to physical bandwidth and power limitations; thus, efficient video compression is required. Nowadays H.263 and MPEG-4 Visual Simple Profile are commonly used in handheld products, but it is foreseen that H.264/AVC [1] will be the video codec of choice for many video applications in the near future. The compression efficiency of the new standard outdoes prior standards roughly by at least a factor of two. Although compression efficiency is the major feature for a video codec to be successful in wireless transmission environments, it is also necessary that a standard provide means to be integrated easily into existing and future networks as well as address the needs of different applications.

This article is organized as follows. Several required features for a video codec to be used in wireless video applications are extracted. We introduce standard components in H.264/AVC that are relevant for wireless communication. We discuss the application of H.264/AVC for bit rate adaptation and error resilience, respectively. Finally, we conclude the article. For space reasons we have decided to exclude the explanation of all acronyms, the provision of extensive references, and the integration of simulation results from the printed version of the article. A supplemental Web page[1] has been set up addressing these issues.

VIDEO OVER WIRELESS

END-TO-END VIDEO TRANSMISSION

Figure 1 attempts to provide a suitable abstraction level of a video transmission system. In order to keep this article focused, we have excluded capturing and display devices, user interfaces, and security issues; also, most computational complexity issues are ignored.

The video encoder generates data units containing the compressed video stream, possibly stored in an encoder buffer before transmission. A wireless transmission system might delay, lose, or corrupt individual data units. The unavailability of a single data unit usually has significant impact on perceived quality due to spatio-temporal error propagation. In modern wireless system designs, data transmission is usually supplemented by additional information between the sender and the receivers, and within the respective entities. Abstract versions of available messages are included in Fig. 1; specific syntax and semantics as well the exploitation in video

[1] *http://www.nomor.de/H264/wcm.html*

1536-1284/05/$20.00 © 2005 IEEE

transmission systems are discussed in more detail. Furthermore, each processing and transmission step adds some delay, which can be fixed, deterministic, or random. The encoder and decoder buffers allow compensating for variable bit rates produced by the encoder as well as channel delay variations to keep the end-to-end delay constant and maintain the timeline at the decoder. Nevertheless, if the *initial playout delay* is not or cannot be too extensive, late data units are commonly treated to be lost.

Wireless Video Applications

Ideally, high-quality video transmission would require high transmission bit rates, error-free delivery, as well as low and constant channel delays. Obviously, not all of the requests of the video application can be fulfilled; one has to live with the features and limitations of wireless systems as discussed in detail in [2]. Wireless transmission systems provide different transmission modes resulting in different quality of service (QoS) in terms of supported bit rates, bit rate variations, delay variations, as well as reliable delivery. The appropriate selection of transmission modes, adapted to the considered video application, is discussed. Furthermore, in Table 1 we also categorize video applications with respect to their maximum tolerable end-to-end delay, the availability and usefulness of different feedback messages, the availability and accurateness of channel state information at the transmitter, and the possibility of online encoding in contrast to pre-encoded content. Typical Third Generation Partnership Project (3GPP) applications within each category are mentioned. Especially real-time services streaming and conversational, but also broadcast, services provide challenges in wireless environments, as in general reliable delivery cannot be guaranteed. The suitability of H.264/AVC for these services is discussed in the following. For a review of the application standards and used protocols please refer to [2].

H.264/AVC in Wireless Systems

Similar to previous video coding standards, the H.264/AVC standard specifies the *decoder operation* for error-free bitstreams as well as the *syntax* and *semantics* of the bitstream. Consequently, the deployment of H.264/AVC still provides a significant amount of freedom for encoders and decoding of erroneous bitstreams. In the following subsections we introduce the essential features of H.264/AVC for wireless systems, categorized as network integration, compression efficiency, error resilience, and bit rate adaptivity. It is important to understand that most features are general enough to be used for multiple purposes rather than assigned to a specific application.

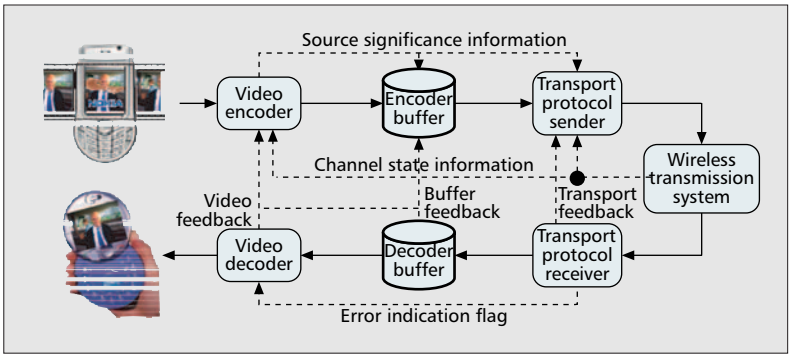


Figure 1. *Abstraction of end-to-end video transmission systems.*

Network Integration

NAL Units — The elementary unit processed by an H.264/AVC codec is called the network abstraction layer (NAL) unit, which can easily be encapsulated into different transport protocols and file formats, such as MPEG-2 transport stream, Real-Time Transfer Protocol (RTP), and MPEG4 file format. There are two types of NAL units, video coding layer (VCL) NAL units and non-VCL NAL units. VCL NAL units contain data that represent the values and samples of video pictures in the form of slices or slice data partitions. One VCL NAL unit type is dedicated ftor a slice in an instantaneous decoding refresh (IDR) picture. A non-VCL NAL unit contains supplemental enhancement information (SEI), parameter sets, picture delimiter, or filler data.

Each NAL unit consists of a one-byte header and the payload byte string. The header indi-

Video application	3GPP	Max. delay	Video/buffer feedback		Transport feedback		CSI	Encoding
			Available?	Useful?	Available?	Useful?	Available?	
Download-and-play	MMS	N/A	No	—	Yes	Yes	—	Offline
On-demand streaming (pre-encoded content)	PSS	≥ 1 s	Yes	Yes	Yes	Yes	Partly	Offline
Live streaming	PSS	≥ 200 ms	Yes	Yes	Partly	Yes	Partly	Online
Multicast	MBMS	≥ 1 s	Limited	Partly	Limited	Partly	Limited	Both
Broadcast	MBMS	≥ 2 s	No	—	No	—	No	Both
Conferencing	PSC	≤ 250 ms	Limited	Yes	No	—	Limited	Online
Telephony	PSC	≤ 200 ms	Yes	Yes	Limited	Yes	Partly	Online

Table 1. *Characteristics of typical wireless video applications.*

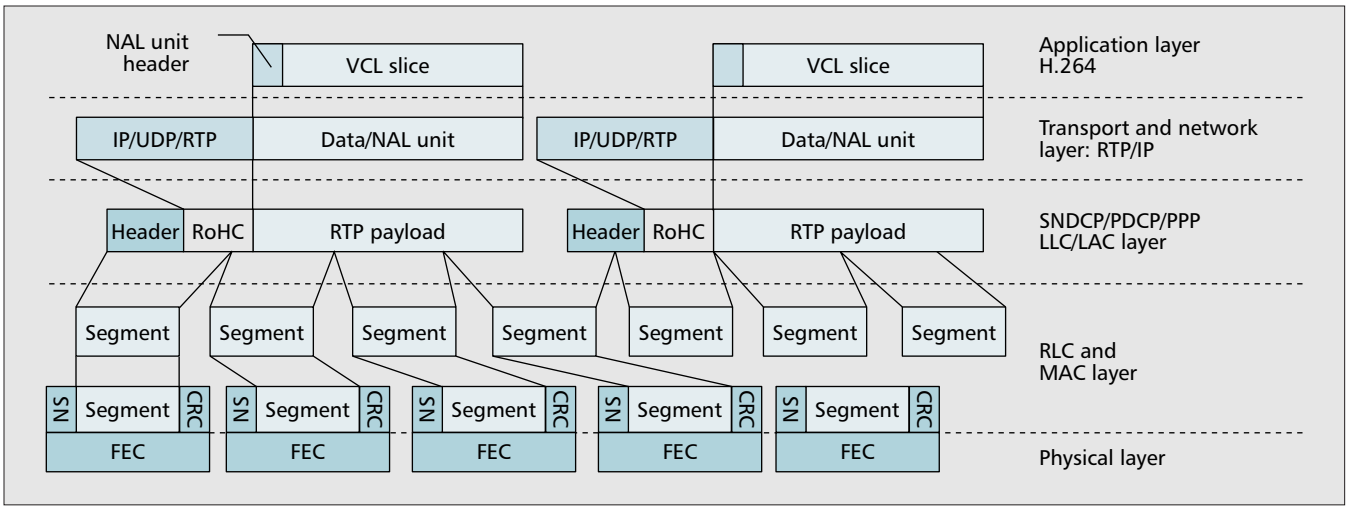


Figure 2. *Integration example of an VCL slice in the RTP payload and 3GPP framework.*

cates the type of NAL unit and whether a VCL NAL unit is part of a reference or non-reference picture. Furthermore, syntax violations in the NAL unit and the relative importance of the NAL unit for the decoding process can be signaled in the NAL unit header.

Parameter Set Concept — H.264/AVC allows sending of sequence and picture level information reliably, asynchronously, and in advance of the media stream that contains the VCL NAL units by the use of parameter sets. Sequence and picture level data are organized into sequence parameter sets (SPS) and picture parameter sets (PPS), respectively. An active SPS remains unchanged throughout a coded video sequence (i.e., until the next IDR picture), and an active PPS remains unchanged within a coded picture. The parameter set structures contain information such as picture size, optional coding modes employed, and macroblock to slice a group map. In order to be able to change picture parameters such as picture size without the need to transmit parameter set updates synchronously to the slice packet stream, the encoder and decoder can maintain a list of more than one SPS and PPS. Each slice header contains a codeword that indicates the SPS and PPS in use.

Integration in RTP and 3GPP Multimedia Services — The integration of multimedia services in 3G wireless systems has been addressed in the recommendations of 3GPP (for details see [2]). In the following we concentrate on packet-based real-time video services. H.264/AVC was lately adopted as a recommended codec for all 3GPP video services. Figure 2 shows the basic processing of a VCL slice within the RTP and 3GPP framework. The slice is packetized in a NAL unit, which itself is encapsulated in RTP/UDP/IP according to [3] and finally transported through the protocol stack of a wireless system such as General Packet Radio Service (GPRS), Enhanced GPRS (EGPRS), Universal Mobile Telecommunications System (UMTS), or code-division multiple access (i.e., CDMA2000). The RTP payload specification supports different packetization modes. In the simplest mode a single NAL unit is transported in a single RTP packet, and the NAL unit header coserves as an RTP payload header. In noninterleaved mode, several NAL units of the same picture can be packetized into the same RTP packet. In interleaved mode several NAL units of potentially different pictures can be packetized into the same RTP packet, and NAL units do not have to be sent in their decoding order. Both the noninterleaved and interleaved modes also allow fragmentation of a single NAL unit into several RTP packets.

In the following we concentrate on UMTS terminology; the corresponding layers for other systems are shown in Fig. 2. After potential RoHC, the generated IP/UDP/RTP packet is encapsulated into a single PDCP packet that becomes an RLC-SDU. As a typical RLC-SDU has a larger size than a RLC-PDU, it is then segmented into smaller units. The length of the RLC-PDU depends on the selected bearer as well as the coding and modulation scheme in use. The RLC layer in wireless systems can operate in unacknowledged mode (UM) and acknowledged mode (AM), both providing RLC-PDU loss detection. However, whereas UM is unidirectional and data delivery is not guaranteed, in AM an automatic repeat request (ARQ) mechanism is used for error correction. The physical layer generally adds forward error correction (FEC) to RLC-PDUs depending on the coding scheme in use so that a constant length channel-coded and modulated block is obtained. This channel-coded block is further processed in the physical layer before it is sent to the far-end receiver. The receiver performs error correction and detection, and possibly requests retransmissions. It is important to understand that in general the detection of a lost segment results in the loss of an entire PDCP packet, so the encapsulated IP and RTP packet as well as the NAL unit is lost.

Compression Efficiency

Compression efficiency is the major attribute for a video codec to be successful in wireless environments. Although the design of the VCL of H.264/AVC basically follows the design of prior

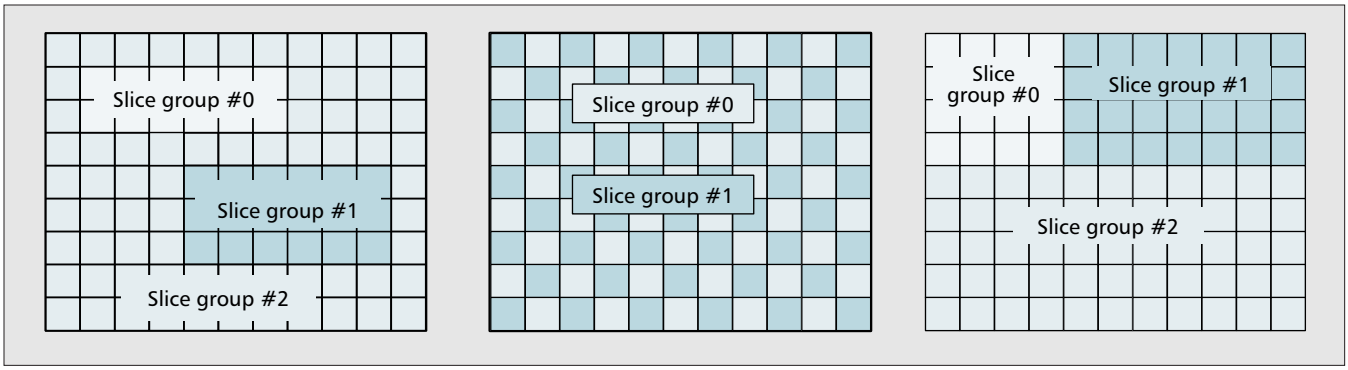


Figure 3. *Macroblock allocation maps: foreground slice groups with one left-over background slice group, checkerboard-like pattern with two slice groups, and sub-pictures within a picture.*

video coding standards, it contains many new details that enable significant improvement in terms of compression efficiency; for details the interested reader is referred to [1, 4]. The gains do not come from a single new technique, but from an ensemble of advanced prediction, quantization, and entropy coding schemes.

The encoder implementation is responsible for appropriately selecting a combination of different encoding parameters, so-called *operational coder control*. When using a standard with a completely specified decoder, parameters in the encoder should be selected such that good rate-distortion performance is achieved. For a video coder like H.264/AVC, the encoder must select parameters, such as motion vectors, macroblock modes, quantization parameters, reference frames, and spatial and temporal resolution, to provide good quality under given rate and delay constraints. Two simplify matters in deciding on good selections of the coding parameters, commonly this task is divided into three levels [5]:

- *Encoder control* performs local decisions, such as the selection of macroblock modes, reference frames, or motion vectors, on the macroblock level and below most appropriately based on a rate-distortion optimized mode selection applying Lagrangian techniques.
- *Rate control* mainly controls the timing and bit rate constraints of the application by adjusting the QP or Lagrange parameter and is usually applied to achieve a constant bit rate (CBR) encoded video suitable for transmission over CBR channels. The aggressiveness of the quantization/Lagrangian parameter change allows a trade-off between quality and instantaneous bit rate characteristic of the video stream.
- *Global parameter selection* selects the appropriate temporal and spatial resolution of the video based on application, profile, and level constraints. Also, packetization modes, like slice sizes, are usually fixed for the entire session. The parameters are mainly determined by general application constraints.

Error Resilience and Bit Rate Adaptivity Features in H.264/AVC VCL

In the following we introduce different error resilience and bit rate adaptivity features included in H.264/AVC VCL with respect to their functionality. For more details we refer to [4, 6, 7, references therein].

Slice Structured Coding — Slices provide spatially distinct resynchronization points within the video data for a single frame. This is accomplished by introducing a slice header, which contains syntactical and semantical resynchronization information. In addition, intra-prediction and motion vector prediction are not allowed over slice boundaries. The encoder can select the location of the synchronization points at any macroblock boundary.

Flexible Macroblock Ordering — FMO allows mapping of macroblocks to slice groups, where a slice group itself may contain several slices. Therefore, macroblocks might be transmitted out of raster scan order in a flexible and efficient way. Some examples of macroblock allocation maps for different applications are shown in Fig. 3. Dispersed macroblock allocations are especially powerful in conjunction with appropriate error concealment (i.e., when the samples of a missing slice are surrounded by many samples of correctly decoded slices).

Arbitrary Slice Ordering — ASO allows that the decoding order of slices within a picture may not follow the constraint that the address of the first macroblock within a slice is monotonically increasing within the NAL unit stream for a picture. This permits, for example, reduction of decoding delay in case of out-of-order delivery of NAL units.

Slice Data Partitioning — In data partitioning mode, each slice can be separated in a header and motion information, intra information, and intertexture information by simply distributing the syntax elements to individual NAL units. Due to this reordering on syntax level, coding efficiency is not reduced, but obviously the loss of individual partitions still results error propagation.

Intra-coding — H.264/AVC distinguishes IDR pictures and regular intra-pictures whereby the latter do not necessarily provide the random access property as pictures before the intra pictures may be used as reference for succeeding predictively coded pictures. H.264/AVC also allows intracoding of single macroblocks for regions

For many use cases it is necessary to adapt the bit rate dynamically in the application to larger bit rates and timescales larger than the initial playout delay allows. In wireless streaming environments bitstream switching provides a simple but powerful means to support bit rate adaptivity.

that cannot be predicted efficiently or due to any other case where the encoder decides for non-predictive mode. The intra mode can be modified such that intra-prediction from predictively coded macroblocks is disallowed. The corresponding constraint intra flag is signaled in the PPS.

Redundant Slices — A redundant coded slice is a coded slice that is a part of a redundant coded picture, which itself is a coded representation of a picture that is not used in the decoding process if the corresponding primary coded picture is correctly decoded. The redundant slice should be coded such that there is no noticeable difference between any area of the decoded primary picture and a decoded redundant picture.

Flexible Reference Frame Concept — H.264/AVC allows reference frames to be selected in a flexible way on a macroblock basis, which provides the possibility to use two weighted reference signals for macroblock interprediction, allows frames to be kept in short-term and long-term memory buffers for future reference, and finally provides temporal scalability. The classical I, P, B frame concept is replaced by a highly flexible and general concept that can be exploited by the encoder for different purposes. However, this concept also requires that not only is the HRD [8] specified in the bitstream domain, but it is also necessary that the encoder be constrained in the amount of frames to be stored in the decoded picture buffer.

Switching Pictures — H264/AVC allows applying mismatch-free predictive coding even where there are different reference signals. So-called primary SP-frames are introduced in the encoded bitstream, which are in general slightly less efficient than regular P-frames but significantly more efficient than regular I-frames. The major benefit results from the fact that this quantized reference signal can be generated mismatch-free using any other prediction signal. If this prediction signal is generated by predictive coding, the frame is referred to as secondary SP-pictures, which are usually significantly less efficient than P-frames as an exact reconstruction is necessary. To also generate this reference signal without any predictive signal, so-called switching-intra (SI) *pictures* can be used. SI pictures are only slightly less inefficient than common I pictures and can also be used for adaptive error resilience purposes. For more details on this unique feature within H.264/AVC the interested reader is referred to [9].

Bit Rate Adaptivity Provision in H.264/AVC

Bit rate adaptivity is one of the most important features for applications in wireless systems to react to the dynamics due to statistical traffic, variable receiving conditions, as well as handovers and random user activity. Due to the applied error control features, these variations mainly result in varying bit rates in different timescales. For applications where online encoding is performed and the encoder has sufficient feedback on the expected bit rate on the channel by some channel state or decoder buffer fullness information, *rate control for VBR channels* can be applied. H.264/AVC obviously supports these features, mainly by the possibility of changing QPs dynamically, but also by the changing temporal resolution.

When channel bit rate fluctuations are not a priori known at the transmitter, or there is no sufficient means or necessity to change the bit rate frequently, *playout buffering* at the receiver can compensate for bit rate fluctuations to some extent. In addition, for anticipated buffer underrun, techniques such as *adaptive media playout* allow a streaming media client, without involvement of the server, to control the rate at which data is consumed by the playout process.

However, these techniques might not be sufficient to compensate for bit rate variations in wireless applications. In this case rate adaptation has to be performed by modifying the encoded bitstream. In today's systems rate adaptation is typically carried out in streaming servers. It is well known that intelligent decisions to drop less important packets rather than dropping random packets — this is treated under the framework of error resilience — can significantly enhance the overall quality. A formalized framework called *rate-distortion optimized packet scheduling* has been introduced [10] and serves as the basis for several subsequent publications. Applying the framework is easiest when important and less important packets are identified in the encoding process. H.264/AVC provides different approaches to support packets with different importance for bit rate adaptivity. First, the temporal scalability features [11] of H.264/AVC relying on the reference frame concept can be used. Second, if frame dropping is not sufficient, one might apply *data partitioning* which can be viewed as a very coarse but efficient method for SNR scalability. Third, flexible macroblock ordering may also be used for prioritization of regions of interest. For example, a background slice group can be dropped in favor of a more important foreground slice group.

For many use cases it is necessary to adapt the bit rate dynamically in the application to larger bit rates and timescales larger than the initial playout delay allows. In wireless streaming environments *bitstream switching* provides a simple but powerful means to support bit rate adaptivity. In this case the streaming server stores the same content encoded with different versions in terms of rate and quality. In addition, each version provides a means to randomly switch into it. IDR pictures provide this feature, but they are generally costly in terms of compression efficiency. The SP-frame concept in H.264/AVC can be used to reduce the loss of compression efficiency in stream switching. In this case the streaming server not only stores different versions of the same content, but also secondary SP pictures as well as SI pictures. When switching streams, the server sends an appropriate secondary SP picture or SI picture [9, 12].

Error Robustness Support Using H.264/AVC

This section discusses the endpoint operation in a wireless H.264/AVC video system. The provided H.264/AVC features can be used exclusively or jointly for error robustness purposes, depending on the application. It is necessary to understand that most codec-level error resilience tools decrease compression efficiency. Therefore, the main goal when transmitting video goes along the spirit of Shannon's famous *separation principle* [13]: Combine compression efficiency with link layer features that completely avoid losses such that the two aspects, compression and transport, can be completely separated. Nevertheless, if errors cannot be avoided, the following system design principles are essential:

- *Loss correction below the codec layer*: Minimize the amount of losses in the wireless channel without completely sacrificing the video bit rate.
- *Error detection*: If errors are unavoidable, detect and localize erroneous video data.
- *Prioritization methods*: If losses are unavoidable, at least minimize loss rates for very important data (e.g., control).
- *Error recovery and concealment*: In case of losses, minimize the visual impact of losses on the actual distorted image.
- *Encoder-decoder mismatch avoidance*: Limit or completely avoid encoder and decoder mismatches resulting in annoying error propagation.

Use cases of the error resilience features for specific applications are discussed.

Error Control Methods — Error control such as FEC and retransmission protocols are the primary tool to provide QoS in mobile systems, especially on the radio access part. QoS methods are essential in good system designs as minimizing or vanishing transmission errors has many advantages for applications. However, usually the trade-offs between reliability vs. delay have to be considered. Nevertheless, to compensate for the shortcomings of non-QoS-controlled networks (e.g., the Internet or some mobile systems) as well as address total blackout periods caused by, say, network buffer overflow or a handover of transmission cells, advanced transport protocols provide features that allow error control to be introduced at the application layer. For example, MBMS services make use of an application-layer FEC scheme. For point-to-point services, selective application layer retransmission schemes can be used to retransmit RTP packets. For many applications it can be assumed that at least a low-bit-rate feedback channel from the receiver to the transmitter exists that allows general back-channel messages to be sent. For example, RTP is accompanied by Real-Time Transport Control Protocol (RTCP) providing control and management messages. Media receivers can send *receiver reports* including instantaneous and cumulative loss rates as well as delay and jitter information. RTCP has recently been extended with the *extended report* packet type, which allows the loss or reception of each RTP packet to be indicated by the receiver to the sender.

Resynchronization and Error Concealment — Despite error control techniques, error resilience in the video is still necessary whenever the video decoder observes residual losses. According to [2], these problems mainly occur in conversational applications due to the delay constraints as well as in multicast/broadcast situations due to the missing feedback link. Slice structured coding typically allows the encoder to choose between two slice coding options, one with a constant number of macroblocks within one slice but arbitrary size of bytes, and one with the slice size bounded to some maximum $S_{\max}$ in bytes, resulting in an arbitrary number of macroblocks per slice. The latter is especially useful to introduce some QoS as commonly the slice size determines loss probability in wireless systems due to the processing shown in Fig. 2.

H.264/AVC decoders should detect losses of slices by keeping a record of which slices of a picture have been received and decoded. Entirely lost reference pictures should be detected based on gaps in the sequence number for reference pictures (the frame_num syntax element of the H.264/AVC bitstream) or prediction from missing pictures in the reference picture buffer (when a bitstream may include subsequences).

As soon as the erroneous macroblocks are detected, *error concealment* for all of them should be invoked. For example, in the H.264/AVC test model software two types of error concealment algorithms have been introduced [7], one exploiting spatial information only, suitable mainly for intra frames, and one exploiting temporal information. It is important to select the appropriate error concealment technique, spatial or temporal, adaptively to obtain a reasonably good visual quality. This selection can concluded from a coded slice (e.g., the macroblock mode information of reliable neighbors), or encoders can assist decoders in the decision-making, say, by including spare picture and scene information SEI messages.

Limitation of Temporal Error Propagation — Despite the presented error control and concealment techniques, packet losses still result in imperfect reconstruction of pictures. Thus, the effects of spatio-temporal error propagation resulting from the motion compensated prediction can be severe. Therefore, the decoder has to be provided with other means that allow error propagation to be reduced or completely stopped.

The most common way to accomplish this task is the reduction of temporal prediction in the encoding process by encoding image regions in intra mode. The straightforward way of inserting IDR frames is quite common for broadcast and streaming applications as these frames are also necessary to randomly access the video sequences. However, when transported over CBR channels, the latency caused by IDR pictures can be undesirable, especially in conversational applications. Therefore, more subtle methods are frequently used to synchronize encoder and decoder reference frames.

In early work it was proposed to introduce intra-coded macroblocks using a constant update pattern, randomly, or adaptively based on a cost function. The selection of an appropriate update ratio depends on different parameters such as

The effects of spatiotemporal error propagation resulting from motion compensated prediction can be severe. Therefore, the decoder has to be provided with other means that allow error propagation to be reduced or completely stopped.

Although the standardization process is finalized, the freedom at the encoder as well as the combination with transport modes such as FEC and retransmission strategies promises optimization potentials.

the sequence characteristics, transmission bit rate, and, most important, channel characteristics. Most suitably, the selection of coding modes can be incorporated in the operational encoder control taking into account the influence of the lossy channel. The encoder control is modified such that the expected decoder distortion is used instead of the encoding distortion. For details on the computation of the expected decoder distortion see [7]. In addition to limiting the error propagation with macroblock intra updates, encoders can also guarantee that macroblock intra updates result in gradual decoding refresh (GDR), that is, entirely correct output pictures after a certain period of time. GDR can be signaled with the recovery point SEI message of H.264/AVC and implemented in H.264/AVC encoders using the isolated regions coding technique [14].

The availability of the feedback channel in conversational and unicast streaming applications has led to different standardization and research activities on interactive error control (IEC) in recent years. If *online encoding* is performed, the slice loss information of the decoder can be directly incorporated in the encoding process to reduce, eliminate, or even completely avoid error propagation. The basics of these methods have been founded under the term *error tracking* [15]. The syntax of H.264/AVC permits incorporating methods for reduced or limited error propagation in a straightforward manner. Similar to operational encoder control for error-prone channels, the delayed decoder state can also be integrated in modified encoder control. By the use of SP-pictures IEC can even be extended to applications with offline encoding.

Prioritization and Data Differentiation — The inherent property of video and multimedia data compared to file download results from the fact that some data might be more important than others. The optimized combination of different video features with different transmission features is also referred to as *cross-layer design*; a similar rate-distortion framework as presented in [10] can be used. Some work in this area shows promising potential to be exploited in future system designs, but it is also important to note that some designs do not provide sufficient gains to replace conventional transmission modes in practical systems. Examples for the provision of different priority modes in end-to-end as well as transmission systems include unequal error protection, unequal erasure protection, selective retransmission, proxies, or different priority queues when accessing shared channels. These systems can be combined with H.264/AVC bit rate adaptivity features, such as different frame types, subsequences, and data partitioning.

Timing constraints, ruling out, for example, retransmissions, only relate to data that is live generated such as conversational video, live streaming or live broadcasting. wherein the sending time of the data is usually closely coupled to the display time, referred to as *time-stamp-based* streaming. If pre-encoded data is transmitted and the decoder buffer is sufficiently large, one can transmit data earlier than its nominal sending time, so-called *ahead-of-time* streaming, which allows better exploitation of the channel. This strategy can be even extended by transmitting more important data earlier, allowing more retransmissions for this important data [12]. H.264/AVC even extends this concept to live encoding by the provision of parameter sets and long-term multiple reference frames.

Summary and Outlook

In this work the benefits of H.264/AVC in wireless transmission environments have been shown. In addition to excellent compression efficiency, H.264/AVC provides features that can be used in one or several application scenarios, and also allows easy integration in most networks. The selection and combination of different features strongly depends on the system and application constraints, namely bit-rates, maximum tolerable playout delays, error characteristics, online encoding possibility, as well as availability of feedback and cross-layer information. Although the standardization process is finalized, the freedom at the encoder as well as combination with transport modes such as FEC and retransmission strategies promise optimization potentials. Therefore, further research in the areas of optimization, cross-layer design, feedback exploitation, and error concealment is necessary to fully understand the potential of H.264/AVC in wireless environments. However, researchers are especially encouraged to integrate transport protocols as well as wireless system options into their considerations rather than assuming QoS-unaware link and transport layers.

Acknowledgments

The authors would like to thank Thomas Wiegand, Stephan Wenger, Ye-Kui Wang, Günther Liebl, and Ingo Viering for useful discussions on the subject of this work.

References

[1] ITU Rec. H.264/ISO IEC 14996-10 AVC, "Advanced Video Coding for Generic Audiovisual Services," 2003.
[2] M. Etoh and T. Yoshimura, "Wireless Video Applications in 3G and Beyond," *IEEE Wireless Commun.*, this issue.
[3] S. Wenger *et al.*, "RTP payload Format for H.264 Video," IETF RFC 3984, Feb. 2005.
[4] G. Sullivan and T. Wiegand, "Video Compression — From Concepts to H.264/AVC Standard," *Proc. IEEE*, Special Issue on Advances in Video Coding and Delivery, vol. 93, no. 1, Jan. 2005, pp. 18–31.
[5] T. Wiegand *et al.*, "Rate-Constrained Coder Control and Comparison of Video Coding Standards," *IEEE Trans. Circuits and Sys. for Video Tech.*, vol. 13, July 2003, pp. 688–703.
[6] S. Wenger, "H.264/AVC over IP," *IEEE Trans. Circuits and Sys.*, vol. 13, no. 7, July 2003, pp. 645–56.
[7] T. Stockhammer, M. M. Hannuksela, and T. Wiegand, "H.264/AVC in Wireless Environments," *IEEE Trans. Circuits and Sys.*, vol. 13, no. 7, July 2003, pp. 657–73.
[8] J. Ribas-Corbera, P. A. Chou, and S. Regunathan, "A Generalized Hypothetical Reference Decoder for H.264/AVC," *IEEE Trans. Circuits and Sys.*, vol. 13, no. 7, July 2003, pp. 674–87.
[9] M. Karczewicz and R. Kurçeren, "The SP and SI Frames Design for H.264/AVC," *IEEE Trans. Circuits and Sys.*, vol. 13, no. 7, July 2003.
[10] P. A. Chou and Z. Miao, "Rate-distortion Optimized Streaming of Packetized Media," *IEEE Trans. Multimedia*, vol. Feb. 2001, http://research.microsoft.com/pachou
[11] D. Tian, M.M. Hannuksela, and M. Gabbouj, "Subsequence Video Coding for Improved Temporal Scalability," *Proc. 2005 IEEE Int'l. Symp. Circuits and Sys.*, May 2005.

[12] T. Stockhammer, M. Walter, and G. Liebl, "Optimized H.264-Based Bitstream Switching for Wireless Video Streaming," *Proc. ICME*, July 2005, Amsterdam, The Netherlands.
[13] C. E. Shannon, *The Mathematical Theory of Communication*, University of Illinois Press, 1948.
[14] M. M. Hannuksela, Y.-K. Wang, and M. Gabbouj, "Isolated Regions in Video Coding," *IEEE Trans. Multimedia*, vol. 6, no. 2, Apr. 2004, pp. 250–67.
[15] B. Girod and N. Färber, "Feedback-Based Error Control for Mobile Video Transmission," *Proc. IEEE*, vol. 97, no. 10, Oct. 1999, pp. 1707–23.

BIOGRAPHIES

MISKA M. HANNUKSELA (miska.hannuksela@nokia.com) is a research manager in the Multimedia Technologies Laboratory, Nokia Research Center. He has been an active participant in the ITU-T Video Coding Experts Group since 1999 and in the Joint Video Team of ITU-T and ISO/IEC since its foundation in 2001. He has co-authored more than 80 technical contributions to these standardization groups. His research interests include video error resilience and video communication systems.

THOMAS STOCKHAMMER (stockhammer@nomor.de) worked at the Munich University of Technology, Germany, and has been a visiting researcher at Rensselear Polytechnic Institute, Troy, New York, and the University of California at San Diego (UCSD). He has published more than 60 conference and journal papers and holds several patents. He regularly participates and contributes to different standardization activities, such as JVT, IETF, and 3GPP and has co-authored more than 70 technical contributions. He is co-founder of Novel Mobile Radio (NoMoR) Research. Since 2004 he has been working as a research and development consultant for Siemens Mobile Devices. His research interests include video transmission, cross-layer and system design, rate-distortion optimization, information theory, and mobile communications.

[P9] V. K. Malamal Vadakital, M. M. Hannuksela, M. Rezaei, and M. Gabbouj, “Method for unequal error protection in DVB-H for mobile television,” *Proceedings of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Sep. 2006.

The 17th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'06)

METHOD FOR UNEQUAL ERROR PROTECTION IN DVB-H FOR MOBILE TELEVISION

Vinod Kumar Malamal Vadakital
Tampere University of Technology
Tampere, Finland

Miska M. Hannuksela
Nokia Research Center
Tampere, Finland

Mehdi Rezaei
Tampere University of Technology
Tampere, Finland

Moncef Gabbouj
Tampere University of Technology
Tampere, Finland

ABSTRACT

This paper introduces a method for unequal error protection (UEP) of media data in a time-sliced DVB-H channel. Media datagrams are assigned priorities using some a-priori knowledge. Datagrams covering a certain period of playback time are first grouped based on the priority assignment. Each group is then protected using Reed–Solomon forward error correction (FEC) codes and packed into multi-protocol encapsulation (MPE) FEC frames as defined by the DVB-H standard. All MPE-FEC frames for a certain period of playback time are then sent back to back without any delay between these MPE-FEC frames. This method of UEP is generic and can be tuned according to the priority assignment algorithm. Simulations using H.264/AVC video were conducted to evaluate the performance of the proposed method. It used a simple priority assignment algorithm. The resulting rate distortion graphs show good performance and an average luma peak signal-to-noise ratio (PSNR) improvement of up to 0.8 dB was achieved.

I. INTRODUCTION

Point-to-multipoint (PTM) wireless data communication using digital television broadcast networks has seen significant scientific interest in recent times. Broadcast networks provide a high bandwidth asymmetric communication link to a large audience. This technology is an excellent alternative for PTM communication when interactivity is not a priority and the target audience is large.

To enable data communication to handheld wireless mobile terminals the Digital Video Broadcasting (DVB) organization has defined a new standard: Digital Video Broadcasting-Handhelds (DVB-H) [1]. DVB-H is very closely based on Digital Video Broadcasting-Terrestrial (DVB-T) [2] and is backward compatible with it. DVB-T was found to be useful for mobile data communications. However, it was found to be insufficient to meet the harsh demands imposed by small handheld, battery-operated devices. Mobile handheld devices have strict power constraints and are more demanding with respect to data robustness when transmitted over highly error-prone radio channels.

To address these drawbacks, the DVB-H working group was formed, and in July 2004, released the final draft of the DVB-H standard. DVB-H uses the same basic concepts of DVB-T but adds additional link-layer features to solve the power constraint and robustness problems. The concept of time-slicing was introduced, reducing the average power consumption of a hand-held mobile terminal by as much as 90-95%. An optional enhancement using Reed-Solomon forward error correction (FEC) codes encapsulated into Multi-protocol encapsulated sections (MPE-FEC) was also introduced to provide added robustness required for hand-held mobile terminals. MPE-FEC improves the carrier-to noise (C/N) and Doppler performance in the DVB-H channel while also providing improved tolerance to impulse interference.

Although MPE-FEC provides the much needed data robustness in wireless channels, under very erroneous conditions it fails. Using a-priori knowledge of the transmitted media and tuning the way MPE-FEC is applied across the media datagrams can provide better robustness.

Unequal error protection (UEP) is such a scheme that uses a-priori knowledge of the media to differentially protect data using FEC. UEP has been used for visual coding in ways which can be coarsely grouped into three classes. The first class uses methods to protect important parts of coded data like headers, motion vectors and DCT coefficients more than other data as in [3]. The second class uses differential protection to layered coding as in [4]. The third class use bias based on the importance of different types of coded pictures in the overall presentation of the video content as in [5]. UEP has also been used for audio where coded frames and coded sample data are unequally protected as in [6].

Although many UEP schemes exist, none deal with a time-sliced DVB-H channel. The UEP scheme in [7] deals with the 3GPP's multimedia broadcast/multicast service (MBMS) channel which is characteristically different from a DVB-H channel. This paper presents a novel method of UEP in a time sliced DVB-H channel. It uses intelligent manipulation of DVB-H time slicing parameters and smart grouping of datagrams into priorities. The method is validated using simulations that show the comparative performance with and without its use.

The rest of the paper is organised as follows. Section II briefly discusses the causes of wireless channel errors and techniques currently used to provide better reliability in wireless channel. Forward error correcting codes (FEC) are introduced in Section III and Section IV briefly introduces the concept of priorities in multimedia contents. DVB-H with its enhancements is described in Section V. Section VI details the new method for UEP in DVB-H. A simple simulation conducted to verify the use of UEP and its advantages is given in Section VII followed by the conclusions in Section VIII.

II. WIRELESS CHANNEL ERRORS

Erroneous data transmission over current generation networks is considered a part of the system's normal operation. Depending on the mechanisms and medium of transmission the causes of errors vary. Wireless transmission errors occur due to physical radio channel characteristics like physical barriers in the environment, interference from other

1-4244-0330-8/06/$20.00©2006 IEEE

signals in the transmission area, multi-path propagation errors, and fading errors.

Among methods to provide reliability in wireless channels, two methods stand out. The first of these is protection using forward error correction (FEC) codes. Here additional parity information is added to the actual data which can then be used to correct data errors when they are detected. The second method is to use automatic repeat request (ARQ) techniques. Using this technique, when a receiver encounters a data loss, it requests the sender to resend the lost data. The server upon receiving such a request makes an identical copy of the lost data and resends it to the receiver.

ARQ techniques are effective when there is a feedback channel available. However, ARQ techniques can increase the transmission delay within the system due to request-response turn-around time. Furthermore, in PTM type transmission, using ARQ techniques is a challenge due to the possibility of request implosion at the sender. The implosion problem occurs when channel conditions for many receivers are bad and all these receivers simultaneously send resend-requests to the sender.

Some transmission channels, like DVB by its own, do not have a feedback channel. Furthermore, in a wireless PTM type channel, different receivers can experience different error conditions based on its location, interference from other signals, and due to receiver motion. FEC techniques have an advantage of correcting varying error conditions experienced by different receivers using the same codes for all the receivers. These reasons make FEC a preferred error resiliency mechanism for PTM type communications.

III. Forward Error Correction

FEC codes transform some number of equal length k symbols into n symbols, where $n > k$, by adding $(n-k)$ additional symbols, called parity symbols. Ideally, an FEC code can reconstruct any k lost/corrupted symbols of the n symbols. This property is called Maximum Distance Separable (MDS) property and most practical FEC systems are bounded by this property. The Reed-Solomon (RS) FEC code is a good example of an FEC code that follows MDS property and is used by DVB-H.

FEC techniques, in general, can be classified into two types based on how it is applied to the data. The first is called media-unaware FEC. Media-unaware FECs are generic algorithms which work independent of the media it is protecting. While they do provide a good recovery mechanism, their bandwidth efficiency is usually suboptimal. The second technique is called media-aware FEC. It takes advantage of the knowledge about the type of media it is protecting and its importance in the overall quality of the transmitted media. However, media-aware FECs are tailored to specific media formats and incompatibility problems arise quite frequently.

Errors in wireless channels occur as clusters of bursts rather than isolated errors. Under heavy channel error conditions FEC cannot always recover the lost data. To minimize the probability of long burst errors, data interleaving is commonly used. Interleaving transforms long burst errors into smaller burst or isolated errors.

IV. Priority Assignment

Different components and data parts of multimedia contents contribute unequally to the perception quality of the multimedia content. For example, audio is considered subjectively more important than video in audio-visual contents like news and documentaries. Delving further into coding specifics, in hybrid video coding algorithms, such as MPEG-1, MPEG-2, MPEG-4 and H.264/AVC, picture frames used for reference (I, P pictures) is more important than picture frame that are not used for reference (B picture for MPEG-X and p non-reference pictures for H.264/AVC). Going even deeper into details of hybrid video coding algorithms, header data, motion vector information, and certain DCT coefficients are considered more important than other coded data. For audio, some audio coding schemes require the presence of codebook information before playback of the content can start, and here the packets carrying the codebook have a higher priority than the content packets.

Based on the importance of components or data-parts of a multimedia presentation (known a-priori), priority assignments can made. For example, audio can be assigned a higher priority than video (content dependent) and in video a reference picture can be allocated higher priority than non-reference picture. Based on these priorities, error resiliency methods such as FEC can be applied unequally: higher priority getting higher protection than lower priority data.

V. DVB-H

Broadcasting of data other than broadband digital television in DVB is called Data casting [8]. Data casting in DVB uses one of the six different profiles, each of them catering to applications with different requirements. These profiles are (a) Data piping (b) Data streaming (c) Multi-Protocol Encapsulation (MPE) (d) Data Carousals (e) Object Carousals and (f) Higher protocols based on asynchronous data streams.

DVB-H is designed to carry Internet Protocol (IP) [9] based data traffic. IP, being an addressable protocol, has special requirement different from what the DVB system was initially designed for. Hence, for seamless integration between the DVB and IP worlds, an intermediary translator capable of understanding both IP and DVB protocols is required. The MPE profile performs this job of protocol translation. Private data Digital Storage Media – Command and Control (DSM-CC) specification is used to encapsulate the OSI layer 3 datagrams into MPE sections. The MPE sections are then mapped onto MPEG-2 system layer transport stream (TS) packets [10].

A. MPE-FEC

MPE-FEC was included in DVB-H to improve the unfavourable carrier-to-noise (C/N) conditions typical of a wireless radio channel. It is an optional multiplexer-layer FEC code based on Reed-Solomon (RS) codes. MPE-FEC is

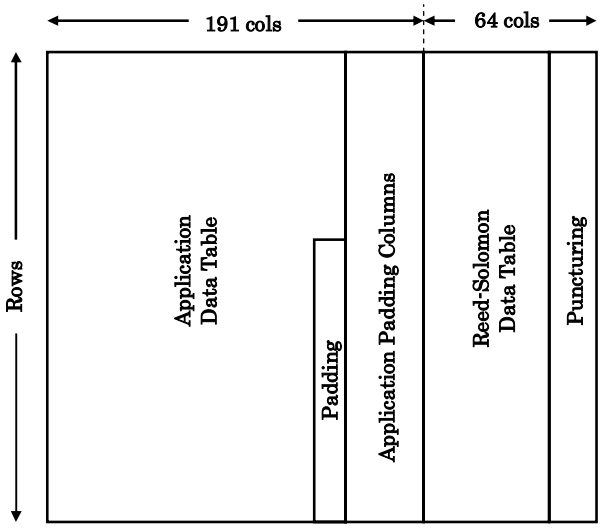


Figure 1: The MPE-FEC matrix structure.

computed over IP packets and encapsulated into MPE sections. MPE-FEC sections are transmitted such that an MPE-FEC ignorant receiver would have no problems receiving just the unprotected data.

To compute MPE-FEC, data (IP packets) are filled into an N x 191 matrix where each cell of the matrix hosts one byte of information and N denotes the number of rows in the matrix. The standard defines the value of N to be one of 256, 512, 768 or 1024. The datagrams are filled into the matrix column-wise. RS codes are computed for each row and concatenated such that the final size of the matrix is of size N x 255. The N x 191 part of the matrix is called the Application data table (ADT) and the next N x 64 part of the matrix is called the RS data table (RSDT). For rate-control and disallowing of IP packet fragmentation between two MPE-FEC frames in the standard, the ADT need not be completely filled. This unfilled part of the ADT is called padding. To control channel code-rate all 64 columns of RSDT need not be transmitted and the un-transmitted RSDT columns is called puncturing. The structure of an MPE-FEC matrix is shown in Figure 1 and further detailed information on the MPE-FEC matrix construction can be obtained from [8].

B. Time-slicing for power saving

Battery-operated mobile devices have a limited source of power. The power consumed in receiving, decoding and demodulating a standard full-bandwidth DVB-T signal would use up substantial amount of battery life in a short time. Time slicing of the MPE-FEC frames is used to solve this problem [11]. The data is received in bursts so that the receiver, utilizing control signals, remains inactive when no bursts are to be received. The bursts are sent at a significantly higher bit-rate compared to bit-rate when conventional bit rate management is used.

Time-slicing in DVB-H uses the delta-t method to signal the start of the next burst. Timing information delivered using delta-t method is relative. In other words, the delta-t time is the difference between the current time and the start of the next burst. The use of delta-t method to signal, removes the need for synchronization between the transmitter and receiver. Its use also provides flexibility since parameters such as burst size, burst duration, burst bandwidth and the off-times can be freely varied. Figure 2 shows two time-sliced bursts and parameters that define time-sliced bursts.

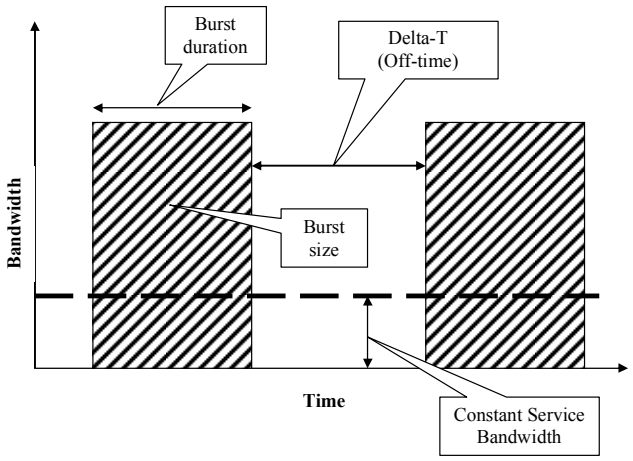


Figure 2: Time-slicing in DVB-H.

VI. METHOD FOR UEP IN DVB-H

A new method for UEP using MPE-FEC and time-slicing is described in this section. In this paper, a service is defined as a multiplex of multimedia packets that are of relevance to a receiver. For example, a multiplex of a video stream, the associated audio stream, and subtitling information relevant to a receiver can be considered to be a service.

The datagrams of a service are assigned priorities either manually or automatically using some a-priori knowledge. Taking a news broadcasting service as an e.g., the audio can have a higher priority than video which in turn has a higher priority than auxiliary media enhancement data like subtitle text. Continuing with the same example, further priority assignment can be made in the video bit stream such that reference picture datagrams such as datagrams of an Intra frame and referenced Inter picture can be assigned higher priority than datagrams of non-referenced Intra pictures. This priority labelling procedure can either be done at the IP Encapsulator or external to it.

The multiplexed media datagrams corresponding to certain duration (either in terms of decoding or output timestamps) are encapsulated into two or more MPE-FEC matrix according to their priority label. These MPE-FEC matrices are referred to as peer MPE-FEC matrices. The number of peer MPE-FEC matrices in a time-sliced burst is equal to the number of unique priority labels assigned to the datagrams. For example, if there are K priority labelled datagrams that can be transmitted in one time-sliced burst and there are P different priorities associated with the media datagrams of the service, then each of the K datagrams will be a part of one of P MPE-FEC peer matrices in the burst.

To construct the peer MPE-FEC matrices in a time-sliced burst, the datagrams are grouped using their priority labels.

The grouping procedure is performed on all the datagrams that go into the time-sliced burst. The grouped datagrams are arranged in ascending order such that the datagrams with the lowest priority comes first in the transmission order and the datagrams with the next higher priority comes next and continuing so forth until the datagram group that has the highest priority comes last in the transmission order. This fact holds true for all time-sliced bursts of a service. For example, K datagrams with L unique priority labels among them where $P = \{p_1, p_2, p_3, \ldots p_L\}$ is the set of L different priority labels, and where the priority ranking is such that $p_L < p_{(L-1)} < \ldots < p_2 < p_1$, then all datagrams are grouped in such a way that all datagrams with a priority label of p_L comes before all datagrams with priority label $p_{(L-1)}$ and continuing similarly until all datagrams with priority p_1 comes last.

The peer MPE-FEC matrices are transmitted back-to-back, i.e. there is no transmission delay or interval between the peer MPE-FEC matrices. One way to implement this is to consider delta-t between peer MPE-FEC matrices being equal to zero. This fact holds true for all time-slices burst of the service. The peer MPE-FEC matrices are arranged in ascending priority order, i.e. the lowest priority MPE-FEC matrix is sent first and the highest priority matrix is sent last. If a receiver starts the reception of the stream in the middle of the period when a certain set of peer MPE-FEC matrices are sent, it is likely that the receiver will receive at least the highest priority MPE-FEC matrix.

The choice of RSDT columns for all the MPE-FEC matrices in all the time-sliced bursts in the service should be such that the average service bit rate when using this method shall not overshoot the maximum allowed service bit rate. The MPE section headers for all sections in the peer MPE-FEC matrices other than the peer MPE-FEC matrix that contains the highest priority datagrams, sets delta-t value in

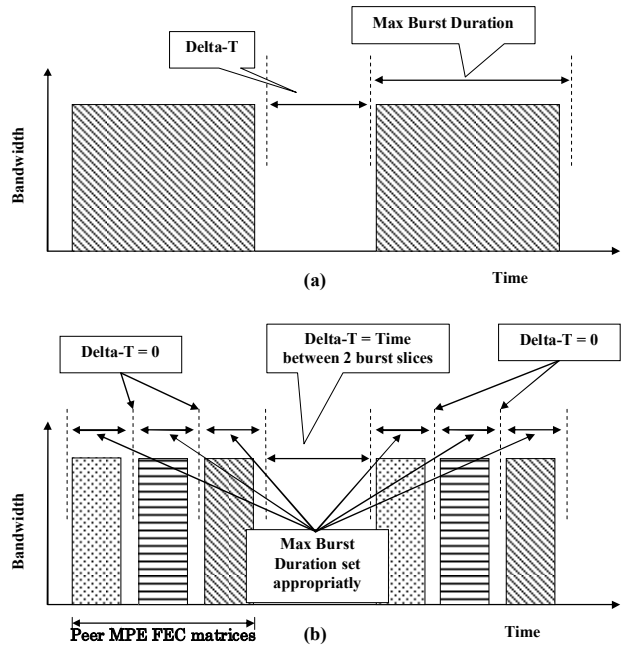


Figure 3: MPE-FEC matrix construction and transmission (a) Without UEP (b) With UEP.

their section headers to zero. Similarly, the MPE section headers of all sections in the peer MPE-FEC matrices set the maximum_burst_duration field as the maximum duration of the peer MPE-FEC matrix reception. The delta-t value in the MPE section headers of MPE-FEC matrix that consists of the datagrams with the highest priority is set to the time when the next time-sliced burst for the service starts. Figure 3 illustrates the method for construction of MPE-FEC matrix in the non UEP case and the UEP case.

VII. SIMULATIONS

Simulations were conducted to verify the performance of the UEP method described in this paper using a simple priority assignment algorithm. For simplifying simulations, a service consisted of a single H.264/AVC coded video stream.

A. Simulation environment

The simulations were carried out for two well known video sequences: Paris and Silent. Since the original lengths of these sequences were short to get any reliable statistics, the sequences were made longer by end to end concatenation. The final length of the video sequences after concatenation was 3000 picture frames. The videos were coded at 15 frames per second using a baseline H.264/AVC encoder. MPEG-2 TS error pattern with an error probability of 0.09 approximating a TU6 channel was created. The number of MPE-FEC rows was set to 512. Two priorities, the highest given to reference pictures (IDR and reference P pictures) and the lowest priority to non-reference p pictures was used. IDR frequency was set to 120 picture frames. The maximum slice-size was set to 1000 bytes and IPV6 was assumed for all simulations.

For PSNR comparison, a simple error-concealment algorithm was used. Lost slices were replaced in the decoder picture buffer (DPB) by a copy of the same slice in the previous picture in presentation order. When an entire picture is lost, a copy of the previous picture in presentation order is used as concealment for the lost picture in DPB.

B. Simulation method

For comparative statistics, simulations using UEP and without UEP was carried out. When using the non-UEP mode, 3/4 code-rate was used i.e. 191 ADT columns and 64 RSDT columns. The following algorithm was used iteratively for the entire coded bit stream.

1. Fill an MPE-FEC matrix at 3/4 code-rate. This would give a non-UEP MPE-FEC matrix.
2. In this non-UEP MPE-FEC frame identify the reference and non-reference frames and their respective IP packets.
3. Separate the reference and non-reference IP packets and form two peer MPE-FEC matrices one for the highest priority datagrams and the other for the lowest priority datagrams.
4. Protect each peer matrix with the appropriate RSDT columns. In our case the code rate was set to 95% of 64 columns for priority-1 peer MPE-FEC matrix and

5% of 64 columns to priority-2 MPE-FEC peer matrix.

It should be noted that by using the above algorithm, channel bit rates for both the UEP case and the non-UEP case is held approximately constant with extremely minuscule differences between the two bitrates.

C. *Simulation results and analysis*

The graphs plotted in Figure 4 and Figure 5 show the rate distortion for the two sequences simulated. It shows that the use of UEP using the priority assignment algorithm as discussed in this paper outperforms the non UEP based approach especially at higher bitrates.

The plots can be explained as follows: when UEP is used, the reference frames are better protected than non-reference frames. Therefore, a reference picture loss probability is lesser than the non-reference picture loss probability. When reference pictures are lost, prediction errors cause subsequent pictures in decoding order to be decoded incorrectly. This is the case even if the coded data for the subsequent pictures are received correctly. The duration of this error propagation depends on the Intra picture insertion interval. On the other hand when non-reference pictures are lost, error propagation does not occur as the picture is not used as a reference for any other picture in the coded sequence. This results in only the lost non-reference picture being decoded and presented incorrectly. Perceptually a non-reference picture loss is sensed as a jerk in moving video, while the loss of reference picture is perceived as garbled pictures for as long as the error propagation occurs.

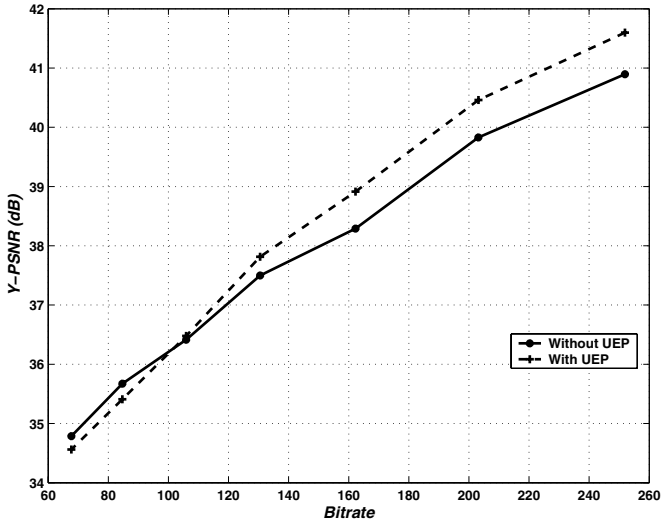


Figure 4: Rate distortion plot for Silent.

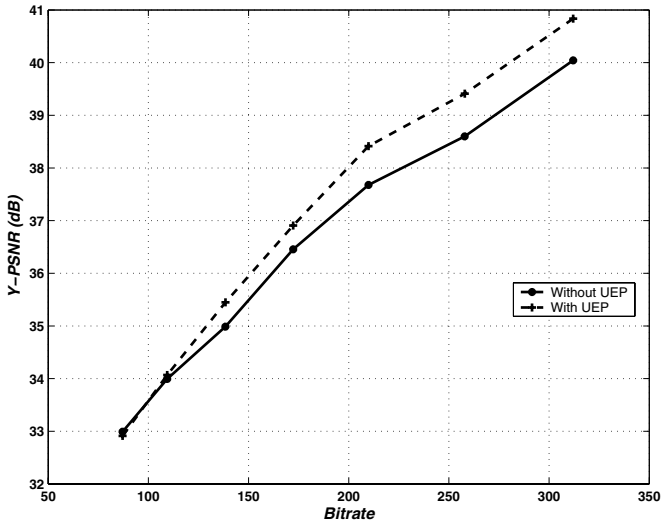


Figure 5: Rate distortion plot for Paris.

VIII. CONCLUSIONS

This paper described a method to provide unequal error protection (UEP) to media data transmitted over a DVB-H channel. The method uses priority labelling and grouping of labelled datagrams to protect data using forward error correction codes packed into multi-protocol encapsulation sections (MPE-FEC). Simulation results using a simple algorithm for priority assignment for an H.264/AVC encoded bit stream show that average luma PSNR improvements of up to 0.8 dB can be achieved. This method of UEP is generic and can be used for any type of media that contains data parts of unequal importance as long as an appropriate priority assignment algorithm is used.

REFERENCES

[1] ETSI, "Digital Video Broadcasting (DVB): Transmission systems for handheld terminals," ETSI standard, EN 302 304 V1.1.1, 2004.

[2] ETSI, "Digital Video Broadcasting (DVB): Framing structure, channel coding and modulation for digital terrestrial television." ETSI standard, EN 300 744, 2001.

[3] J. T. H. Chung-How; D. R. Bull, "Loss Resilient H.263+ Video over the Internet", Signal Processing, Image Commun., vol. 16, pp. 891-908, 2004.

[4] M. Gallant; F. Kossentini, "Rate-distortion optimized layered coding with unequal error protection for robust Internet video", Circuits and Systems for Video Technology, Volume 11, Issue 3, March 2001 pp 357 – 372.

[5] F. Marx and J. Farah, "A Novel Approach to Achieve Unequal Error Protection for Video Transmission over 3G Wireless Networks", Signal Processing, Image Commun., vol. 19, pp. 313-323, 2004.

[6] Chi Wai Yung; Hung Fai Fu; Chi Ying Tsui; R.S. Cheng; D. George; "Unequal error protection for wireless transmission of MPEG audio", ISCAS, June 1999, vol.6, pp 342 – 345

[7] D. Tian., M.V. Vinod, M. Hannuksela, S. Wenger, M. Gabbouj, "Improved H.264/AVC video broadcast/multicast," Visual Communications and Image Processing 2005, VCIP 2005, Beijing, China, Proceedings of SPIE, pp. 71-82, July 2005.

[8] ETSI, "Digital Video Broadcasting (DVB): Specification for Data Broadcasting," ETSI standard, EN 301 192, V1.3.2 2003.

[9] J. Postel, "RFC 791: Internet protocol," Sept. 1981.

[10] ISO/IEC 13818-1: "Information technology – Generic coding of moving pictures and associated audio information: Systems," Nov. 1994.

[11] J. Aaltonen, H. Pekonen, T. Auranen, K. Laiho, P. Talmola "Power saving considerations in mobile datacasting terminals," IEEE International Symposium on Consumer Electronics 2002.

[P10] M. M. Hannuksela, V. K. Malamal Vadakital, and S. Jumisko-Pyykkö, “Comparison of error protection methods for audio-video broadcast over DVB-H,” *EURASIP Journal on Advances in Signal Processing*, doi:10.1155/2007/71801, 2007.

Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2007, Article ID 71801, 12 pages
doi:10.1155/2007/71801

Research Article

Comparison of Error Protection Methods for Audio-Video Broadcast over DVB-H

Miska M. Hannuksela,[1] Vinod Kumar Malamal Vadakital,[2] and Satu Jumisko-Pyykkö[3]

[1] *Nokia Research Center, P.O. Box 1000, 33721 Tampere, Finland*
[2] *Institute of Signal Processing, Tampere University of Technology, P.O. Box 553, 33101 Tampere, Finland*
[3] *Institute of Human-Centered Technology, Tampere University of Technology, P.O. Box 553, 33101 Tampere, Finland*

Received 1 September 2006; Revised 21 February 2007; Accepted 16 April 2007

Recommended by Anthony Vetro

The paper discusses methods for robust audio-video broadcast over the digital video broadcasting-handheld (DVB-H) system. DVB-H includes a link-layer forward error correction (FEC) scheme known as multiprotocol encapsulation (MPE) FEC, which provides equal error protection (EEP) to the transmitted media streams. Several approaches for unequal error protection (UEP) have been proposed in the literature, and the applicability of some of them to DVB-H is analyzed in the paper. A link-layer UEP method based on priority segmentation of the media streams is chosen for more detailed analysis. According to the method, audio and the most important coded video pictures are protected by MPE-FEC more robustly compared to the remaining coded pictures. In order to compare EEP and UEP in a DVB-H environment, an error-prone DVB-H channel was simulated, audio-visual clips were sent through it, and a comprehensive subjective quality evaluation was conducted in a controlled laboratory environment. The results of the subjective evaluation revealed that the use of UEP improves the subjective quality of some test clips noticeably when the channel conditions were severe, while in other tested channel conditions and clips, UEP and EEP performed equally well.

1. INTRODUCTION

Mobile television services are expected to gain popularity in the next few years. Digital video broadcasting-handhelds (DVB-H) [1] is among the most used technical solutions for providing low interactivity, mass mobile television services. DVB-H is downward compatible with the DVB-Terrestrial (DVB-T) standard [2], thus enabling it to reuse the same network infrastructure as well as radio frequencies as used by DVB-T. The elementary transmission unit for DVB-H is a 188-byte MPEG-2 transport stream (TS) packet, specified in the MPEG-2 systems specification [3]. In contrast to DVB-T, where usually audio-video elementary streams were directly packetized to MPEG-2 TS packets, DVB-H is primarily designed for carriage of Internet protocol (IP) datagrams. In order to maintain compatibility with DVB-T, IP datagrams are packetized to multi-protocol encapsulation (MPE) sections as specified in [4], which are then carried over MPEG-2 TS packets.

FEC codes transform some number of equal-length k symbols into n symbols, where $n > k$, by adding $(n - k)$ additional symbols, called parity or repair symbols. Ideally, an FEC code can reconstruct any $(n - k)$ corrupted symbols of the n symbols, when the location of errors is known and $(n-k)/2$ corrupted symbols when the location is not known. This property is called maximum distance separable (MDS) property and most practical FEC systems are bounded by this property. The Reed-Solomon (RS) FEC code [5] is a good example of an FEC code that follows MDS property and is used by DVB-H. Errors in wireless channels typically occur as clusters of bursts rather than isolated errors. Therefore, applications that can endure the longer latency time required for FEC computing are better suited to use the DVB-H transmission.

DVB-H adds additional link-layer features to solve the power constraint and robustness problems associated with handheld mobile terminals. The concept of time-slicing was introduced, reducing the average power consumption of a hand-held mobile terminal by as much as 90–95%. An optional enhancement using Reed-Solomon forward error correction (FEC) codes encapsulated into multiprotocol encapsulated sections (MPE-FEC) was also introduced to provide added error robustness required for hand-held mobile terminals.

Even though DVB-H can convey any IP datagrams, the audio and video codecs for IP-based broadcasting are specified in [6] to facilitate interoperability of DVB-H service providers and receivers. The high efficiency advanced audio coding version 2 (HE AAC v2) [7] is recommended for audio compression, and advanced video coding (H.264/AVC) [8] is recommended for video compression. A number of profiles are specified in H.264/AVC. A profile consists of a subset of the algorithmic features or coding tools of the standard and a set of constraints on those features. A profile is typically targeted for a family of applications sharing similar trade-off between memory, processing, latency, and error resiliency requirements. Decoders conforming to a profile must support all the features of a profile. Five IP integrated receiver-decoder (IRD) capabilities are specified in [6] to facilitate service tailoring for different types of terminals. IP-IRD capabilities for battery-powered devices require the support of H.264/AVC baseline profile with the constraint_set1_flag syntax element of H.264/AVC being equal to 1, which is also referred to as the constrained baseline profile.

Unequal error protection (UEP) takes advantage of the fact that different portions of the coded bit stream have different levels of importance to the overall subjective quality of the presentation. UEP aims at providing graceful degradation of subjective quality under harsh transmission conditions and hence the overall quality of all recipients in any transmission conditions is expected to improve in comparison to the quality obtained with equal error protection (EEP). When applied to coded video, UEP requires that video bit streams be partitioned to segments of different priorities according to the segments' impact to subjective quality. Segments are then protected with unequal amount of FEC repair data. The priority partitioning methods can be roughly categorized into data partitioning, region-of-interest prioritization, spatial, quality, and temporal layering.

This paper uses only temporal layering for priority assignment. This is because the goal of the design was to maintain H.264/AVC constrained baseline profile compatibility and using other types of priority partitioning would have required more advanced H.264/AVC profiles support or the scalable extension of H.264/AVC (under development). Temporal layering refers to the encoding of a temporally scalable bit stream. Any bit stream can be partitioned into two temporal layers, one that contains the intra pictures only, and another containing the remaining ones. Many video coding schemes enable nonreference pictures, which are not used for inter prediction of any other picture. Modern video coding standards such as H.264/AVC also enable hierarchical temporal scalability, in which subsequences of coded pictures, including also reference pictures, can be removed from a bit stream without affecting the decoding of the remaining bit stream. It has been shown that temporal scalability improves compression efficiency [9] even with the constrained baseline profile of H.264/AVC, which does not include bi-predictive pictures (also known as B pictures).

In this paper, we analyze which methods for UEP can be applied to DVB-H in a straightforward manner without substantial changes in the system. In addition, we compare the UEP method that we found the most applicable with the EEP scheme provided by MPE-FEC in different radio conditions.

The rest of this paper is organized as follows. Section 2 reviews the DVB-H protocols and system to an extent that is necessary for understanding of this paper. Section 3 provides an overview of those features of H.264/AVC and its packetization format for real-time transport protocol (RTP) that are essential for the presented UEP method. A brief review of some UEP methods is provided in Section 4 and their applicability to DVB-H is analyzed. Furthermore, one of the reviewed UEP methods is presented in more details in Section 4. The operation of the conventional MPE-FEC-based EEP method and the presented UEP method was simulated in a DVB-H environment and the resulting audio-visual test clips underwent a subjective viewing test. The simulation and test setup is presented in Section 5, and the results are analyzed in Section 6. Finally, Section 7 concludes the paper.

2. OVERVIEW OF DVB-H PROTOCOLS AND SYSTEM

This section introduces the fundamentals of DVB-H and is organized as follows. Section 2.1 presents the protocol stack of DVB-H. The FEC coding of DVB-H is reviewed in Section 2.2. Finally, the method for time-slicing is explained in Section 2.3.

2.1. DVB-H protocol stack

The protocol stack for DVB-H is presented in Figure 1. IP packets are encapsulated to MPE sections for transmission over DVB protocols in the medium access (MAC) sublayer. Each MPE section consists of a header, the IP datagram as a payload, and a 32- byte cyclic redundancy check (CRC) for the verification of payload integrity. The MPE section header contains addressing data among other things. The MPE sections can be logically arranged to application data tables in the logical link control (LLC) sub-layer, over which RS FEC codes are calculated and MPE-FEC sections are formed. The process for MPE-FEC construction is explained in more detail in Section 2.2. The MPE and MPE-FEC sections are mapped onto MPEG-2 TS packets.

2.2. MPE-FEC

MPE-FEC was included in DVB-H to combat long burst errors that cannot be efficiently corrected in the physical layer. MPE-FEC is based on the Reed-Solomon FEC coding. Since Reed-Solomon code is a systematic code, that is, the source data remains unchanged after FEC encoding, MPE-FEC decoding is made optional for DVB-H receivers. MPE-FEC is computed over IP packets and encapsulated into MPE sections. MPE-FEC sections are transmitted such that an MPE-FEC ignorant receiver could just receive the unprotected data while ignoring the protection data that follows.

To compute MPE-FEC, data (IP packets) are filled into an $(N \times 191)$ matrix where each cell of the matrix hosts one byte of information and N denotes the number of rows in the matrix. The standard defines the value of N to be one of

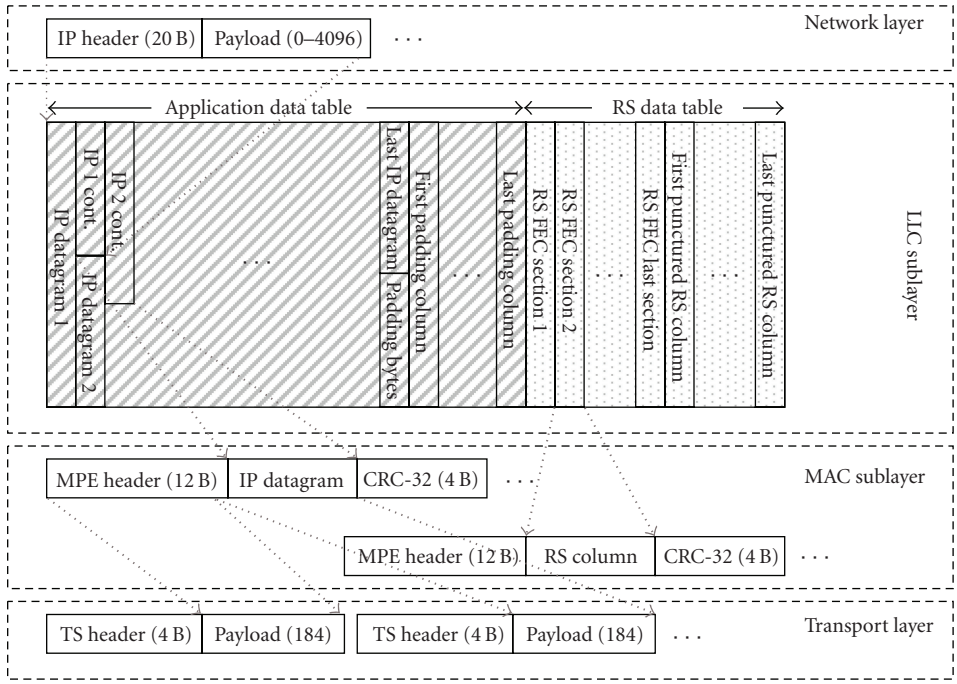


Figure 1: A subset of the protocol structure of DVB-H.

256, 512, 768, or 1024. The datagrams are filled into the matrix columnwise. RS codes are computed for each row and concatenated such that the final size of the matrix is of size $(N \times 255)$. The $(N \times 191)$ part of the matrix is called the application data table (ADT) and the adjacent $(N \times 64)$ part of the matrix is called the RS data table (RSDT). For ratecontrol and disallowing of IP packet fragmentation between two MPE-FEC frames in the standard, the ADT need not be completely filled. This unfilled part of the ADT is called padding. To control channel coderate, all 64 columns of RSDT need not be transmitted, that is, the RSDT may be punctured. The structure of an MPE-FEC matrix is shown in Figure 2 and further information on the MPE-FEC matrix construction can be obtained from [4].

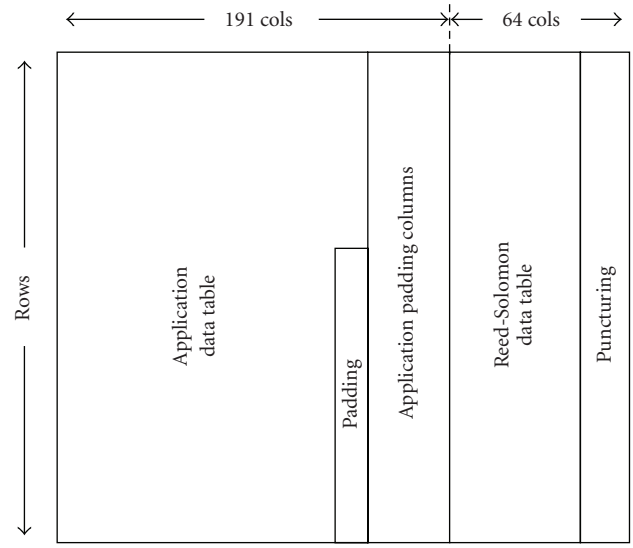


Figure 2: The MPE-FEC matrix structure.

2.3. Time slicing

Battery-operated mobile devices have a limited source of power. The power consumed in receiving, decoding, and demodulating a standard full-bandwidth DVB-T signal would use up substantial amount of battery life in a short time. Time slicing of the MPE-FEC frames is used to solve this problem [10]. The data is received in bursts so that the receiver, utilizing control signals, remains inactive when no bursts are to be received. The bursts are sent at a significantly higher bit rate compared to bit rate when conventional bit rate management is used.

Time slicing in DVB-H uses the Delta-T method to signal the relative start of the next burst, that is, the difference between the current time and the start of the next burst. The use of Delta-T method provides flexibility since parameters such as burst size, burst duration, burst bandwidth, and the offtimes can be freely varied. Figure 3 shows two time-sliced bursts and parameters that define time-sliced bursts.

3. H.264/AVC VIDEO CODING AND RTP ENCAPSULATION

H.264/AVC enables storage of multiple reference pictures for inter prediction and selection of the used reference picture on

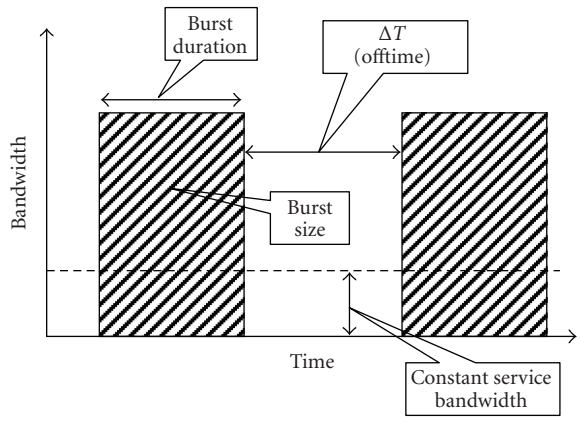


Figure 3: Time slicing in DVB-H.

macroblock or macroblock partition basis. In order to maximize compression efficiency, a motion vector is accompanied by a variable-length-coded index to a reference picture list. The reference picture list is initialized according to picture decoding order for inter slices and according to picture output order for bi-predictive slices. Slice headers may contain commands for reference picture list reordering.

Coded pictures of H.264/AVC can be categorized into three types: instantaneous decoding refresh (IDR) pictures, other reference pictures, and nonreference pictures. An IDR picture contains only intra-coded slices and causes marking of all previous reference pictures to be no longer used as references for subsequent pictures. An IDR picture can therefore be used as a random access point for the start of decoding or joining a session. It also provides a resynchronization point for decoding after transmission errors have occurred. A reference picture is stored and maintained as a prediction reference for inter prediction until it is no longer used for reference according to the reference picture marking process of H.264/AVC. A non-reference picture is not used for reference in inter prediction and can therefore be removed from a bit stream without any effect on other pictures.

The elementary unit for the output of an H.264/AVC encoder and the input of an H.264/AVC decoder is a network abstraction layer (NAL) unit. For transport over packet-oriented networks or storage into structured files, NAL units are typically encapsulated into packets or similar structures. NAL units can be categorized into video coding layer (VCL) NAL units, such as coded slices, and non-VCL NAL units, such as sequence and picture parameter sets.

The RTP payload format specification for H.264/AVC [11] includes the syntax and semantics of the RTP payload format, RTP packetization rules for H.264/AVC, informative RTP depacketization guidelines, and multipurpose Internet mail extensions (MIME) definition for use with session description protocol (SDP), including SDP offer-answer model consideration for codec capability exchange. The payload format specification contains three packetization modes: single NAL unit mode, noninterleaved mode, and interleaved mode.

In the single NAL unit packetization mode, one NAL unit is transmitted without any additional payload header in one RTP packet. In the non-interleaved mode, NAL units are transmitted in decoding order and multiple NAL units of one access unit can be encapsulated into the same RTP packet. Encapsulating multiple NAL units into the same RTP packet is especially beneficial when the size of the NAL units is relatively small, which is typically the case for parameter set NAL units, for example. The non-interleaved mode therefore helps to reduce the bit rate overhead caused by protocol headers compared to the transmitting relatively small NAL units with the single NAL unit mode.

The interleaved mode allows transmission of NAL units out of NAL unit decoding order and encapsulating of NAL units from different access units into the same RTP packet. In the interleaved mode, a decoding order number (DON) indicating the decoding order of NAL units is conveyed or derived for each NAL unit. In very low bitrates the interleaved packetization mode allows for encapsulating NAL units from more than one access unit into the same packet, which helps to reduce protocol header overhead. The interleaved mode can also be used for robust packet scheduling for unicast streaming [12, 13]. When interleaved transmission order is used, the decoding order of NAL units must be recovered in the receiver to obtain correct operation of the decoder. The receiver includes a receiver buffer to reorder packets from transmission order to the NAL unit decoding order.

4. UEP METHODS AND THEIR APPLICABILITY TO DVB-H

Priority encoding transmission (PET) [14] established the work towards UEP in packet-oriented systems. The data to be transmitted is partitioned to messages, which are protected one at a time. The messages are then classified to priority segments according to known characteristics of the source signal. For example, a group of pictures (GOP) can be considered as a message, and priority segments can be assigned according to the picture type (I, P, B) [15]. FEC repair data is then generated for each priority segment, and the resulting coded stream is divided into a certain amount of packets, each containing a fixed-length block of data from the resulting coded stream. The amount of FEC repair data is a function of the priority class. The PET scheme results into packets which contain data from each priority segment, and the number of packets required to reconstruct a priority segment can be tuned with the amount of FEC repair data for each priority segment. Horn et al. developed a similar scheme [16] compared to PET and provided details on the practical implementation and application with a spatially scalable video codec.

IETF RFC 2733 [17] specifies an RTP payload format for XOR-based FEC protection. The payload header of FEC packets contains a bit mask identifying the packet payloads over which the bitwise XOR operation is calculated and a few fields for RTP header recovery of the protected packets. One XOR FEC packet enables recovery of one lost source packet. Work is going on to replace IETF RFC 2733 with similar RTP

payload format for XOR-based FEC protection also including the capability of uneven levels of protection (ULP) [18]. The payloads of the protected source packets are split into consecutive byte ranges starting from beginning of the payload. The first byte range starting from the beginning of the packet corresponds to the strongest level of protection and the protection level decreases as a function of byte range order. Hence, the media data in the protected packets should be organized such a way that the data appears in descending order of importance with a payload and a similar number of bytes correspond to similar subjective impact in quality among the protected packets. The number of protected levels in FEC repair packets is selectable and an uneven level of protection is obtained when number of levels protecting a set of source packets is varied. For example, if there are three levels of protection, one FEC packet may protect all three levels, a second one may protect the two first levels, and a third one only the first level.

Both PET and the method proposed by Horn et al. produce packets in an interleaved manner such that they contain data of all priority classes as well as repair data. The packet transmission format therefore requires deinterleaving of payload data even when FEC decoding is not necessary. Furthermore, the packet formats are not compatible with any of the existing standards.

RFC 2733 and ULP operate in application layer and are therefore unable to utilize MPE-FEC efficiently. Both RFC 2733 and ULP are based on XOR, which is known to be clearly inferior to Reed-Solomon FEC when the size of the FEC matrix is relatively large. RFC 2733 and ULP also limit the FEC matrix to a size that may be too small for being efficiently used when applied to DVB-H.

We proposed a UEP scheme first for the 3GPP's multimedia broadcast/multicast service (MBMS) [19] but later specifically tailored for DVB-H [20]. The scheme classifies multimedia data to priority segments and computes an uneven amount of FEC repair data over priority segments similarly to what is done in PET and many subsequent UEP methods. However, in contrast to earlier methods, the packet format remains identical to the case in which EEP is applied. This maintains compatibility with terminals that are not capable of UEP data reception. Furthermore, MPE-FEC is reused instead of introducing any new FEC and packetization scheme at the application layer. Therefore, this method of UEP incurs a small amount of implementation changes compared to the existing DVB-H implementations. In other words this UEP scheme can be considered as a DVB-H-friendly version of PET and the method proposed by Horn et al.

The method proposed in [20] is briefly described next. First, the priority segmentation is performed across all media streams of the same service. In this paper, the audio stream is ranked as high priority, and for video we utilize temporal layering only. It is proposed that H.264/AVC bit streams are encoded in a temporally scalable manner and priority is assigned to temporal level of the pictures. For example, if non-hierarchical temporal scalability is used, that is, one or more non-reference pictures are present between each pair of reference pictures, the reference pictures can be assigned a higher priority compared to the non-reference pictures.

The multiplexed media datagrams corresponding to certain duration are encapsulated into two or more MPE-FEC matrices according to their priority label. These MPE-FEC matrices are referred to as peer MPE-FEC matrices. The number of peer MPE-FEC matrices in a time-sliced burst is equal to the number of unique priority labels assigned to the datagrams.

To construct the peer MPE-FEC matrices in a time-sliced burst, the datagrams are grouped using their priority labels. The grouping procedure is performed on all the datagrams that go into the time-sliced burst. The grouped datagrams are arranged in ascending order such that the datagrams with the lowest priority come first in the transmission order and the datagrams with the next higher priority comes next and continuing so forth until the datagram group that has the highest priority comes last in the transmission order. Figure 4 illustrates the priority grouping of a service consisting of a temporally scalable video stream and an audio stream. The audio stream and the reference pictures of the video stream are assigned the highest priority, whereas the non-reference pictures are grouped to low-priority MPE-FEC matrices.

The number of RSDT columns for all the MPE-FEC matrices in all the time-sliced bursts in the service should be such that the average service bit rate when using this method will not overshoot the maximum allowed service bit rate. All peer MPE-FEC matrices should be recoverable in normal channel conditions, and in bad channel conditions at least the high priority peer MPE-FEC matrix should be recoverable. Padding and puncturing are used to obtain the desired MPE-FEC code rates.

The estimation of code rates for varying channel error conditions is difficult in DVB-H. Firstly, due to the broadcast nature of the channel some users might be experiencing extremely harsh conditions, while at the same time other users might be having an excellent reception. If a transmitter, sending a service at a single code rate, caters to really harsh channel conditions by using a very low code rate, then there is an inefficient use of bandwidth for users having good reception. On the other hand if the transmitter sends a service at a high code-rate, making efficient use of the bandwidth, the capability of the receivers to receive and decode the service data under bad reception conditions is substantially reduced. Catering to both these groups optimally requires knowledge of the number of users having bad reception versus number of users having good reception. This again is a difficult task because DVB-H by its own does not provide any return channel. However, best practices for adjusting the code rate for sufficient reception quality on average can be derived from network measurement statistics or simulated channel models. For example, in [21] the rate distortion at different error, rates for H.264/AVC was evaluated, and the code rate of 3/4 was shown to be most efficient among the tested cases. This code rate was used in the simulations performed in this paper.

In order to obtain identical receiver power consumption compared to conventional data casting over DVB-H, the peer

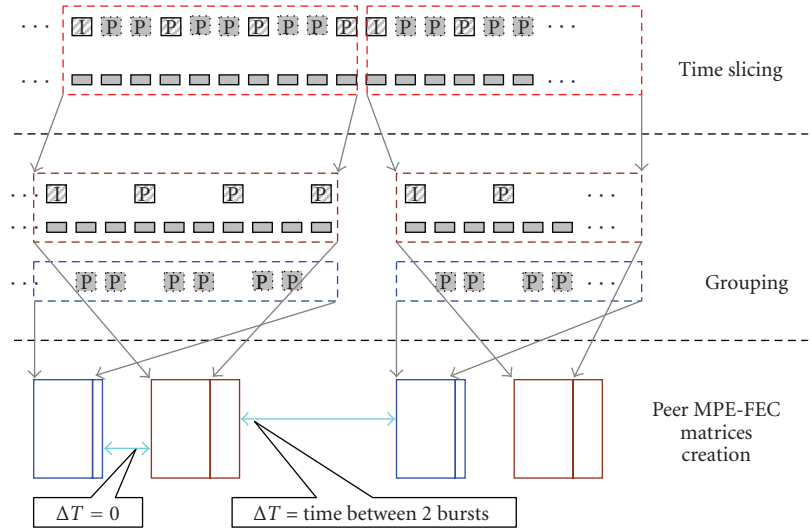


Figure 4: Priority assignment and peer matrix creation using video subsequences.

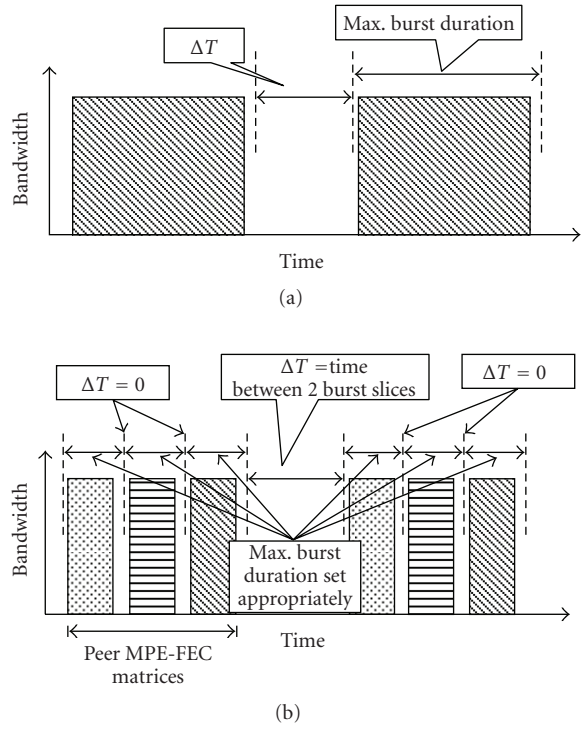


Figure 5: MPE-FEC matrix construction and transmission: (a) without UEP and (b) with UEP.

MPE-FEC matrices are transmitted back to back, that is, there is no transmission delay or interval between the peer MPE-FEC matrices. The Delta-T value in the MPE section headers for all sections in the peer MPE-FEC matrices other than the peer MPE-FEC matrix that contains the highest priority datagrams is assigned accordingly. The Delta-T value in the MPE section headers of MPE-FEC matrix that consists of the datagrams with the highest priority is set to indicate the time when the next time-sliced burst for the service starts. Figure 5 illustrates the method for construction of MPE-FEC matrix in the non-UEP case and the UEP case.

All packets for a particular peer MPE-FEC matrix are transmitted consecutively before any packet of another MPE-FEC matrix. Hence, FEC decoding for a priority segment can happen immediately after it has been completely received. The interleaved packetization mode of the RTP payload format for H.264/AVC is used to arrange the H.264/AVC RTP packets to the order required for the composition and transmission of the peer MPE-FEC matrices. The decoding order of packets is recovered when all peer MPE-FEC matrices of a time-sliced burst are received. As packet interleaving does not exceed time slice boundaries, the de-interleaving process does not add latency compared to conventional IP data casting beyond the processing delay for de-interleaving.

When a recipient tunes in and receives at least one but not all the peer MPE-FEC matrices for a particular time slice, it can decode and render the time slice with reduced quality compared to the reception of all peer MPE-FEC matrices. When the proposed UEP method is applied to an H.264/AVC stream with two temporal layers, the picture rate after tuning in may be reduced for the playback duration of the first received time slice. If the MPE-FEC source matrices of time slices were transmitted in descending order of importance, a newly joined recipient would have to wait until the first highest peer MPE-FEC matrix becomes available.

5. DVB-H SIMULATION AND TEST SETUP

As far as the authors are aware, there are no objective metrics that would satisfactorily reflect the subjective audio-visual quality experience, when perceived audio and video are degraded by both source coding and channel errors. For example, the peak signal-to-noise ratio (PSNR), frequently used

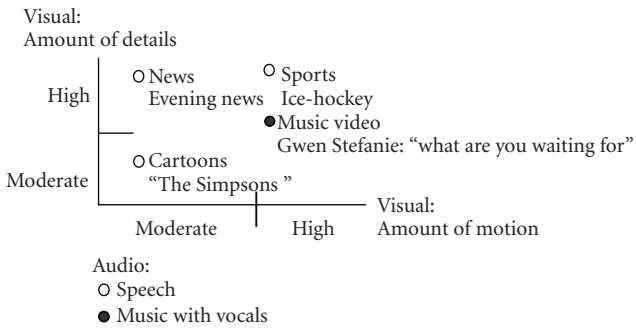


Figure 6: Genre of stimuli sequences, contents, and their audio-visual characteristics.

in measuring visual quality in video compression studies, provides consistent results only as long as the video signals being compared are affected by the same type of impairment [22]. Thus, subjective tests were carried out in a controlled laboratory environment to compare EEP provided by MPE-FEC and the UEP method presented in the previous section. Recommendations by International Telecommunication Union (ITU) [23, 24] were modified because no subjective test methodology in literature tuned specifically for this kind of work was found. The audio-visual bit streams presented to the subjective test participants where prepared by simulating a DVB-H channel.

5.1. Participants

45 participants, equally stratified by age group (18–45 years) and gender participated in the quality evaluation experiment. The number of experienced assessors, people engaged in multimedia processing or having extremely positive attitude towards technology [25] was restricted to 20%. All participants were verified to have normal or corrected-to-normal vision and hearing.

5.2. Test material selection and encoding

Four stimuli sequences representing different genre and contents with different audio-visual characteristics were chosen from a set of television broadcast material as described in Figure 6. The duration varied from 61 seconds to 64 seconds, because it was desirable to have semantically complete, meaningful, and understandable sequences for the participants.

The selected test materials were encoded using recommended codecs for the IP data casting service over DVB-H. Advanced audio coding (AAC) was used for audio and H.264/AVC for video encoding. The bit rate, sampling rate, and frame rate were selected according to the results of a previous study [26]. Mono-aural audio, which in [27] is shown to be more preferred than stereo at low bit rates, was coded at a bit rate of 32 kbps with a sampling rate of 16 kHz. Video bit streams were coded at a picture size of 176×144 pixels, a bit rate of 128 kbps, and a frame rate of 12.5 frames per second. Two sets of video sequences were encoded. The first set of sequences was targeted for the conventional method for audio-video broadcast over DVB-H and therefore contained only reference pictures. The second set of sequences was targeted for the proposed UEP scheme and therefore two non-reference pictures were coded between each pair of reference pictures. In both sets of sequences, at least one IDR frame was coded per DVB-H time slice to reduce the tuning-in delay at the receiver and provide better error resiliency against residual transmission errors. The first set of sequences was conventionally protected with MPE-FEC code rate of 3/4. For the second set of sequences, two MPE-FEC peer matrices were generated as described in Section 4, and the high-priority MPE-FEC peer matrix had a code rate of 3/4 while the low priority MPE-FEC peer matrix was unprotected by MPE-FEC. The time-sliced transmission burst interval for all sequences was set to approximately 1.5 seconds. This choice of code rates for the peer MPE-FEC matrices was chosen based on experimentation. It was found that under such harsh channel conditions as simulated in this paper, the best subjective quality was obtained when all the protection was dedicated to the most important priority while leaving the low-priority data unprotected.

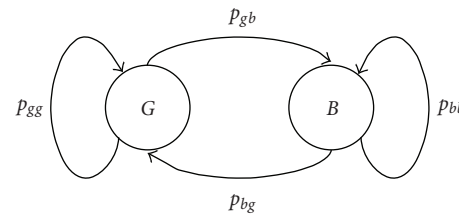


Figure 7: Gilbert-Elliot error model.

5.3. Channel simulation

Various stochastic models have been proposed for simulation of errors in a wireless channel. Among these, the Gilbert-Elliot (GE) model [28], shown in Figure 7, is popular and widely used because of its simplicity while it still produces a good representation of errors in a wireless channel. The GE model has been confirmed useful for simulating the packet error behavior also in DVB-H [29].

The model consists of two states representing two different channel conditions: the good state G and the bad state B. Each of these states is associated with bit error probabilities: e_g in the good state and e_b in the bad state where $e_g \ll e_b$. The average lengths of the error bursts are determined by the state transition probabilities p_{gb}, p_{bg} between the two states and the bit error probabilities e_g and e_b. In a simplified GE model e_g and e_b are set to zero and one, respectively. The state transmission matrix T is then given by the matrix

$$T = \begin{bmatrix} p_{gg} & p_{gb} \\ p_{bg} & p_{bb} \end{bmatrix}. \quad (1)$$

To simulate loss in the DVB-H channel, the results of a field trial carried out in an urban environment with an operable DVB-H system were used as basis. The receiver in the

field trials was located in a car, and the modulation used was 16 QAM. The field test results were used to train a simplified GE model for erroneous time-slices and estimate the state transition matrix.

The field test results were in the form of an MPE-FEC error pattern indicating which MPE-FEC frames contained uncorrectable transmission errors. This error pattern was first used as a training sequence for a simplified GE model resulting into the following state transition matrix:

$$T_{\text{mpe-fec}} = \begin{bmatrix} 0.8478 & 0.1522 \\ 0.4227 & 0.5773 \end{bmatrix}. \quad (2)$$

The state transition matrix was then used to generate an initial MPE-FEC error pattern. Finally, the length of randomly selected error bursts in the initial MPE-FEC error pattern was reduced gradually until error patterns of rates 6.9% and 13.8% were obtained.

MPE-FEC frame error rates (MFER) 6.9% and 13.8% after FEC decoding were chosen into the simulations based on an earlier test [30], in which the boundary of overall acceptability lied between these two rates, that is, the majority of participants considered the audio-visual quality resulting from 6.9% and 13.8% erroneous time-slice rate acceptable and nonacceptable, respectively. It is emphasized that the tested error rates are significantly higher than expected typical error rates for DVB-H services. The aim of the tests was to study the operation of audio-video broadcasting over DVB-H under extreme channel conditions. It is noted that MFER 5% has been conventionally used as an operative quality of restitution (QoR) limit for mobile reception [31].

To generate the error patterns for the transport stream (TS) packets within the uncorrectable MPE-FEC frames, a second simplified GE model was implemented. Based on manual assessment of some TS error patterns, we assumed that the average total number of TS packet errors was 235 and the average error burst length was 95 continuous TS packets. In a simplified GE model the average error rate E is given by $E = (1 - p_{gg})/(2 - p_{gg} - p_{bb})$ and the average burst error lengths B is given by $B = 1/(1 - p_{bb})$. Solving for p_{gg} and p_{bb} a state transition matrix

$$T_{\text{ts}} = \begin{bmatrix} 0.99 & 0.01 \\ 0.01 & 0.99 \end{bmatrix} \quad (3)$$

was obtained, which was used to generate the TS error patterns within an erroneous MPE-FEC frame. The result was a TS error pattern that approximated the results of the actual field test.

The generated TS packet errors were used to corrupt the coded audio-visual sequences. Error correction operation using MPE-FEC was simulated and the resulting residual IP packet error pattern was obtained. The residual IP error pattern reflected the uncorrectable errors in the channel.

5.4. Decoder error concealment

The video decoder used a simple error concealment procedure. When the decoder encountered residual errors in or losses of reference pictures, it stopped decoding of any subsequent pictures until an IDR picture arrived. During the period when the decoder stopped decoding, it presented the last uncorrupted decoded picture. Subjectively, when this method is used, a transmission error is perceived as discontinuous motion in visual streams. The duration of these discontinuities in visual streams depends on the IDR interval and the placement of the error between two IDR pictures. When the decoder encountered losses of non-reference pictures, the previous correct picture in output order was rendered and decoding continued from the next picture in decoding order. Consequently, if residual errors were present in the peer MPE-FEC matrix for the non-reference pictures but not present in the corresponding peer MPE-FEC matrix for audio and reference pictures, users perceived temporary fluctuations of picture rate, that is, jerky but generally continuous motion.

AAC audio frames are essentially independent of each other and a loss of any one frame of the bit stream does not substantially affect any other frames of an audio channel. When an audio frame was lost, it was replaced with a null frame perceived as discontinuous audio.

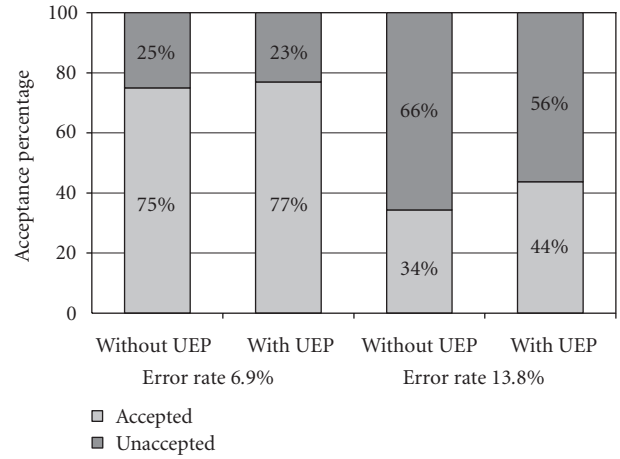


Figure 8: Overall acceptability rating of UEP scheme.

5.5. Subjective test procedure

Before the start of the test session, the participants were briefed about the test and their sensorial acuity was measured and they filled the demographic questionnaire. The sensorial tests included in the measurements of visual acuity (20/40), color vision [32, 33], and the aural acuity [34–36].

The subjective test started with a combination of anchoring and training. Participants were shown the extremes of quality range of stimuli to familiarize the participants with the test task, the contents, and the variation in quality they could expect in the actual tests that followed. The tests used retrospective overall evaluation based on the absolute category rating (ACR), also known as single stimulus method, which is typically used in system or performance evaluation [24]. The test sequences were presented one at a time and

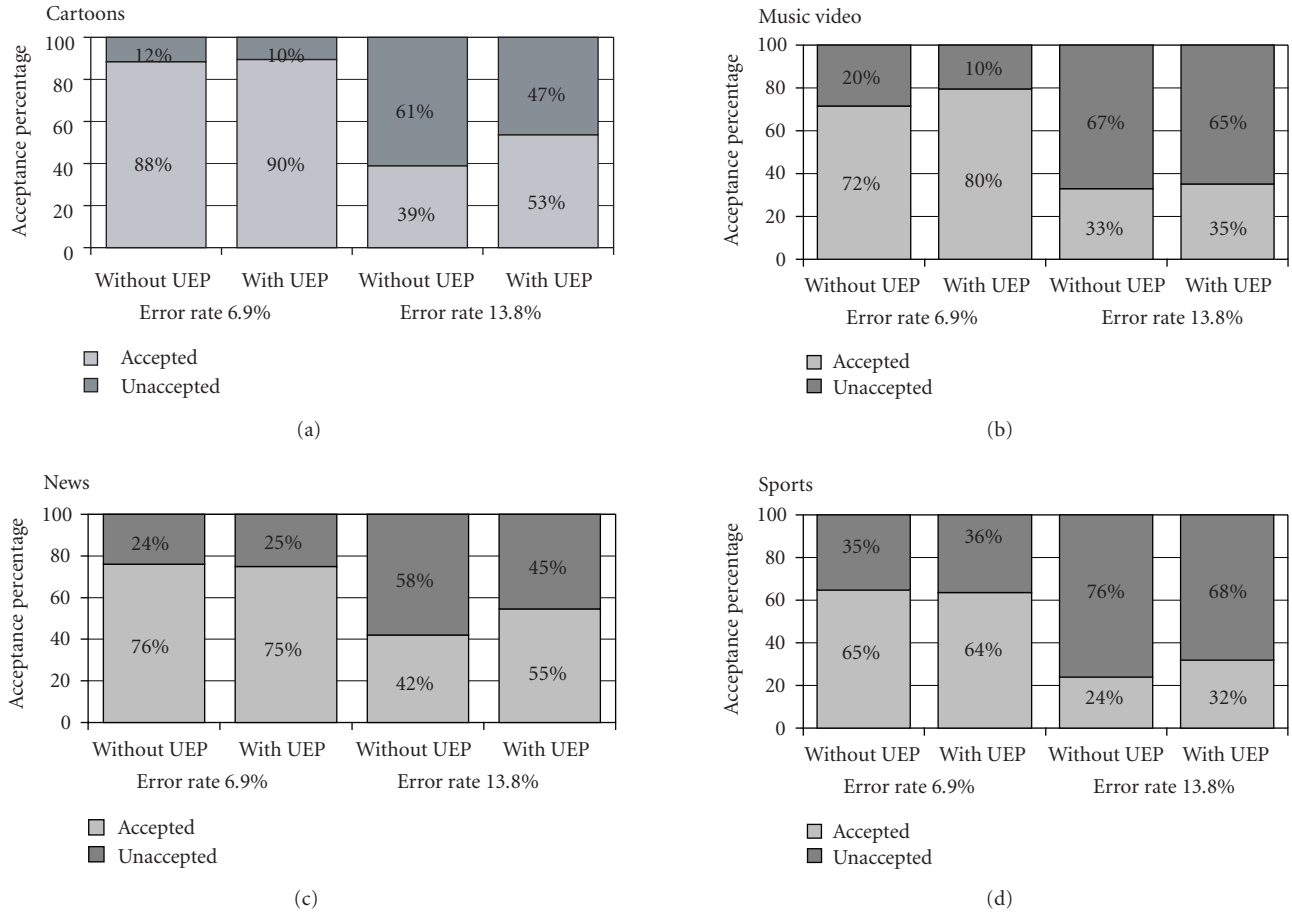


Figure 9: Per-sequence acceptability ratings.

they are rated independently after each presentation [24]. The quality ratings were given during a 5-second-long answering time by using a discrete, unlabelled 11-point scale and the acceptance of quality (yes/no choice). The whole test session for a participant consisted of two rounds with two sets of audio-visual clips [A, B] and the starting round was randomized. After the actual test, qualitative data of experiences on the erroneous streams were gathered. One test session lasted about 1.5 hours.

The clips were presented with Nokia 6630 mobile phone, which was enclosed in a stand that left only the screen and buttons of the device visible. The device and the front of the stand were vertically aligned and the viewing distance was set to 44 cm. The headphones delivered in Nokia 6630 sales package were used for audio playback. Audio playback loudness level was adjusted to 75 dB(A) (+ 10 dB(A) for peaks).

5.6. Data analysis methods

For data analysis, two different nonparametric methods were used. Overall quality ratings were analyzed with Wilcoxon matched pair signed rank test which was used to measure the differences between two related and ordinal data sets because

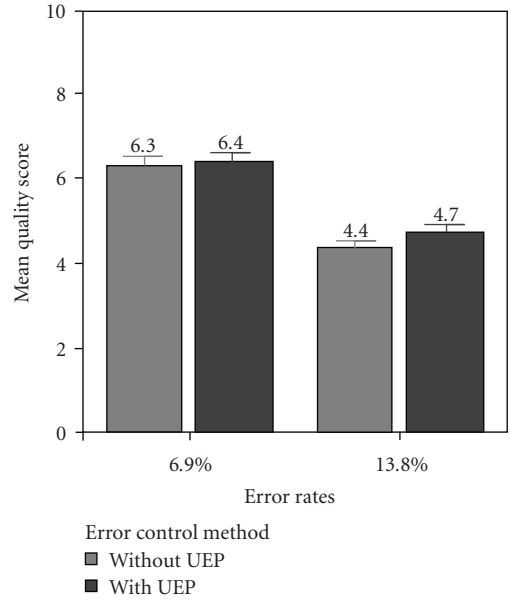


Figure 10: Overall mean satisfaction ratings for UEP scheme. The error bars show 95% CI of mean.

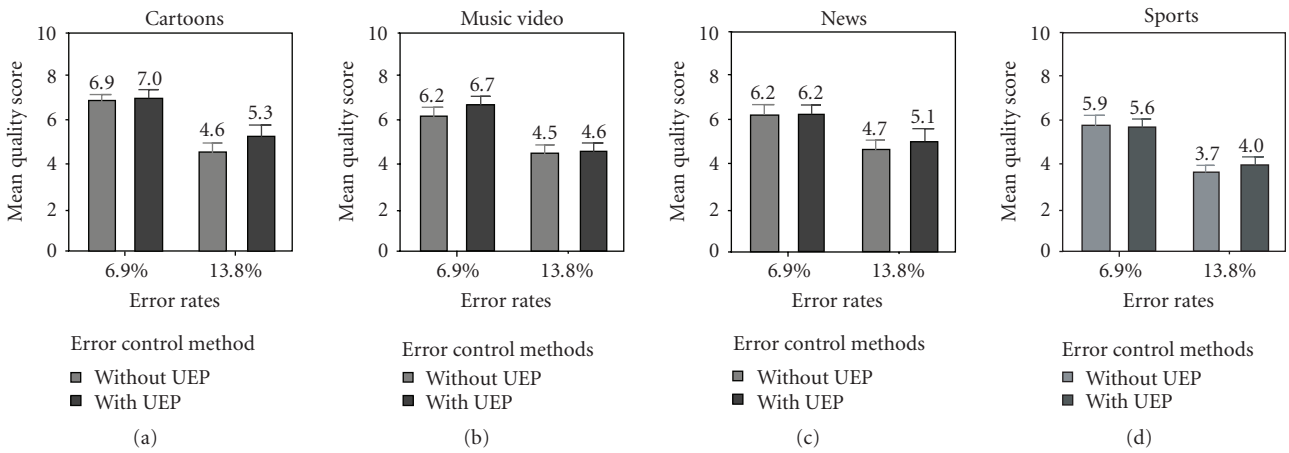


Figure 11: Per-sequence satisfaction ratings for UEP scheme. The error bars show 95% CI of mean.

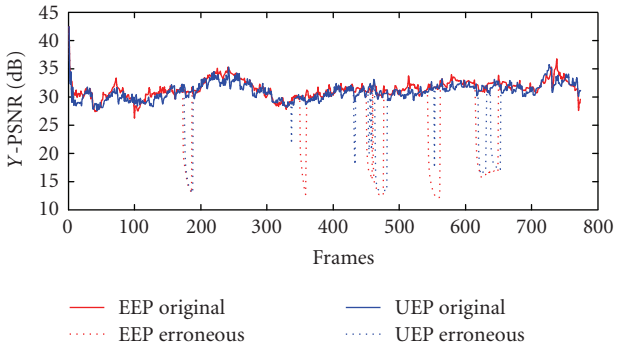


Figure 12: Per-frame PSNR for sports sequence at 13.8% MFER.

the preassumption of parametric methods (normality) was not filled [37]. For the nominal acceptance evaluations McNemar's test was applied to test the differences between two categories in the related data [37]. The significance level of $P < .05$ was adopted in this study.

6. RESULTS

Figure 8 shows the cumulative acceptability statistics and Figure 10 shows mean satisfaction scores for all audio-visual sequences at the two simulated error rates. When the residual time slice error rate was 6.9%, the proposed UEP method did not have a significant impact on overall acceptance or satisfaction rating compared to the conventional method (McNemar $P > .05$, Wilcoxon $Z = -0.71$, $P > .05$). A majority of participants rated sequences of both error control methods as acceptable. When the residual time slice error rate was 13.8%, the proposed UEP method outperformed the conventional method significantly (McNemar $P < .001$, Wilcoxon $Z = -4.1$, $P < .001$), which can also be seen in the number of acceptable clips in Figure 8. However, on average, the sequences of both the proposed UEP method and the conventional method remained unacceptable.

Figures 9 and 11 show the acceptability and mean satisfaction statistics for each of the four audio-visual sequences at 6.9% and 13.8% residual MPE-FEC time slice error rates, respectively. At the error rate of 6.9%, the improvement provided by the proposed UEP method was not significant in any sequences (McNemar, Wilcoxon $P > .05$). However, at the error rate of 13.8% the proposed UEP scheme outperformed the conventional scheme significantly in animation (McNemar $P < .01$, Wilcoxon $Z = -3.7$, $P < .001$), news (Wilcoxon $Z = -2.0$, $P < .05$), and sports (McNemar $P < .05$). Moreover, a majority of participants rated the animation and news sequences of the proposed UEP scheme as acceptable under residual time slice error rate of 13.8%, whereas the corresponding conventionally coded and transmitted sequences were rated as unacceptable by a majority of participants. In other words, the threshold for a transmission error rate yielding an unacceptable audio-visual quality was increased due to the proposed UEP scheme.

Figure 12 shows the per-frame PSNR behavior for the sports sequence at 13.8% MFER for both EEP and UEP. It clearly illustrates how some burst errors in the EEP case can be transformed into isolated single picture errors in the UEP case.

7. CONCLUSIONS

The paper reviewed some methods for unequal error protection (UEP) and analyzed their applicability to DVB-H. A method based on priority segmentation of the media streams of a service was chosen for more detailed analysis. The presented UEP method was compared to equal error protection (EEP) provided by the link layer forward error correction scheme (MPE-FEC) of DVB-H. Several audio-visual streams were processed through a DVB-H channel model for the comparison, and the resulting streams were presented in a comprehensive subjective quality evaluation conducted in a controlled laboratory environment. Two MPE-FEC error rates (MFER) were selected for the evaluation, 6.9% and

13.8%, which resulted into acceptable and unacceptable average quality, respectively, according to a previous study. The results of the evaluation revealed that, at MFER of 6.9%, the presented UEP scheme was at least as good as the EEP case obtained by conventional use of MPE-FEC. However, at MFER of 13.8%, the use of the proposed UEP method improved the subjective acceptability of the tested multimedia sequences on average, as the share of participants rating the sequences acceptable was 10 percent units higher in the UEP case compared to the EEP case.

ACKNOWLEDGMENTS

This study was funded by Radio- ja Televisiotekniikan Tutkimus Oy (RTT). RTT is a nonprofit organization that contributes to the research and development of new radio and television technologies in Finland. Satu Jumisko-Pyykkö's work is supported by the Graduate School in User-Centered Information Technology.

REFERENCES

[1] European Telecommunications Standards Institute (ETSI), "Digital video broadcasting (DVB); transmission system for handheld terminals (DVB-H)," European Standard EN 302 304, version 1.1.1, November 2004.

[2] European Telecommunications Standards Institute (ETSI), "Digital Video Broadcasting (DVB): framing structure, channel coding and modulation for digital terrestrial television," ETSI standard, EN 300 744, 2001.

[3] ISO/IEC 13818-1, "Information technology—generic coding of moving pictures and associated audio information: systems," November 1994.

[4] European Telecommunications Standards Institute (ETSI), "Digital video broadcasting (DVB); DVB specification for data broadcasting," European Standard EN 301 192, version 1.4.1, November 2004.

[5] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *SIAM Journal of Applied Mathematics*, vol. 8, no. 2, pp. 300–304, 1960.

[6] European Telecommunications Standards Institute (ETSI), "Digital Video Broadcasting (DVB); specification for the use of video and audio coding in DVB services delivered directly over IP protocols," European standard TS 102 005 V1.2.1, November 2005.

[7] International Organization for Standardization (ISO)/ International Engineering Consortium (IEC), "Information technology—generic coding of moving picture and associated audio information—part 3: audio," International Standard 14496-3 (2001), including "Bandwidth Extension," International Standard 14496-3 AMD-1 (2001) and "Parametric Coding for High Quality Audio," International Standard 14496-3 AMD-2 (2004).

[8] Telecommunication Standardization Sector of International Telecommunication Union (ITU-T), "Advanced video coding for generic audiovisual services," Recommendation H.264, March 2005.

[9] D. Tian, M. M. Hannuksela, and M. Gabbouj, "Sub-sequence video coding for improved temporal scalability," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS '05)*, vol. 6, pp. 6074–6077, Kobe, Japan, May 2005.

[10] J. Aaltonen, H. Pekonen, T. Auranen, K. Laiho, and P. Talmola, "Power saving considerations in mobile datacasting terminals," in *Proceedings of IEEE International Symposium on Consumer Electronics*, pp. F43–F48, Erfurt, Germany, September 2002.

[11] S. Wenger, M. M. Hannuksela, T. Stockhammer, M. Westerlund, and D. Singer, "RTP Payload Format for H.264 Video," IETF RFC 3984, February 2005.

[12] T. Schierl, M. Kampmann, and T. Wiegand, "H.264/AVC interleaving for 3G wireless video streaming," in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME '05)*, pp. 868–871, Amsterdam, The Netherlands, July 2005.

[13] T. Schierl, T. Wiegand, and M. Kampmann, "3GPP compliant adaptive wireless video streaming using H.264/AVC," in *Proceedings of IEEE International Conference on Image Processing (ICIP '05)*, vol. 3, pp. 696–699, Genova, Italy, September 2005.

[14] A. Albanese, J. Blömer, J. Edmonds, M. Luby, and M. Sudan, "Priority encoding transmission," *IEEE Transactions on Information Theory*, vol. 42, no. 6, part 1, pp. 1737–1744, 1996.

[15] C. Leicher, "Hierarchical encoding of MPEG sequences using priority encoding transmission (PET)," Tech. Rep. TR-94-058, International Computer Science Institute, Berkeley, Calif, USA, November 1994.

[16] U. Horn, K. Stuhlmüller, M. Link, and B. Girod, "Robust internet video transmission based on scalable coding and unequal error protection," *Signal Processing: Image Communication*, vol. 15, no. 1-2, pp. 77–94, 1999.

[17] J. Rosenberg and H. Schulzrinne, "An RTP payload format for generic forward error correction," Internet Engineering Task Force Request for Comments 2733, December 1999.

[18] A. H. Li, "RTP payload format for generic forward error correction," Internet Engineering Task Force Internet Draft draft-ietf-avt-ulp-18.txt, June 2006.

[19] D. Tian, V. K. Malamal Vadakital, M. M. Hannuksela, S. Wenger, and M. Gabbouj, "Improved H.264/AVC video broadcast /multicast," in *Visual Communications and Image Processing 2005*, vol. 5960 of *Proceedings of SPIE*, pp. 71–82, Beijing, China, July 2005.

[20] V. K. Malamal Vadakital, M. M. Hannuksela, M. Rezaei, and M. Gabbouj, "Method for unequal error protection in DVB-H for mobile television," in *Proceedings of the 17th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '06)*, pp. 1–5, Helsinki, Finland, September 2006.

[21] V. K. Malamal Vadakital, M. M. Hannuksela, H. Pekkonen, and M. Gabbouj, "On datacasting of H.264/AVC over DVB-H," in *Proceedings of the 7th IEEE Workshop on Multimedia Signal Processing*, pp. 1–4, Shanghai, China, October 2005.

[22] B. Girod, "Psychovisual aspects of image communication," *Signal Processing*, vol. 28, no. 3, pp. 239–251, 1992.

[23] International Telecommunications Union—Radiocommunication sector, "Methodology for the subjective assessment of the quality of television pictures," ITU-R BT.500-11, 2002.

[24] International Telecommunications Union—Telecommunication sector, "Subjective video quality assessment methods for multimedia applications," ITU-T P.910, 1999.

[25] E. M. Rogers, *Diffusion of Innovation*, Free Press, New York, NY, USA, 5th edition, 2003.

[26] S. Jumisko-Pyykkö and J. Häkkinen, "Evaluation of subjective video quality on mobile devices," in *Proceedings of the 13th ACM International Conference on Multimedia*, pp. 535–538, Singapore, November 2005.

[27] S. Winkler and C. Faller, "Maximizing audiovisual quality at low bitrates," in *Proceedings of the 1st International Workshop on Video Processing and Quality Metrics for Consumer Electronics (VPQM '05)*, Scottsdale, Ariz, USA, January 2005.
[28] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell Systems Technical Journal*, vol. 39, pp. 1253–1265, 1960.
[29] J. Poikonen and J. Paavola, "Comparison of finite-state models for simulating the DVB-H link layer performance," in *Proceedings of the 2nd International Symposium on Wireless Communications Systems Conference (ISWCS '05)*, pp. 153–157, Siena, Italy, September 2005.
[30] S. Jumisko-Pyykkö, V. K. Malamal Vadakital, M. Liinasuo, and M. M. Hannuksela, "Acceptance of audiovisual quality in erroneous television sequences over a DVB-H channel," in *Proceedings of the 2nd International Workshop on Video Processing and Quality Metrics for Consumer Electronics (VPQM '06)*, pp. 1–5, Scottsdale, Ariz, USA, January 2006.
[31] G. Faria, J. A. Henriksson, E. Stare, and P. Talmola, "DVB-H: digital broadcast services to handheld devices," *Proceedings of the IEEE*, vol. 94, no. 1, pp. 194–209, 2006.
[32] Ishihara's tests for color deficiency 24 Plates edition. Tokyo, Kanehara trading INC, 2005.
[33] International Standardization Organization, "Visual acuity testing," ISO 8596, 1994.
[34] International Standardization Organization, "Audiometric test methods," ISO 8523-1- 2, 1989.
[35] International Standardization Organization, "ISO Standards Handbook 35," Acoustics. 1st edition, p. 386. Switzerland, 1990.
[36] International Standardization Organization, "Statistical distribution of hearing threshold," ISO 7029 SFS-EN Acoustics, 2000.
[37] H. Coolican, *Research Methods and Statistics in Psychology*, J. W. Arrowsmith, London, UK, 4th edition, 2004.

Miska M. Hannuksela is a Research Leader in Nokia Research Center, Tampere, Finland. He has more than 10 years of experience in video compression and multimedia communication systems. He has been an active delegate in international standardization organizations, such as the Joint Video Team, the Digital Video Broadcasting Project, and the 3rd Generation Partnership Project. His research interests include scalable and error-resilient video coding, real-time multimedia broadcast systems, and human perception of audiovisual quality. He holds more than 15 international patents and has authored several tens of academic papers.

Vinod Kumar Malamal Vadakital received his B.Tech. degree in computer science and engineering from Bangalore University, Bangalore, India, and an M.S. degree in information technology from Tampere University of Technology, Tampere, Finland, in 1998 and 2005, respectively. From 1999 to 2001, he worked as a Project Assistant at the Indian Institute of Science, Bangalore, India. From 2001 to 2003 he was a Research Engineer at Fraunhofer Institute of Integrated Circuits (IIS-B), Erlangen, Germany. From 2003 to 2005, he worked as a Research Assistant at Tampere University of Technology. Currently he is a researcher at the Tampere University of Technology and he is working towards his doctoral degree. His research interests are in the areas of video coding algorithms, video quality analysis, and mobile multimedia communications.

Satu Jumisko-Pyykkö received the M.S. degree in software engineering in 2005 from Tampere University of Technology. She has broad studies in multimedia, human-computer interaction, computer-aided learning, and psychology from the University of Helsinki. She is currently a Ph.D. student in the Graduate School in User-Centered Information Technology and is working as a researcher in the Institute of Human-Centered Technology at Tampere University of Technology. Her research interests are focused on human-centered approach to multimedia quality and development of research methods for understanding and measuring experienced multimodal quality.