Error control for non-scalable and scalable video over error-prone networks

Fang, Tao

2007

Fang, T. (2007). Error control for non-scalable and scalable video over error-prone networks. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/4241>

<https://doi.org/10.32657/10356/4241>

Nanyang Technological University

Downloaded on 26 Apr 2023 09:33:11 SGT

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

**ERROR CONTROL FOR NON-SCALABLE AND
SCALABLE VIDEO OVER ERROR-PRONE
NETWORKS**

FANG TAO

FANG TAO

SCHOOL OF ELECTRICAL & ELECTRONIC ENGINEERING

2007

ERROR CONTROL FOR NON-SCALABLE AND
SCALABLE VIDEO OVER ERROR-PRONE NETWORKS

2007

Error Control for Non-scalable and Scalable Video over Error-prone Networks

Fang Tao

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

2007

*To all in my family,
for their love.*

Acknowledgements

First and foremost, I would like to express my deepest appreciation and sincere gratitude to my supervisor, Dr. Lap-Pui Chau, for his invaluable instruction and great support. It is he who gave me strong support and sincere encouragement throughout the tough times of the research. Within these three years, I not only learned specialized knowledge from him, but also learned the spirit of exploring science from him. This work would not have been possible without his excellent guidance and persistent encouragement.

I would like to thank my parents, my grandparents, and all the members in my family for teaching me to understand the value of hard work and supporting me during my education as well as in life in general. Their love and support are always the greatest inspiration to me from the first day I came to the world.

I would also like to thank all of my colleagues, the faculty and technicians in Media Technology Laboratory, NTU. All of my experiences and interactions with these people have shaped me and my work, and helped making my graduate experience worthwhile and unforgettable.

Summary

Recent advances in technology have resulted in a significant growth in wireless communications. As wireless access becomes more commonplace, the need for reliable transmission of mixed media information, such as speech, text, image and video data, over wireless channels is becoming an increasingly important application requirement. As a result, the study of video transmission over error-prone environment is highly promising.

Due to the characteristics of wireless transmission environments, namely low bandwidth and high bit error rate, efficient video transmission becomes challenging and error control must be applied to improve the robustness of the video coder.

In the thesis, we firstly give an overview of the strategies that can be adopted to make the video data robust. After that, the proposed error control techniques are introduced, which we believe to be the state-of-the-art technologies for video communications over error-prone networks. The contributions are summarized as follows in terms of the coding methods (non-scalable and scalable, respectively) applied.

- 1) In the first part, several efficient error-resilience methods for non-scalable video coding have been proposed, including:

a) two content-based resynchronization marker insertion methods, which improve perceptual image quality by intelligently selecting the positions of resynchronization markers.

b) an error resilient group-of-picture (GOP) structure, which improves robustness by selecting the optimal position of the I frame in a GOP.

c) an optimal macroblock (MB) interleaving, which groups MB's into slices considering the maximal minimal distance of these MB's to help error concealment work more efficiently.

2) In the second part, several efficient error control methods for scalable video coding have been proposed, including:

a) an optimal resynchronization for scalable video, which inserts resynchronization markers in each bitplane according to the importance of the bitplane and the time-varying channel condition. The performance is maximized with the proposed method.

b) a GOP-based unequal error protection (UEP) for scalable video, in which unequal amounts of protection are allocated to the different frames (I- or P-frame) of a GOP and in each frame, unequal amounts of protection are allocated to the progressive bit-stream of scalable video. With such a two-dimensional channel rate allocation, the video quality has been much improved.

Content

Acknowledgments	i
Summary	ii
List of Figures	ix
List of Tables	xiv
<u>Chapter 1</u> Introduction	1
1.1 Motivation.....	1
1.2 Objective and Contributions.....	3
1.3 Outline of the Thesis	4
<u>Chapter 2</u> Literature Review	7
2.1 Error Resilience	7
2.1.1 Mitigation of Spatial Error Propagation	8
2.1.2 Mitigation of Temporal Error Propagation	13

2.1.3 Multiple Description Coding (MDC).....	17
2.1.4 Scalable Coding	20
2.2 Error Protection.....	23
2.2.1 Forward Error Correction (FEC)	23
2.2.2 Unequal Error Protection (UEP).....	26
2.2.3 Joint Source and Channel Coding (JSCC)	28
2.3 Error Concealment	31
2.3.1 Recovery of Texture Information	31
2.3.2 Recovery of Motion Vectors.....	34
2.3.3 Recovery of Coding Modes	34
<u>Chapter 3</u> Content-based Resynchronization Approach for Subjective Quality Enhancement	35
3.1 Introduction.....	35
3.2 Content-based Resynchronization.....	37
3.2.1 Perceptual Impact of Error on Video Quality	37
3.2.2 Content-based Resynchronization Scheme.....	39
3.2.3 Simulation Results	42
3.3 Properties of the Proposed Scheme	45
3.4 Conclusion	46
<u>Chapter 4</u> Error Resilient GOP Structure with Efficient Reverse-play Functionality	47

4.1 Introduction.....	47
4.2 Proposed GOP Architecture	50
4.2.1 Proposed Error Resilient GOP Structure	50
4.2.2 Selection of Optimal Position of I Frame in a GOP	52
4.2.3 Analysis of Coding Efficiency, System Delay and Complexity	54
4.3 Efficient Reverse-play Functionality of the Proposed GOP Structure	58
4.4 Experimental Results.....	61
4.5 Conclusion	66
 <u>Chapter 5</u> Optimal MB Interleaving Scheme Using Maximum Minimal Distance	
5.1 Introduction.....	68
5.2 Optimal Interleaving Scheme	71
5.2.1 Optimal Interleaving Scheme	72
5.2.2 Properties of the Proposed Interleaving Scheme	75
5.3 Simulation Results	76
5.4 Conclusion	78
 <u>Chapter 6</u> Resynchronization Marker Insertion for Layered Video	
6.1 Introduction.....	82
6.2 Resynchronization Approach for FGS Video.....	84
6.2.1 FGS and its Characteristics	84

6.2.2 Factors Affecting Video Quality.....	85
6.2.3 Optimization Problem.....	87
6.2.4 Hill-climbing Algorithm	90
6.3 Experimental Results.....	91
6.4 Conclusion	95
 <u>Chapter 7</u> GOP-based Channel Rate Allocation Using Genetic Algorithm for Scalable Video Streaming.....	97
 7.1 Introduction.....	97
 7.2 Proposed Channel Rate Allocation Approach	100
7.2.1 Proposed Framework	100
7.2.2 Problem Formulation	104
7.2.3 Analytical Model for Channel Distortion	105
7.2.4 Model of Wireless Packet-Erasure Channel	107
7.2.5 Application of GA for Fast Channel Rate Allocation.....	109
 7.3 Experimental Results.....	114
7.3.1 Channel Rate Allocation Using GA.....	115
7.3.2 Performance Comparison.....	118
 7.4 Conclusion	121
 <u>Chapter 8</u> Conclusion and Recommendations	123
 8.1 Conclusion	123
 8.2 Recommendations for Future Research Work	127

Author's Publications	130
Journal Publications	130
Conference Publications	131
Bibliography	133

List of Figures

Fig. 1.1. Video transmission system.....	2
Fig. 1.2. Error propagation.....	3
Fig. 2.1. Resynchronization. (a) H.263 packet approach and (b) MPEG-4 periodic approach.....	9
Fig. 2.2. RVLC.....	11
Fig. 2.3. Data partitioning.....	12
Fig. 2.4. Error confinement.....	16
Fig. 2.5. A multiple description coding communication system.....	18
Fig. 2.6. Reed-Solomon code.....	24
Fig. 2.7. Rate-distortion relation for (a) source coding and (b) joint source and channel coding.....	29
Fig. 3.1. Impact of error location on perceptual video quality according to different foreground data location in one slice.....	37

Fig. 3.2. (a) Segmentation result and the spatial position of resynchronization markers of (b) H.263 (version 1) packet approach, (c) MPEG-4 periodic approach and (d) this work.....	39
Fig. 3.3. Impact of error on a QCIF image quality when the location of error is in the end of a GOB of H.263. (a) H.263 (version 1) packet approach, (b) MPEG-4 periodic approach and (c) the approach used in this work.....	41
Fig. 3.4. Impact of error on a QCIF image quality when the location of error is in the beginning of a GOB of H.263. (a) H.263 (version 1) packet approach, (b) MPEG-4 periodic approach and (c) the approach used in this work.....	41
Fig. 3.5. Decoded Frame 93 of Foreman QCIF (a) without channel error, (b) this work with $BER=10^{-5}$, (c) MPEG-4 approach with $BER=10^{-5}$ and (d) H.263 approach with $BER=10^{-5}$	42
Fig. 3.6. Foreman QCIF (200Kbps) average PSNR comparison of (a) overall picture and (b) foreground.....	43
Fig. 3.7. Foreman QCIF (100Kbps) average PSNR comparison of (a) overall picture and (b) foreground.....	43
Fig. 3.8. Carphone QCIF (200Kbps) average PSNR comparison of (a) overall picture and (b) foreground.....	44
Fig. 3.9. Carphone QCIF (100Kbps) average PSNR comparison of (a) overall picture and (b) foreground.....	45
Fig. 4.1. (a) Conventional MPEG GOP structure and (b) the proposed GOP structure...	48
Fig. 4.2. GOP structure without B frames of (a) MPEG and (b) this work.....	51

Fig. 4.3. Selection of optimal position of switching point in the proposed GOP structure.....	53
Fig. 4.4. Average length of error propagation vs. i . GOP length $n = 8, 9, 10, 11$, respectively.....	54
Fig. 4.5. (a)-(c): Number of bits used by conventional MPEG and this work of Mother & daughter QCIF, Foreman QCIF and Stefan QCIF, respectively. (d)-(f): Corresponding PSNR value of the three sequences.....	55
Fig. 4.6. Bandwidth requirement with respect to the reverse play operation for sending (a) Mother & Daughter, (b) Foreman and (c) Stefan.....	60
Fig. 4.7. Slice losses at frame 7, 19, 33, 46, 59 and 72.....	62
Fig. 4.8. Average PSNR comparison according to different delay requirements. (a) Mother & Daughter, (b) Foreman and (c) Stefan.....	64
Fig. 4.9. GOP structures applied in the experiments with (a) delay of 5 frames, (b) delay of 3 frames and (c) delay of 1 frame.....	65
Fig. 4.10. GOP structures with B frames of (a) the proposed structure and (b) a compliant structure that starts with an I-frame with compliant prediction directions.....	66
Fig. 4.11. Average PSNR comparison between the proposed structure and the compliant structure for (a) Foreman and (b) Stefan.....	67
Fig. 5.1. Partition of a QCIF frame into 9 slices when using (a) checker-board type of mapping and (b) this work. The number signifies the number of the slice that the MB belongs to.....	69
Fig. 5.2. The loss patterns of (a) checker-board and (b) this work. Packet loss rate=10%.....	73

Fig. 5.3. Error concealed images for Foreman QCIF when packet loss rate=10%. (a) No loss, PSNR= 37.5, (b) checker-board, PSNR=29.4, (c) this work, PSNR=29.8.....	77
Fig. 5.4. Error concealed images for Coastguard QCIF when packet loss rate=10%. (a) No loss, PSNR= 32.7, (b) checker-board, PSNR=28.6, (c) this work, PSNR=29.3.....	77
Fig. 5.5. The loss patterns of (a) checker-board and (b) this work. Packet loss rate=20%.....	80
Fig. 5.6. Error concealed images for Foreman QCIF when packet loss rate=20%. (a) No loss, PSNR= 37.5, (b) checker-board, PSNR=23.6, (c) this work, PSNR=25.3.....	81
Fig. 5.7. Error concealed images for Coastguard QCIF when packet loss rate=20%. (a) No loss, PSNR= 32.7, (b) checker-board, PSNR=23.9, (c) this work, PSNR=26.5.....	81
Fig. 6.1. FGS structure at the encoder and decoder.....	84
Fig. 6.2. The effect on the quality of the video due to loss of different bitplane.....	86
Fig. 6.3. Two-bitplane scalability results for Foreman.....	92
Fig. 6.4. Two-bitplane scalability results for Coastguard.....	93
Fig. 6.5. Three-bitplane scalability results for Foreman.....	94
Fig. 6.6. Three-bitplane scalability results for Coastguard.....	95
Fig. 7.1. System overview. Different gray colors denote the degree of importance: The darker, the more important.....	100
Fig. 7.2. A frame is partitioned into two slice groups. The shaded macroblocks belong to slice group 1 and the white macroblocks to slice group 2.....	101
Fig. 7.3. A GOP is partitioned into two sub-bitstreams. The shaded slices belong to sub-bitstream 1 and the white slices to sub-bitstream 2.....	102
Fig. 7.4. Structure of channel rate allocation for a sub-bitstream.....	103

Fig. 7.5. Flowchart of GA to solve the optimization problem.....	110
Fig. 7.6. Selection of the number of generation when applying GA.....	115
Fig. 7.7. Selection of the size of population when applying GA.....	117
Fig. 7.8. Channel rate allocation for a GOP with 12 frames and 4 SNR scalable layers in each frame for Foreman QCIF.....	118
Fig. 7.9. PSNR comparison of the proposed UEP scheme against other two schemes...	119
Fig. 7.10. Performance of the proposed method with mismatch.....	121

List of Tables

Table 4.1: Bit rate and the PSNR value from MPEG GOP structure and this work for three sequences: Mother & daughter QCIF, Foreman QCIF and Stefan QCIF.....	57
Table 4.2: Number of frames to be transmitted over the network.....	59
Table 4.3: Detailed comparisons of bits to be sent over the network between the proposed and the conventional structures.....	61
Table 4.4: Affected frames according to different distribution of slice losses.....	63
Table 5.1: The coding efficiency of the proposed interleaving scheme. QP=20 in the experiments.....	75
Table 6.1: Pseudo-code of resynchronization marker assignment algorithm. M is the number of MBs in each bitplane and T_l is the number of bits in bitplane l . The variables $\vec{N}_{opt}$, $\vec{N}_{last}$, and $\vec{N}_{temp}$ are vectors that store redundancy assignments.....	90
Table 6.2: The average slice size in each bitplane for three-bitplane scalability for Foreman CIF and Coastguard CIF (in MBs).....	91

Chapter 1

Introduction

1.1 Motivation

Recent advances in technology have resulted in a significant growth in wireless communications and the widespread access to information via the Internet, which have resulted in a strong demand for reliable transmission of video data.

Fig. 1.1 shows the basic block diagram of a video transmission system. Input video is compressed by the video encoder to generate a compressed bitstream, which is converted by transport coder into data units suitable for transmission over communication channels. Typical operations carried out in the transport coder include channel coding, framing of

Fig. 1.1. Video transmission system.

data, modulation, and control operations required for accessing the communication channel. At the receiver side, the inverse operations are performed to reconstruct the video signal for display.

Since the amount of bandwidth available on current communication channels is limited, low or very low rate compression algorithms have to be applied to the video data before it can be transmitted. Due to the application of predictive coding techniques, the effect of channel errors on the video can be extremely severe when compressed video data is transmitted over error-prone channels (see Fig. 1.2). A single bit error can cause substantial degradation if there is no action taken at either coder or decoder. The loss of entire compressed video data packets is even more serious. Real-time video streaming applications are also sensitive to transport delay because a packet of compressed video arriving too late to the decoder will be useless and regarded as a lost packet.

Fig. 1. 2. Error propagation.

1.2 Objective and Contributions

To alleviate the effect of transmission error, the video applications have to provide sufficient robustness, i.e., to have *error control*, to ensure that the quality of the decoded video is not overly affected by the channel unreliability. This is also the main objective of the thesis.

Normally error control methods can be categorized into 3 classes:

1) *Error resilience*. In such techniques, the encoder plays the primary part in improving the error robustness, typically by introducing redundancy in the transmitted information.

2) *Error protection*. When the transport channel is not lossless, there are two kinds of distortion observed at the decoder. The first is the quantization noise introduced by the waveform coder. The second is the distortion due to transmission errors. Transport coder (including forward error correction (FEC), packetization, and transport protocols) should

be designed to minimize the distortion due to transmission errors. Essentially, they all add a controlled amount of redundancy in the transport coder.

3) *Error concealment*. In such techniques, the decoder plays the primary part and does concealment of errors by estimation and interpolation using information it has already received. All these methods make use of the correlation between a damaged MB and its adjacent MB's in the same frame and/or the previous frame to accomplish error concealment.

This thesis will focus on the first two classes of error control and the contributions are summarized as follows:

1) In the first part, which includes Chapter 3, Chapter 4 and Chapter 5, several efficient error-resilience methods for *non-scalable* video coding have been proposed, including content-based resynchronization marker insertion, error resilient GOP structure and optimal MB interleaving.

2) In the second part, which includes Chapter 6 and Chapter 7, one efficient error-resilience method (optimal resynchronization) and one error protection method (GOP-based unequal error protection (UEP)) are proposed for *scalable* video coding.

The brief description of the proposed methods will be given below in Section 1.3, which outlines the thesis.

1.3 Outline of the Thesis

This thesis is organized as follows:

Chapter 2 A review of the existing error control methods is presented.

Chapter 3 Using MB-based segmentation results, we propose a framework for the positioning of resynchronization markers such that the image quality of foreground can be improved at the expense of sacrificing the unimportant background.

Chapter 4 We propose a novel GOP structure for robust transmission of MPEG video bitstream. By selecting the optimal position of the I frame in a GOP, robustness can be achieved without reducing any coding efficiency. Another advantage of the proposed GOP structure is also analyzed: compared with the conventional GOP structure, it provides reverse-play operation for MPEG video streaming with much less requirement on the network bandwidth.

Chapter 5 We propose a novel interleaving scheme for robust transmission of video sequences. By optimally grouping MB's into slices, robustness can be achieved with negligible reduction in coding efficiency. We group MB's into slices considering the maximal minimal distance of these MB's. Since every lost MB has, as many as possible, spatial neighbors that belong to other correctly received slices, an error-concealment mechanism has a lot of information it can employ for efficient concealment.

Chapter 6 A framework is proposed for the positioning of resynchronization markers for FGS video such that the resynchronization markers can be optimally inserted in each bitplane according to the importance of the bitplane and the time-varying channel condition. The effectiveness of this scheme lies in reducing loss of information based on significance of each bitplane of FGS video streaming.

Chapter 7 We address the problem of UEP for scalable video transmission over wireless packet-erasure channel. Unequal amounts of protection are allocated to the different frames (I- or P-frame) of a group-of-pictures (GOP) and in each frame, unequal

amounts of protection are allocated to the progressive bit-stream of scalable video to provide a graceful degradation of video quality as packet loss rate varies. We use a genetic algorithm (GA) to quickly get the allocation pattern, which is hard to get with other conventional methods, like hill-climbing method.

Chapter 8 A conclusion is presented and future research direction relative to this thesis is proposed.

Chapter 2

Literature Review

As described in Chapter 1, error control techniques can be divided into three categories, error resilience, error protection and error concealment, which are discussed respectively in the following.

2.1 Error Resilience

In this approach, the encoder operates in such a way so that transmission errors on the coded bitstream will not adversely affect the decoder operation and lead to unacceptable distortions in the reconstructed video quality. Compared to coders that are optimized for coding efficiency, coders with error resilience are normally less efficient because more bits are used to obtain the same video quality in the absence of any transmission errors. The extra bits that used for error

resilience are called redundancy bits, and they are applied to enhance the video quality when bitstream is corrupted by transmission errors. The design goal in error resilient coders is to achieve a maximum gain in error resilience with the minimum amount of redundancy. As all video codecs developed over time, H.264 also includes a number of error-resilience tools [63]. As in previous video coding standards, their application and adaptation is chosen by the encoder, many of those tools were present in previous video compression schemes.

2.1.1 Mitigation of Spatial Error Propagation

Because the multiplexed video bitstream consists of variable length code (VLC) words, a single bit error often causes a loss of synchronization and a series of erroneous code words at the decoder. This is called as *spatial error propagation*. Residual redundancy in non-compact VLC's can be used to design self-synchronization codes in order to obtain valid symbols again after some slippage [1]. However, even if resynchronization is regained quickly, the appropriate location of the decoded information within the video frame is impossible to be known because the number of missing symbols may not be decided. What's more, the subsequent code words are useless if the information is encoded differentially [120] [121]. For example, intra prediction in H.264 and inter prediction for motion vectors in most video coding standards are both the case. Therefore, some error resilience techniques to stop spatial error propagation are introduced.

1) Resynchronization.

Fig. 2.1. Resynchronization. (a) H.263 packet approach and (b) MPEG-4 periodic approach.

In this method, unique synchronization code words are inserted into the bitstream in regular intervals (see Fig. 2.1). Bit patterns of resynchronization markers are specially designed and placed at approximately regular intervals in video bitstream. The function of these markers is to divide the compressed video bitstream into independent segments. When decoding, by searching for these markers, the decoder can reliably locate each segment without actually decoding the packet. Therefore it prevents error propagation across separated segments by these markers.

Generally each segment of the bitstream contains one or several complete logical entities of video information (i.e., blocks, MB's, etc.). Thus it helps to minimize decrease in coding efficiency due to not exploiting dependencies between segments. Length of each segment is usually chosen to achieve a tradeoff between the overhead to encode the markers, and reliability to detect errors [119].

One of the resynchronization approaches adopted by H.263 is shown in Fig. 2.1 (a). In H.263 packet approach, the group of blocks (GOB) header contains a GOB start code, which contains information allowing the decoding process to be restarted. This packet approach is based on spatial resynchronization, in which once a specific MB location is reached in the encoding process, a resynchronization marker will be inserted into the bitstream. The disadvantage of this approach is that since the encoding process is at variable rate, these resynchronization markers will be unevenly spaced in the bitstream. Therefore, some parts of the frame, such as areas with high motion, will be more susceptible to errors and more difficult to conceal as well.

MPEG-4 can use the above slice structure. Another option adopts periodic approach (see Fig. 2.1 (b)), which is different from the packet approach utilized by H.263. By contrast, this periodic approach is based on providing periodic resynchronization markers throughout the bitstream. That is, the length of the video packets is not determined by the number of MB's, but by the number of bits contained in that packet. If the number of bits contained in the current video packet is more than a predetermined threshold, a new video packet will be created at the start of the next MB [119].

Because any conditional encoding across the resynchronization point must be forbidden, the resynchronization header normally provides anchor values for absolute location in the frame or present quantization parameter (QP) step size. Also, relatively long synchronization code words are normally used to reduce the probability of accidental emulation in bitstream [3] [4], although the length of the synchronization code word can be minimized [2] [121].

2) Robust Entropy Coding.

Fig. 2.2. RVLC.

In this category, Redmill and Kingsbury have developed error-resilient entropy coding (EREC) [5], in which bitstreams with variable length from individual blocks are distributed into slots with equal size. Firstly, the coded data for each block is fully or partially put into designated slot for the block. After that, a predefined offset sequence is applied to search for empty slots to put remaining bits of blocks, which are larger than the slot size, until all the bits are placed into one of the slots [5]. The decoder regains synchronization at the beginning of each block with the use of EREC, which also ensures that bits at the start of each block is more robust to error propagation than those in the end. Therefore, error propagation is predominant only in higher frequency coefficients [5] [121].

Another robust entropy coding method is Reversible Variable Length Coding (RVLC). With RVLC, the VLC words are designed such that they can be decoded both in the forward as well as backward direction (see Fig. 2.2). Intelligently designed RVLC's and corresponding decoding methods can significantly improve the error robustness of bitstream with little or no loss of coding efficiency [6][7] [121]. To make full use of the error localization properties of the RVLC's, the syntax for the MB layer needs to be changed as to group together all data coded with one RVLC table. Thus the backward decoding will not be blocked by nonreversible code words or reversible code words from

Resynchronization Marker	MB Address	QP	HEC	Header and Motion Data	Motion Marker	Texture
--------------------------	------------	----	-----	------------------------	---------------	---------

Fig. 2.3. Data partitioning.

another table. Note that data partitioning is often used in conjunction with RVLC because it provides the necessary syntax change for the applications of RVLC's by grouping together code words for the same type of information for all the MB's in a packet and placing markers between different logic units [119].

3) Data Partitioning.

To achieve better error isolation in the video packet and synchronization approaches, MPEG-4 introduced data partitioning to further improve the ability of the decoder to localize errors and to enhance error concealment capability [8].

When the data partitioning is used, video bitstream between two consecutive resynchronization markers is partitioned into smaller logic units (see Fig. 2.3). Each logic unit contains one type of information (e.g., motion vector or texture information) for all the MB's in the whole packet. For example, it can separate the data within a video packet into a motion part and a texture part (DCT coefficients) with a unique Motion Marker (MM). All the syntactic elements of the video packet that related to motion information are placed in the motion partition and all the remaining syntactic elements that related to the DCT data are placed in the texture partition. Thus, if the texture information is lost due to some errors, data partitioning helps to save the motion information, which can then be used to conceal the errors in a more effective way [121]. This is in contrast to the non-

data-partitioned method, in which each MB contains its own header, motion, and texture data.

For the decoder to locate each logic unit, MMs are placed between logic units as shown in Fig. 2.3. Unlike the resynchronization marker, which needs to be free of emulation from header, motion, and DCT data, these MMs need only to be free from emulation by data that immediately proceed them in the logic units. For instance, the marker between motion data and DCT data needs only to be free from emulation by motion data, but not DCT data [119].

After the decoder detects an error with the use of the data partitioning syntax, it can then search for the next MM in the packet and begin to decode the next logic unit within the same packet. More data can be salvaged and utilized since the decoder only needs to discard the rest of the logic unit, instead of the rest of the packet. Thus, each correctly decoded logic unit contains one type of information for all MB's in the packet and therefore the task of error concealment is made much easier. On the contrary, the decoder would need to compensate for the lost of header, motion and DCT data for all MB's from the packet in which the error is detected if data partitioning is not applied [119].

2.1.2 Mitigation of Temporal Error Propagation

Temporal error propagation happens due to the recursive structure of the decoder when it operates in the interframe mode. In such a mode, the previously decoded frame is used as a reference for the prediction of the current frame. Therefore errors remaining after concealment will propagate to successive frames and remain visible for a period of time. Moreover, errors happening on each frame could accumulate and result in even poorer

image quality although effects of individual errors are small. In general, three basic approaches are eligible to alleviate errors from the prediction loop, once they are introduced:

1) Intra Update.

A reliable method to stop temporal error propagation is to regularly insert I-frames when coding video sequences. Unfortunately, for the same picture quality, I-frames typically require several times more bits than P-frames, which are encoded with reference to previous frames. Therefore, the use of I-frames should be restricted to the minimum, especially for low bit-rate applications. In addition, additional delay in the order of several frame intervals is introduced because the regular insertion of I-frames leads to a significant variation in bit rate, which must be smoothed with the help of buffering to achieve constant bit rate transmission. To avoid this, INTRA coding can be distributed over several frames such that a certain percentage of each frame is updated.

The determination of the INTRA percentage is a non-trivial issue that is influenced by the error characteristic of the channel, the decoder implementation, and the encoded video sequence. In a common scheme, each MB is assigned a counter that is incremented if the MB is encoded in interframe mode. If the counter reaches a threshold, the MB is encoded in INTRA mode and the counter is reset to zero. By assigning a different initial offset to each MB, the update time of MB's can be distributed uniformly within the INTRA update interval.

Other INTRA update schemes have been proposed in the literature to further improve the efficiency of INTRA coding. In [9] and [10] it has been shown that it is advantageous to consider video content when deciding on the frequency of the INTRA coding. INTRA

update can also be included in a rate-distortion optimized encoding framework as proposed in [11]-[16] [112]. For example, an end-to-end approach to R-D optimized INTRA update was proposed in [112]. In this end-to-end scheme, three factors were identified to have impact on the constructed video quality at the receiver side: 1) the source behavior, e.g., quantization and packetization; 2) the channel characteristics; and 3) the receiver behavior, e.g., error concealment. A theory for globally optimal mode selection (INTER/INTRA) was developed based on the characterizations. With considering the network congestion status and the receiver behavior, this scheme is illustrated to outplay the classical approaches for Internet video applications [115].

2) Error Confinement.

Another method, which is called as independent segment decoding (ISD), splits a frame into several regions and a region in this frame can only be predicted from the same region in the previous frame [17]. For example, in MPEG-4, video object planes (VOP's) with arbitrary shape can be independently encoded and then superimposed at the decoder. Thus if an error happens in the bitstream of one VOP, the interframe error propagation is confined to that VOP since other VOP's will not refer to it when having motion compensated prediction [118].

In this method, the video sequence is normally partitioned into independent subvideos, which have also been included in H.263 as an optional extension. In the following, we assume each subvideo (segment) as a GOB for simplicity. When ISD is used, each GOB is individually encoded as a subvideo, which is independent from other GOB's. In particular, all GOB boundaries are treated just like picture boundaries. Because image content outside the current GOB could not be used for prediction, the ISD

Fig. 2.4. Error confinement.

mode significantly reduces the efficiency of motion compensation, particularly for vertical motion. The ISD mode is therefore often combined with the unrestricted motion vector (UMV) mode to reduce the loss of coding efficiency [118]. In this way, it allows motion vectors pointing outside the coded picture area by extrapolating the image borders.

When transmission errors occur, the ISD mode assures that errors inside a GOB will not propagate to other GOB's, as can be seen in Fig. 2.4. It is important to note that the ISD mode itself does not solve the problem of temporal error propagation, which must be combated by feedback-based INTRA updates, or by the use of reference picture selection (RPS).

3) Reference Picture Selection (RPS).

Rather than INTRA update at the encoder to stop temporal error propagation at the decoder, the coder could also encode the current frame with reference to a previous frame that has already been successfully decoded [18] [19]. This is called RPS, which can lower the excess bit-rate due to NACK-induced INTRA coding [109], [110].

H.263 has included RPS as an option, which can be operated in two different modes. In the *negative acknowledgment mode*, when the encoder receives only negative acknowledgments, the operation of the encoder is not changed if there is no error happening. Therefore the GOB's of the previous frame will be used as a reference. If a transmission error is detected, the decoder sends an NACK for an erroneous GOB and thereby requests older intact frames to provide a GOB as the reference. The advantage of the RPS mode is that it increases coding efficiency than simply switching to INTRA mode. This is because that fewer bits are needed for encoding the motion-compensated prediction error than for the video signal itself although there are several frame intervals between the current frame and the reference frame [118].

In the *positive acknowledgment mode*, all correctly received GOB's are acknowledged (which means it requires feedback) and the encoder uses only these GOB's as the reference. Thus the coding performance decreases in the case of no error happening because the encoder has to use older reference pictures for motion-compensated prediction with increasing round-trip time. The advantage of this mode is that error propagation is totally avoided since only pictures without errors are used for prediction [118].

2.1.3 Multiple Description Coding (MDC)

In the most common implementation, a *multiple description* (MD) coder generates two descriptions with equal rate and equal importance so that each description alone provides low but acceptable quality and both descriptions together lead to higher quality. The two descriptions are individually packetized and sent through either the same or separate

Fig. 2.5. A multiple description coding communication system.

physical channels as shown in Fig. 2.5. As long as the two descriptions are not simultaneously affected by packet losses, an acceptable quality can be secured. In the more general case, more than two descriptions with or without identical rates can be generated.

One main reason for the increasing popularity of MDC is that it can provide acceptable quality without retransmitting any lost packets (unless the loss rate is very high). This has two important implications. On one hand, it makes MDC particularly appealing for real-time interactive applications such as videophone and video conferencing, for which retransmission is usually not acceptable because it incurs unacceptable time delay. On the other hand, it makes the network design much simpler. That is, all the packets can equally be treated and no feedback or retransmission is required [121].

However, the above advantages of MDC come at a price: to achieve the same picture quality as a conventional coder does in the absence of any transmission errors, the MD coder uses more bits, which are intentionally inserted to make the bitstream more resilient to transmission errors. Therefore, the primary objective in designing an MD coder is to minimize the amount of extra bits while meeting an end-to-end distortion requirement if transmission loss is taken into account [121] [117].

There exist some unique challenges to design MC coders. As is well known, motion-compensated prediction, which can effectively exploit the temporal correlation between video frames, is popularly applied in all video coding standards. As a result, most MD video coders have to incorporate motion-compensated prediction. Unfortunately, if error occurs during transmission, whenever an encoder uses a signal for prediction that is unavailable to the decoder, a phenomenon, which is called as mismatch, occurs. Based on this observation, MDC is designed around the principle that information may be lost during transmission and therefore mismatch is a primary design concern in MD video coders [117]. As efficient predictors of MDC are more likely to introduce mismatch, a main technical challenge is to balance prediction efficiency with mismatch.

A variety of practical MD compression algorithms have been proposed during recent years: *subsampling* either in the spatial, temporal, or frequency domain, *MD quantization*, and *MD transform coding (MDTC)*.

MD subsampling decomposes the original signal into several subsets, either in the spatial, temporal, or frequency domain, where each subset corresponds to a different description. MD subsampling algorithms normally take advantage of the fact that video data samples that are spatially or temporally adjacent are correlated. Therefore one

description can be estimated from another [117]. Algorithms in this category ranges from temporal frame interleaving [18], to spatial pixel interleaving [20] [21], to transform coefficient interleaving [22], [23], to optimal partitioning strategies [24].

In MD quantization algorithms [25]-[32], two indexes are assigned to the descriptions of a quantizer. After that, the reconstructed signal is estimated by the MD decoder based on the received index(es). The assignment of indexes to each quantization bin can be designed to control the redundancy introduced by an MD scalar quantizer (MDSQ) and the corresponding side distortion.

In MDTC [33] [34], a controlled amount of redundancy between two sets of coefficients is introduced by a correlating transform. Within each description, coefficients should be made uncorrelated to maximize coding efficiency. At the decoder, missing coefficients can be estimated from the received ones. For instance, the pairwise correlating transform (PCT) [33] transforms a pair of random variables with the help of a non-orthogonal linear transform. Each of the two outputs lies in one of two non-orthogonal subspaces, respectively. The optimal transform with minimal side distortion for a given redundancy is parameterized by a single variable, which controls the amount of correlation introduced, which in turn controls the redundancy and the side distortion [117].

2.1.4 Scalable Coding

Scalable coding, or layered coding, is used to code a video into a base layer and one or several enhancement layers. The base layer provides a low but acceptable level of quality, and each additional enhancement layer will incrementally improve the picture

quality. By itself, scalable coding is a way to enable users with different bandwidth capacity or decoding powers to access the same video at different quality levels. To serve as an error resilience tool, scalable coding must be paired with unequal error protection (UEP) in the transport system, so that the base layer is protected more strongly.

There are many ways to divide a video signal into two or more layers in the standard block-based hybrid video coder. Usually in MPEG and H.263, there are three options, which are known as temporal, spatial, and SNR scalability, respectively. For simplicity, DCT-based system is used in the following to briefly introduce these three classes of scalability.

1) Temporal Scalability

Temporal scalability is defined to represent the same video content in different temporal resolutions or frame rates. Typically, temporally scalable video is encoded with use of temporally up-sampled pictures from a lower layer as a prediction in a higher layer.

2) Spatial Scalability

Spatial scalability is defined to represent the same video content in different spatial resolutions or sizes. Spatially scalable video is efficiently encoded with use of spatially up-sampled pictures from a lower layer as a prediction in a higher layer.

3) SNR Scalability

SNR scalability is defined to represent the same video in different SNR or perceptual quality. To be specific, SNR scalable coding quantizes the DCT coefficients to different levels of accuracy by making use of different quantization parameters. The resulting

streams have different SNR or quality levels. In other words, the smaller the quantization parameter is, the better quality the video stream can achieve.

In scalable video coding, each video representation has different significance and bandwidth requirement. The base layer is more important, while the enhancement layer is less important. The base layer needs less transmission bandwidth due to its coarser quality and an enhancement layer requires more transmission bandwidth due to its finer quality [116]. As a result, temporal/spatial/SNR scalability yields bandwidth scalability, with which the same video content can be transported at different rates.

In scalable coding, the different scalable layers can be transmitted in different bitstreams, which are called *sub-bitstreams*. On the other hand, they can also be transmitted in an *embedded bitstream*. The difference between embedded bitstream and layered bitstream is that embedded bitstream is formed by interleaving the base layer with the enhancement layer(s). Application-aware networks can select certain layers, which are less important, from an embedded bitstream and discard them to match the available bandwidth, so an embedded bitstream is also bandwidth scalable [116].

It is worthy to point out that only basic scalable mechanisms, that is, temporal, spatial, and SNR scalability, are introduced above. There can be combinations of the basic mechanisms, such as spatiotemporal scalability. Moreover, other scalability mechanisms include frequency scalability, object-based scalability, fine granular scalability (FGS), wavelets and SVC techniques in MPEG and JVT [123].

2.2 Error Protection

The techniques in this category are implemented at the transport layer. Error robustness can be achieved by introducing redundancy, which helps to detect or correct the errors that may happen during transmission.

2.2.1 Forward Error Correction (FEC)

FEC is well known for both error detection and error correction in data communications. Since FEC has the effect of increasing transmission overhead and hence reducing usable bandwidth for the payload data, it must be used judiciously in video services that are very demanding in bandwidth but can tolerate a certain degree of losses. FEC has been studied for error recovery in video communications [35]–[40]. For example, in H.261, an 18-bit error-correction code is computed and appended to 493 video bits for detection and correction of random bit errors in integrated services digital network (ISDN). For packet video, because several hundred bits have to be recovered when a packet loss occurs, it is more difficult to apply error correction [120].

Among FEC codes, Reed–Solomon (RS) codes are a well-known class of block codes with good erasure correction properties. RS codes are described by (N, k, t) , where N is the length of the codeword which contains $k = N - 2t$ source symbols and $2t$ protection symbols. It is defined by the set of all polynomials $C(x)$ over $GF(2^m)$ with degree less than k . m is the symbol length in bits. In video communication, m is commonly equal to 8 because we usually access video in bytes. The total number of words N in the code must divide $2^m - 1$, but it is very easy to get an (n, k, t) RS code with codeword length

Fig. 2. 6 Reed-Solomon code.

$n \leq N$ by simply dropping redundancy samples by puncturing. Thus the codeword length is only restricted by the condition $n \leq 2^m - 1$. RS codes can be used to correct errors, erasures, or both. A (n, k, t) code will then be able to correct up to $2t$ erasures or t errors, or combinations thereof with each error counting as two erasures. It will be unable to recover the original k data symbols if more than t errors or $2t$ erasures occur. Fig. 2.6 shows an example of Reed-Solomon coding.

Since recovery is carried out entirely at the receiver, FEC approach can scale to arbitrary number of receivers in a large multicast group. In addition, due to its ability to recover any packets regardless of which packets are lost, it allows the network and receivers to discard some of the packets which cannot be handled due to limited bandwidth or processing power. As a result, FEC is also applicable to heterogeneous networks and receivers with different capabilities [115]. However, there are also some disadvantages associated with FEC, which are listed as follows.

a) It increases the transmission bit rate.

This is because FEC adds redundant packets to every original packet. In addition, the higher the loss rate is, the higher the transmission rate is required to recover from the loss. The higher the transmission rate is, the more congested the network gets, which

leads to an even higher loss rate. This makes FEC vulnerable for short-term congestion. However, efficiency may be improved by making use of unequal error protection (UEP), which will be introduced below.

b) It increases delay.

This is because: 1) an FEC encoder must wait for all packets in a segment (i.e., k video data packets in Fig. 2.6) before it can generate the redundant packets ($n-k$ FEC packets in Fig. 2.6) and 2) the receiver must wait for at least k packets of a block before it can playback the video segment. In addition, recovery from bursty loss, which is very common for wireless channels, requires the use of either longer blocks or techniques like interleaving. In either case, delay will be further increased. But for video streaming applications, the increase in delay may not be an issue since they can tolerate relatively large delay.

c) It is not adaptive to time-varying error/loss rate.

As analyzed above, if more than $n-k$ packets of a block are lost, FEC cannot recover any portion of the original segment. This makes FEC useless when the short-term loss rate exceeds the recovery capability of the code. On the other hand, if the loss rate is well below the recovery capability of FEC, the redundant information is more than necessary and a smaller ratio would be more appropriate. Feedback can be used to improve the adaptive capability of FEC. That is, if the receiver sends the loss rate to the source, the channel encoder can adaptively add the redundancy [115].

2.2.2 Unequal Error Protection (UEP)

A significant portion of previous research on FEC for video transmission has involved *equal error protection* (EEP), in which all the bits of the compressed video stream are treated equally, and given an equal amount of redundancy, regardless of their different sensitivity to errors and their contribution to overall video quality. However, the compressed video stream typically does not consist of bits of equal significance. For example:

- a) In MPEG, an I-frame is more important than a P-frame while a P-frame is more important than a B-frame.
- b) In an MB, the header information and the motion vector are more important than the DCT coefficients.
- c) Different layers of scalable coding are not equally important since all the enhancement-layers will be useless if the base-layer is lost.

Therefore, current research is heavily weighted toward *unequal error protection* (UEP) schemes, in which the more significant information bits are to be given more protection.

A representative work of UEP is the priority encoding transmission (PET) [89]. An important feature of the PET scheme is to set different priorities of error protection for different segments of the video bitstream. As a result, PET is very efficient and suitable for transporting MPEG video, which has an inherent priority hierarchy (i.e., I-, P-, and B-frames).

UEP is also adopted in the scalable video coding. As stated above, for the scalable video transmission, the base layer is the most important layer, and it must be transmitted

correctly. Otherwise, the encoded video cannot be reconstructed even if all the enhancement layers are correctly received. Therefore the base layer will be assigned a large amount of FEC to make sure it can be correctly decoded. And less FEC will be assign to the enhancement layers, since they contribute less to the overall video quality. For example, Tan and Zakhor proposed a receiver-driven hierarchical FEC (HFEC) [111] to provide error recovery in layered multicast video. In HFEC, additional streams with only FEC redundant information are generated along with the video scalable layers. Each of the FEC bitstreams is used for recovery of a different video scalable layer, and each of the FEC bitstreams is sent to a different multicast group. Higher level of protection can be achieved by subscribing to more FEC groups. As a result, HFEC achieves good tradeoff between flexibility of providing recovery and bandwidth efficiency [111].

It is important to note that UEP can also be used together with automatic repeat request (ARQ), which is called hybrid ARQ-FEC, to achieve even better performance [105]-[107]. For instance, in [107], Kallel proposed a method that combines the ARQ protocols with adaptive FEC. The basic idea of this method is to adjust the coding rate of error correction according to the system parameters, such as SNR, round trip delay and buffer size to maximize the overall efficiency. In [122], the authors take a theoretical approach to study a basic multicasting problem with incremental FEC.

Localization or region of interest (ROI) scalability is worth mentioning here because it is pre-processing before having UEP when we consider about region of interest like foreground. More important region (ROI) will be added more protection while the less important region will be added less protection [124]. Chapter 3 will give more explanation about this.

2.2.3 Joint Source and Channel Coding (JSCC)

Due to Shannon's separation theorem, coding was generally divided into source coding and channel coding. Source coding was concerned with developing efficient source coding techniques while channel coding was concerned with developing robust channel coding techniques. However, Shannon's separation theorem is not strictly applicable when the delay is bounded, which is the case for real-time video applications [115]. The motivation of *joint source and channel coding* (JSCC) for video comes from the following observations:

1) According to the rate-distortion theory, the lower the source coding rate for a video unit is, the larger the distortion of the video unit is (see Fig. 2.7 (a)).

2) Assume that the total rate, which is the sum of the source coding rate and the channel coding rate, is fixed and channel loss rate does not change. The higher the source coding rate for a video unit is, the lower the channel coding rate would be. In this case, if the channel condition is severely bad, the overall distortion will be very large. On the other hand, the higher the channel coding rate for a video unit is, the lower the source coding rate would be. This will lead to a large source distortion.

From the above observations, it can be argued that, given a constant total rate, there exists an optimal source coding rate (or channel coding rate) that achieves the minimum distortion as can be seen in Fig. 2.7 (b) [115].

The objective of JSCC is therefore to find the optimal source/channel coding rate, design source and channel coding schemes to achieve the optimal rates. The use of JSCC techniques ranges from bit allocation between source and channel coding, to layered coding and UEP, through channel-optimized vector design (COVQ) design, to systems

(a)

(b)

Figure 2. 7 Rate-distortion relation for (a) source coding and (b) joint source and channel coding.

that iteratively optimize the source and channel coders. Normally they can be classified into two categories for different systems that they are implemented with:

1) Subband Coding Systems.

The JSCC methods in this category include [41]-[46]. Cheung and Zakhor [41] propose applying RCPC channel codes to a 3-D subband coder, which makes use of layered quantization for successive refinement. They use different channel coding rate for different bitplane in every subband. They derive a constrained bit allocation objective function, which they minimize by making use of distortion-rate curves. Srinivasan and Chellappa [43] examine a similar subband system. The difference is that they apply a finite-state Markov model to emulate fading channels. Kondi, *et al.* [46] discuss the problem for motion-compensated DCT-based SNR scalable coding. In their scheme, the available bit rate is allocated between all scalable layers and between source and channel coding within each layer.

2) DCT-Based Systems.

Early work on JSCC of DCT coded images is done by Modestino *et al.* [47], who study UEP for the indexes of the transform coefficients. They show that it is better to allocate more bits to the channel coder and fewer to the source coder as the channel bit error rate increases. Vaishampayan and Farvardin [48] propose a different approach, in which they design channel optimized scalar quantizers for the DCT coefficients. Implicit in this approach is index assignment. After obtaining the distortion-rate curves, they have bit allocation among the different coefficients. They also investigate performance under mismatched channel conditions and find that the noisy channel design provides substantial robustness. Marcellin *et al.* [49] combine trellis-coded quantization (TCQ) with DCT based image coding, in which the use of residual redundancy could be straightforward [114].

2.3 Error Concealment

Error concealment has a long history and it has been available from H.261 and MPEG-2. It means that decoder recovers or estimates lost information due to transmission errors. Given the block-based hybrid coding paradigm, three types of information may need to be estimated in a lost or erroneous MB: the *texture information*, which consists of the pixel or DCT coefficient values; the *motion information*, which includes MV for a MB that is coded in either P- or B-mode; and finally the *coding mode* of the MB [120] [121].

As is well known, there are predominantly low frequency components in images of natural scenes. For example, the color values of spatially and temporally adjacent pixels vary smoothly, except in regions with edges. This smoothness property of image/video signals has been used by all the error concealment techniques that have been developed for recovering texture information. Essentially they all perform some kind of spatial or temporal interpolation. The MV field, to a lesser extent, also has the smoothness property and can also be recovered by applying spatial or temporal interpolation. Regarding the coding mode information, the methods that have been developed tend to be more driven by heuristics [121]. In the following, some representative techniques for each category are reviewed, respectively.

2.3.1 Recovery of Texture Information

1) Motion Compensated Temporal Prediction.

A simple whereas very effective approach to recover a damaged or lost MB in the decoder is to copy the corresponding MB in the previously decoded frame, based on the

MV for this MB. In this approach, the recovery performance is critically influenced by the availability of the MV [121]. If the MV is also missing, it must first be estimated, which is discussed below.

2) Spatial Interpolation.

Another simple approach is to interpolate pixels in a damaged block by making use of pixels in adjacent correctly received blocks. The only available neighboring blocks are usually those in the current row and the row above because all blocks or MB's in the same row are put into the same packet. As most pixels in the available blocks are too far away from the missing samples, usually only the boundary pixels in these blocks are used for spatial interpolation [51].

Instead of interpolating individual pixels for the lost block or MB, a simpler approach is to estimate the DC coefficient of the damaged block and replace the damaged block by a constant. The constant is equal to the estimated DC value, which can be estimated by averaging the DC values of surrounding blocks or by the α -trimmed mean of pixels in the neighboring blocks [52]. It is worthy to point out that an interleaved packetization mechanism can be used to facilitate spatial interpolation so that the loss of one packet will damage only every other blocks or MB's in a row [120] [121]. Note that interleaving reduces compression performance if spatial prediction is used.

3) Spatial and Temporal Interpolation by Maximizing the Smoothness of Resulting Video.

A problem with spatial interpolation approaches is how to determine an appropriate interpolation filter. Moreover, spatial interpolation approaches mentioned above ignore the use of received DCT coefficients, if any. These problems are resolved in the approach

of [53], which requires the recovered pixels in a damaged block to be smoothly connected with its neighboring pixels both temporally in the previous or following frames and spatially in the same frame. If some but not all DCT coefficients are received for this block, subject to the constraint that the DCT on the recovered block would produce the same values for the received coefficients, the estimation should make the recovered block as smooth as possible. The above objectives can be formulated as an unconstrained optimization problem, the solutions of which correspond to different interpolation filters in the spatial, temporal, and frequency domains [52] [53] [121].

More sophisticated error concealment methods include a combination of spatial/spectral and temporal interpolations with motion vector estimation [54]. Object-based error concealment [55] uses *a priori* information for concealment through a principal component analysis on models.

4) Spatial Interpolation with Projection onto Convex Sets (POCS).

Another way to accomplish spatial interpolation is by using the POCS method [56], [57]. The general idea behind POCS-based estimation methods is to formulate each constraint about the unknowns as a convex set. The optimal solution is the intersection of all the convex sets, which can be obtained by recursively projecting a previous solution onto individual convex sets. The spatial smoothness criterion is formulated in the frequency domain by requiring the discrete Fourier transform (DFT) of the recovered block to have energy only in several low frequency coefficients, when applying POCS for recovering an image block. The DFT coefficients are required to be distributed along a narrow strip orthogonal to the edge direction, i.e., low-pass along the edge direction and all-pass in the orthogonal direction, if the damaged block is believed to contain an edge in

a particular direction. The requirement on the range of each DFT coefficient magnitude can also be converted into a convex set, so is the constraint imposed by any received DCT coefficient. This approach may not be suitable for real-time applications because the solution can only be obtained through an iterative procedure [120] [121].

2.3.2 Recovery of Motion Vectors

For estimating lost MV's, there are several simple operations [59]:

- a) Setting the lost MV's to be zeros, which works well for video sequences with relatively small motion;
- b) Applying the MV's of the corresponding block in the previous frame;
- c) Applying the average of the MV's from spatially adjacent blocks;
- d) Applying the median of MV's from spatially adjacent blocks;
- e) Re-estimating the MVs [52].

Typically, when an MB is damaged, its horizontally adjacent MBs are also damaged, and hence the average or mean is taken over the MVs above and below. It has been found that the last two methods produce the best reconstruction results [52] [120] [121].

2.3.3 Recovery of Coding Modes

One way to estimate the coding mode for a damaged MB is to collect the statistics of the coding mode pattern of adjacent MB's and to find a most likely mode based on the modes of surrounding MB's [58]. A simple and conservative approach is to assume that the MB is coded in the intra-mode and use only spatial interpolation for recovering the underlying blocks [53] [120] [121].

Chapter 3

Content-based Resynchronization

Approach for Subjective Quality

Enhancement

3.1 Introduction

Previous video coding standards such as H.261 and H.263 (Version 1) logically partition each of the images to be encoded into rows of MBs called Group Of Blocks (GOBs) and resynchronization markers are allowed to occur only at the left edge of the images. Since the resynchronization markers are likely to be unevenly spaced in the bitstream, some

areas of the picture will be more susceptible to errors [65]. In MPEG-4 encoder, resynchronization markers can be inserted in the bitstream at approximately constant intervals. Thus, the MB interval between the resynchronization markers is a lot closer in the high activity areas and a lot farther apart in the low activity areas. In the presence of a short burst of errors, the decoder can quickly localize the error to within a few MBs in the important high activity areas of the image and preserve the image quality in these important areas [65].

Besides the conventional approaches mentioned above, there are still many literatures considering the insertion of resynchronization markers [66], [13], [67]. Yoo [66] proposes an adaptive resynchronization marker positioning (RMP) method for rapidly resynchronizing the VLC decoding synchronization. Taking into account the channel condition and the error concealment method, Cote, *et al* [13] optimize placement of synchronization markers in the compressed bitstream. Yang, *et al* [67] present simpler but efficient techniques to optimize placement of synchronization markers considering the impact of the data partitioning in error concealment.

In this chapter, using segmentation results, we propose a content-based resynchronization method, which is denoted as *content-based resynchronization*. That is, after pictures are segmented into foreground and background at the MB level, we design a novel resynchronization scheme to improve the subjective quality of video sequences.

The rest of this chapter is organized as follows. In Section 3.2, we first give a description of content-based resynchronization. Section 3.3 gives the properties of the scheme and Section 3.4 concludes the chapter.

Fig. 3. 1. Impact of error location on perceptual video quality according to different foreground data location in one slice.

3.2 Content-based Resynchronization

In Subsection 3.2.1, we first give a description of perceptual impact of error on video quality, after which, we propose a resynchronization approach (content-based resynchronization) using content-based segmentation results in Subsection 3.2.2. Subsection 3.2.3 gives simulation results.

3.2.1 Perceptual Impact of Error on Video Quality

Regardless of the resynchronization scheme being used, the foreground data may generally distribute in one slice (i.e., between two resynchronization markers) in four cases: in the beginning, in the middle, in the end, one part in the beginning and the other part in the end. Then we classify the location of error in three cases: in the beginning, in

the middle, in the end. Thus, there will be 4×3 different impacts on perceptual video quality according to the different positions where the foreground data distributes and where the error happens (see Fig. 3.1). Considering the best protection of the foreground data, we can summarize the perceptual impact of error on video quality as follows:

Best case: foreground in the beginning. As we can see from Fig. 3.1 (b), (f) and (j), when the error happens in the beginning, one part of foreground data will be lost. (Note that in this work, if an error is detected using the error detection techniques at the decoder, all the following bits until the next marker will be discarded.) However, in both cases that error happens in the middle or in the end, the foreground data will not be affected by the error at all.

Worst case: foreground in the end. In such a case, wherever the error happens in the bitstream of one slice, the foreground will be affected less more or less. The effect depends on the location where error happens. This is shown in Fig. 3.1 (c), (g) and (k).

Others: foreground in the middle/one part in the beginning and one part in the end. For these two cases, the foreground data will be less affected than in the worse case. We can see these two cases from Fig. 3.1 (a), (e), (i), (d), (h) and (l). For example, if the foreground data is in the middle, one part of foreground will not be dropped when the error happens in the middle and all the foreground data may not be affected when the error happens in the end. If one part of the foreground distributes in the beginning and the other in the end, a portion of the foreground data may survive from the error wherever the error happens. Of course, how much of foreground data will be discarded depends on the location of the error.

Fig. 3. 2. (a) Segmentation result and the spatial position of resynchronization markers of (b) H.263 (version 1) packet approach, (c) MPEG-4 periodic approach and (d) this work.

It is now clear that the impact of error on the perceptual video quality depends tremendously on the location of the foreground data in one slice. Below using the result of MB-based segmentation, which is shown in Fig. 3.2 (a), we propose a novel resynchronization scheme with which perceptual quality of video will be significantly improved. Note that the segmentation result is based on an MB-based algorithm. Normally, the segmentation is carried based on pixel level [68], [69]. Based on the result of pixel-level segmentation, our MB-based segmentation algorithm works like this: If the number of foreground pixels in one MB is noted as N , the decision for an MB to be either in foreground or in background can be made as follows:

$$MB_{i,j} = \begin{cases} \text{foreground MB, if } N \geq T \\ \text{background MB, if } N < T \end{cases} \quad (3.1)$$

where i and j are MB indexes. T is the threshold and in our experiment we set it to

$$\frac{3}{4} \times 16 \times 16.$$

3.2.2 Content-based Resynchronization Scheme

From Fig. 3.2 (b), we can see that for H.263 (version 1) packet approach, almost all the slices belong to the case that foreground data is in the middle of one slice. That means

that error may have a moderate effect on the foreground data, which leads to a moderate impact on the perceptual quality of the video sequence. On the other hand, for MPEG-4 periodic approach, which is shown in Fig. 3.2 (c), all the cases (i.e., best case, worst case and others mentioned above) will happen when error occurs. Intuitively, if we want the best case happen most of time, the simplest way is to put the resynchronization mark just before the foreground data, which is shown in Fig. 3.2 (d). (Note that for the first slice, the resynchronization marker is put at the beginning of one picture instead of the beginning of foreground data because the resynchronization marker in the first slice is also supposed to be the resynchronization marker of the whole frame.) Thus, wherever error happens, we can get the best case for all the slices.

Fig. 3.3 and Fig. 3.4 show two scenarios when error happens at different locations in one bitstream of a frame. The bitstream is generated with H.263+ encoder. In the first scenario as shown in Fig. 3.3, the error happens in the end of a GOB of H.263. If H.263 packet approach is used, the error can be isolated until the next resynchronization marker and a part of foreground data is lost. Unfortunately, if the MPEG-4 periodic approach is applied, the error propagation may not be stopped since the location of the next resynchronization marker may distribute spatially everywhere in the frame. As shown in Fig. 3.3 (b), a very bad case happens that a large amount of foreground data before the next synchronization marker will all be dropped. On the contrary, this phenomenon cannot happen for the approach proposed in this work because the next resynchronization marker can help the error propagation stop before the next part of foreground data. This is shown in Fig. 3.3 (c).

Fig. 3. 3. Impact of error on a QCIF image quality when the location of error is in the end of a GOB of H.263. (a) H.263 (version 1) packet approach, (b) MPEG-4 periodic approach and (c) the approach used in this work.

Fig. 3. 4. Impact of error on a QCIF image quality when the location of error is in the beginning of a GOB of H.263. (a) H.263 (version 1) packet approach, (b) MPEG-4 periodic approach and (c) the approach used in this work.

In the second scenario as shown in Fig. 3.4 (a)-(c), the error happens in the beginning of a GOB of H.263. For the approaches adopted by H.263 and MPEG-4, foreground data will be lost less or more while it will be protected well by the intelligent resynchronization marker insertion in our approach.

Aside from the analysis above, which is based on the error location in the bitstream of a whole picture, there is another way to verify the advantage of the proposed method. In this way, analysis is made based on the error location in the bitstream of each slice. Using L as the number of bits since the location of the last resynchronization marker plus the

Fig. 3. 5. Decoded Frame 93 of Foreman QCIF (a) without channel error, (b) this work with $BER=10^{-5}$, (c) MPEG-4 approach with $BER=10^{-5}$ and (d) H.263 approach with $BER=10^{-5}$.

number of bits of the current MB, and P_e as the bit error rate (BER), we determine the probability p , that current MB is lost, as

$$p = 1 - (1 - P_e)^L \quad (3.2)$$

from which we can see that, with small L , p will be small. That is to say, if we put the foreground data in the beginning of each slice, the error probability of foreground MBs will be decreased.

3.2.3 Simulation Results

In the experiment, 300 frames of the Foreman QCIF and Carphone QCIF at frame rate of 30 fps were encoded with H.263+ encoder to test our proposed scheme. The results of the approaches of H.263 and MPEG-4 are also given for comparison. To make the comparison fair, the number of resynchronization markers inserted was the same as shown in Fig. 3.2 (b), (c) and (d). Due to the random nature of losses and their impact on the compressed video, 100 different runs of experiments were conducted for the three resynchronization approaches. For each experiment, a different loss pattern was generated by a random number generator initialized each time with a different seed. We

Fig. 3. 6. Foreman QCIF (200Kbps) average PSNR comparison of (a) overall picture and (b) foreground.

Fig. 3. 7. Foreman QCIF (100Kbps) average PSNR comparison of (a) overall picture and (b) foreground.

use both subjective and objective performance measures to evaluate the performance of the three algorithms. One error concealment strategy is to replace the luminance and chrominance components of the erroneous MBs with the luminance and chrominance of the corresponding MBs in the previous frame of the video sequence. We use this concealment method in our simulation since it works simply and effectively although other methods can also be used.

Fig. 3. 8. Carphone QCIF (200Kbps) average PSNR comparison of (a) overall picture and (b) foreground.

The subjective quality of Foreman QCIF at bitrate of 200Kbps is shown in Fig. 3.5.

The results of this work and other two approaches are given with $BER=10^{-5}$. The figures clearly show that the area of interest is much improved with the proposed method, which is the result of inserting the resynchronization markers just before the foreground data. The degradation in the background region of the result from the proposed scheme is hardly noticeable since we concentrate on the foreground when looking at the whole picture.

To make the result more convincing, we will show much statistical objective data below. Plots displaying the average PSNR achieved from 100 runs of experiments are provided in Fig. 3.6-Fig. 3.9 for Foreman QCIF and Carphone QCIF at different bitrates, respectively. The results of this work, MPEG-4 approach and H.263 approach are given with BER from 10^{-6} to 10^{-4} . Fig. 3.6 (a)-Fig. 3.9 (a) show the overall PSNR comparison of three approaches. The results revealed that our scheme and the MPEG-4 scheme achieve a higher PSNR than the H.263 scheme. This is expected since in the approaches of this work and MPEG-4, the decoder can quickly localize the error to within a few MBs

Fig. 3. 9. Carphone QCIF (100Kbps) average PSNR comparison of (a) overall picture and (b) foreground.

in the important high activity areas of the image and preserve the image quality in these important areas. Fig. 3.6 (b)-Fig. 3.9 (b) show the average PSNR comparison of foreground for three approaches. Comparing with the other two schemes, our scheme achieves a PSNR improvement in foreground of up to 2dB. Furthermore, the improvement increases with the increase of BER. Although the PSNR of background decreases, our scheme achieves a significant improvement in subjective quality of the video sequences.

3.3 Properties of the Proposed Scheme

There are several fundamental issues in using the proposed scheme in this chapter.

First, the approach is particularly applicable for video telephony and video conferencing applications as they mostly exhibit head-and-shoulders sequences. For those that do not belong to header-and-shoulders sequences, foreground may only represent a small part of the whole picture. In such a case, we insert resynchronization marks in the

following way: for the portion of the picture including foreground, we use the proposed scheme. On the other hand, for the rest portion, we will adopt the periodic approach of MPEG-4 to preserve the image quality in the important high activity areas of the image against the burst errors.

Second, generally, the foreground is the more active area in one picture, so we actually insert resynchronization markers at a sub-uniformly spaced bit intervals in our proposed schemes. That holds firmly when the sequence is a head-and-shoulders one because the size of foreground area in each slice is almost the same.

Thirdly, it is important to notice that if the segmentation algorithm does not work as precisely as expected, or even fails, the result just degrades to the cases when we use the packet approach or the periodic approach. That is to say, our scheme will not take negative effect to the subjective quality of the video.

Last, the proposed scheme can be easily applied to the H.263+ standard or MPEG-4 standard.

3.4 Conclusion

In this chapter, we proposed a content-based resynchronization scheme so that we can have better protection on the areas of interest, i.e., foreground, resulting in better subjective video quality. The simulation results indicated that our proposed scheme encodes the video sequences with higher quality on the foreground as compared to that of the conventional approaches over error-prone channels with BER from 10^{-5} to 10^{-4} .

Chapter 4

Error Resilient GOP Structure with Efficient Reverse-play Functionality

4.1 Introduction

The MPEG video international standard [71] specifies the coded representation of the video data. In order to achieve the highly efficient compression and meet the conflicting requirements of random access, three picture types have been defined: 1) intra (I) picture, coded without reference to other pictures, 2) predictive (P) picture, coded with reference

Fig. 4.1. (a) Conventional MPEG GOP structure and (b) the proposed GOP structure.

to previous (I or P) picture, and 3) bidirectionally predictive (B) picture, coded with reference to an immediate previous (I or P) picture, as well as an immediate future (P or I) picture. The organization of the three picture types in a sequence is done around the concept of a GOP. The I frame at the beginning of a GOP serves as a basic entry point to facilitate random seeking or channel switching and also provides coding robustness to transmission error, but it is coded with only moderate compression to reduce the spatial redundancies. P frames are coded more efficiently using motion compensated prediction from a past I or P picture and are generally used as a reference for further prediction. B frames provide the highest degree of compression, but require both past and future

reference pictures for motion compensation. A GOP is defined by its length N_{GOP} , which is the distance between I pictures in a sequence. An example of a GOP in MPEG with $N_{GOP} = 6$ is presented in Fig. 4.1 (a), where I, P and B denote picture encoding in the intra picture mode, predicted mode and bidirectionally predicted mode, respectively.

In this chapter, we propose a novel GOP structure for robust video transmission. Instead of putting the I frame at the beginning of a GOP in conventional video coding standards, we insert the I frame in the middle of the GOP (see Fig. 4.1 (b)). That means the P frames previous to I frame in this GOP will be backwardly coded with this I frame as the starting reference for motion compensation. To achieve the optimal error-resilient performance, the best position of the I frame in each GOP will be analyzed. At the same time, we will see that putting I frame in the beginning of a GOP, as MPEG does, is actually the worst position selection in terms of error resilience. The proposed structure will also be demonstrated to improve the robustness of the steaming video without reducing any coding efficiency, which is highly desirable for bandwidth-limited networks.

Another advantage of the proposed GOP structure will be shown as well: compared with the conventional GOP structure, it provides reverse-play operation for MPEG video streaming with much less requirement on the network bandwidth. With the rapid growth of online multimedia content, the video streaming system is much anticipated to have the capability of providing fast and efficient browsing. Such a key technique is to provide full videocassette recording (VCR) functionality, which includes forward, backward, stop, pause, fast forward, fast backward and random access. Unfortunately, the MPEG video coding standards were designed for forward-play operations, so the predictive processing techniques employed in MPEG severely complicate the reverse-play operation. One

straightforward but naïve approach to implement a reverse play of MPEG compressed video is to decode the GOP up to the current frame, and then go back to decode from the beginning of the GOP again up to the next frame to be displayed. However, this requires high bandwidth of the network [72]. Some works on the implementation of reverse-play for MPEG compressed video streaming have been introduced [72]-[75]. However, these approaches require either much computation or extra decoder complexity. On the contrary, the proposed GOP structure in this chapter is simple but effective to provide reverse-play operation for MPEG video streaming. By putting the I frame in the middle of each GOP, it partitions the forward-predicted P frames into two parts: half are backward-predicted coded and half are forward-predicted coded. This structure will moderately make the reverse-play easy to handle by eclectically selecting the position of I frame in a GOP.

The rest of this chapter is organized as follows. Section 4.2 gives a description of the proposed GOP structure. Meanwhile, its robustness, coding efficiency and other relative properties are analyzed. In Section 4.3, we mainly discuss the efficient reverse-play functionality it causes. Section 4.4 gives the simulation results and Section 4.5 concludes the chapter.

4.2 Proposed GOP Architecture

4.2.1 Proposed Error Resilient GOP Structure

The proposed GOP structure is shown in Fig. 4.1 (b). For the purpose of legibility but without loss of generality, we will not include B frames in the following analysis since

Fig. 4.2. GOP structure without B frames of (a) MPEG and (b) this work.

they do not virtually affect the architecture. We show the GOP structures without B frames of MPEG and this work in Fig. 4.2. For the proposed GOP structure shown in Fig. 4.2 (b), after encoding the frame P_i (note that the value of i will be discussed below in the next subsection), the encoder jumps to encode the next I frame. After that, it encodes the uncoded P frames from P_n up to P_{i+1} in a backward manner. This procedure will repeat sequentially. Note that this I frame will be stored in the reference buffer as the reference for the P frames (P_1 to P_j) behind it and also for the P frames (P_n up to P_{i+1}) in front of it.

When decoding, the decoder will carry a reverse procedure of the encoding one. That is, after the decoder decodes to the frame P_i , it will decode the next I frame and then decode P frames in a backward direction from the frame P_n to the frame P_{i+1} and finally

the frames P_1 to P_j . For comparison, the coding procedure of MPEG is also illustrated in Fig. 4.2 (a).

4.2.2 Selection of Optimal Position of I Frame in a GOP

Now, let's study the optimal switching point between forward and backward predictions. Before the optimization problem, a certain model should be built to describe the effect of error on the decoded video quality. As well known, PSNR is the most popular criterion for video quality evaluation. Complex theoretical models considering the effect of error propagation in terms of PSNR [76]-[77] or MSE [16] have been proposed. However, such models involve complicated parameter estimation, which varies with complexities of both video content and video codec. On the other hand, the error propagation properties are analyzed with the criterion of the length of error propagation (LEP) [78], which is actually a simplified transformation of the model in [79], where the number of impaired MBs (MBs) is taken as a criterion. These two models are motivated by the fact that the fewer slices or MBs are corrupted, the better quality of the reconstructed video would be achieved. In order to investigate the error propagation, we have applied such kind of a model in this chapter since it works simply and effectively.

We assume that there are totally $n+1$ frames in a GOP with one I frame and n P frames (see Fig. 4.3). Then in our proposed architecture, the I frame can be put in any position in the GOP with P_1 to P_i behind and P_1' to P_{n-i}' in front of it. If a lost slice belongs to I frame, the length of the error propagation will be $n+1$ since the error will

Fig. 4.3. Selection of optimal position of switching point in the proposed GOP structure.

propagate to all the P frames at the both sides. If a lost slice belongs to one of forward predicted P frames, say P_k ($k \leq i$), the length of the error propagation will be $i - k + 1$. If a lost slice belongs to one of backward predicted P frames, say P_k' ($k \leq n - i$), the length of the error propagation will be $n - i - k + 1$. Thus, the average error propagation length

$L_{average}$ will be

$$L_{average} = (n+1)p + \sum_{k=1}^i (i-k+1)p + \sum_{k=1}^{n-i} (n-i-k+1)p$$

$$= (i^2 - ni + \frac{n^2 + 3n + 2}{2})p \quad (4.1)$$

where p is the average slice lost rate. By solving the following problem:

$$\min_{0 \leq i \leq n} L_{average} \quad (4.2)$$

we can get the optimal position of I frame in a GOP, i.e.,

$$i = \begin{cases} n/2, & \text{if } n = 2, 4, 6, \dots \\ (n-1)/2 \text{ or } (n+1)/2, & \text{if } n = 1, 3, 9, \dots \end{cases} \quad (4.3)$$

Thus, if n is an even number, we will put the I frame in the very middle of each GOP to achieve optimal robustness. If n is an odd number, we can have $i = (n-1)/2$ or $i = (n+1)/2$, but we will set $i = (n+1)/2$ in our proposed structure out of consideration of reducing system delay, which will be discussed below in the next subsection.

Fig. 4.4. Average length of error propagation vs. i . GOP length $n = 8, 9, 10, 11$, respectively.

Fig. 4.4 illustrates the relationship between average length of error propagation and i . We can see that the optimal selection of i by the new GOP structure minimizes the average length of error propagation. Meanwhile, it is clearly shown that putting I frame in the beginning of a GOP, as MPEG does, is actually the worst position selection, which leads to the largest length of error propagation.

4.2.3 Analysis of Coding Efficiency, System Delay and Complexity

Now, we will discuss some properties that are relative to the proposed GOP structure.

Fig. 4.5. (a)-(c): Number of bits used by conventional MPEG and this work of Mother & daughter QCIF, Foreman QCIF and Stefan QCIF, respectively. (d)-(f): Corresponding PSNR value of the three sequences.

1) Coding efficiency: The predictive coding explores the temporary dependency between consecutive frames, so no matter which frame, previous or future, the current picture takes as its reference, the dependency will exist, which only depends on the

predicting distance and the inherent characteristics of the video sequences. Thus, the coding efficiency of the proposed GOP structure will be comparable with that of MPEG GOP structure. Fig. 4.5 confirms our estimation. All sequences were coded in the same condition. Every I frame was followed by 10 P frames. The frame rate was 10 frames/s. The sequences were compressed using the same QP 20. As we can see from the figure, the number of bits used by MPEG and this work is almost the same. So is the PSNR value. Table 4.1 shows the statistical average values of bit rate and PSNR, which clearly demonstrate that the proposed GOP structure encodes the video sequences with higher quality while consuming fewer bits. The reason that the proposed GOP structure is more efficient can be analyzed in this way: for MPEG GOP structure, source error, which is caused by quantization, will accumulate from previous frames (I or P) to the following P frames in a GOP, but for the proposed GOP structure, the length of error accumulation is halved by inserting the I frame in the very middle of a GOP with P frames at each side encoded in forward and backward manners, respectively. This can be verified in Fig. 4.5, from where we can see that the frames, which save encoding bits or with higher PSNR value, are mostly the 5 P frames in front of each I frame in the proposed GOP. These 5 P frames in the proposed GOP structure are predictively encoded with less distance to the nearest I frames than those in conventional MPEG GOP structure.

2) System delay: For the proposed GOP structure, the coding and transmitting order of the sequence $P'_{n-i}P'_{n-i+1}...P'_1IP_2...P_i$ should be $IP'_1P'_2...P'_{n-i}P_1P_2...P_i$. Thus for displaying the frame P'_{n-i} at the receiver, there will be a maximal delay of totally $n-i$ frames with one I frame and $(n-i-1)$, P frames. However, for video broadcast, surveillance, or other one-directional connections over error-prone environments, delay is less critical. In these

Table 4.1: Bit rate and the PSNR value from MPEG GOP structure and this work for three sequences: Mother & daughter QCIF, Foreman QCIF and Stefan QCIF

Sequence	Average Bit Rate (Kbps)		Average PSNR (dB)	
	MPEG	This Work	MPEG	This Work
Mother & daughter	36.7	35.6	34.1	34.2
Foreman	67.5	66.6	33.4	33.5
Stefan	175.3	169.8	30.9	31.0

cases, the proposed scheme is much preferred. For the delay-sensitive applications, like video telephony, we can still find a tradeoff between the delay and robustness to make the caused delay acceptable according to different system requirements. The only parameter we need to change is i , alternatively, the position of I frame in a GOP. Then the optimization problem becomes

$$\min_{0 \leq i \leq n} L_{average}, \text{ subject to } t_{delay} \leq T \quad (4.4)$$

where T is the system delay requirement and $t_{delay} = t_I + (n - i - 1)t_P + t_{other}$ is the caused delay with t_I and t_P coding time of I and P frames respectively and t_{other} delayed time caused by other factors in the system. By solving the above minimization problem, we can both satisfy the delay requirement and improve robustness of the video transmission over error-prone channels. The different error resilience results will be shown according to varying system requirements in our experimental results.

3) System complexity: For the conventional coder without usage of multiframe video coding, the buffer needs to store one previous frame for the reference of the current coding frame at both the encoder and the decoder. For the proposed GOP structure, we enlarge the buffer to store an extra I frame at the encoder side while for the decoder, the maximal number of the frames we need to store for orderly displaying is $n - i$ since

before we display P'_{n-i} , we have already decoded $IP'_1P'_2...P'_{n-i-1}$. Therefore, the extra buffer to store the sequence in the backward predicting direction is added for the right displaying.

4.3 Efficient Reverse-play Functionality of the Proposed GOP Structure

Reverse playback is the most common VCR function in many digital video players. However, the predictive processing techniques employed in MPEG severely complicated the reverse-play operation. The proposed approach is quite efficient to provide the reverse-play operation over networks with fewer requirements on the network bandwidth. For VCR, we assume such a sequence like $...P_{-1}I_0P_1P_2P_3P_4P_5P_6P_7P_8I_9P_{10}...$. Table 4.2 illustrates the frames, which the server needs to send to the decoder according to different starting and ending points of the reverse play. We compare our structure with the conventional approach, which decodes the GOP up to the current frame, and then goes back to decode from the beginning of the GOP again up to the next frame to be displayed. From the table, we can see that the proposed approach outperforms the conventional method for the underlined data, where it saves a lot of retransmissions compared with the conventional GOP structure. For instance, by the conventional GOP structure, if the starting point of the reverse play is I_9 , the server needs to send totally one I frame (I_0) and eight P frames (P_1 to P_8) to display P_8 while for the proposed GOP structure in this chapter, the server only needs to send one P frame (P_8) to display it. For the other data in Table 4.2, the two structures perform the same.

Table 4.2: Number of frames to be transmitted over the network

			Starting Point of the Reverse Play								
			P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	I_9
Ending Point of the Reverse Play	Conventional	I_0	I	I	I	I	I	I	I	I	I
		P_1		I+P	I+P	I+P	I+P	I+P	I+P	I+P	I+P
		P_2			I+2P	I+2P	I+2P	I+2P	I+2P	I+2P	I+2P
		P_3				I+3P	I+3P	I+3P	I+3P	I+3P	I+3P
		P_4					I+4P	I+4P	I+4P	I+4P	I+4P
		P_5						I+5P	I+5P	I+5P	I+5P
		P_6							I+6P	I+6P	I+6P
		P_7								I+7P	I+7P
		P_8									I+8P
	Proposed	I_0	I	I	I	I	I	I	I	I	I
		P_1		I+P	I+P	I+P	I+P	I+P	I+P	I+P	I+P
		P_2			I+2P	I+2P	I+2P	I+2P	I+2P	I+2P	I+2P
		P_3				I+3P	I+3P	I+3P	I+3P	I+3P	I+3P
		P_4					I+4P	I+4P	I+4P	I+4P	I+4P
		P_5						<u>P</u>	<u>2P</u>	<u>3P</u>	<u>4P</u>
		P_6						<u>P</u>	<u>2P</u>	<u>3P</u>	
		P_7							<u>P</u>	<u>2P</u>	
		P_8								<u>P</u>	

Experimental results also show that as compared to the conventional system, the new GOP structure can reduce the required bandwidth significantly. Fig. 4.6 depicts the bandwidth requirement of the reverse play operation for three sequences Mother & daughter, Foreman and Stefan with GOP length $N_{GOP} = 11$. The starting points of the reverse play are frame $kN_{GOP} + 1$, ($k = 1, 2, 3, \dots$), which are the I frames in the sequence. The figure shows that the proposed structure outperforms the conventional one for all sequences. Note that, as shown in Fig. 4.6, the required bits for displaying the previous 5 frames behind each I frame are the same by the proposed structure and the conventional one, but for the other frames, the MPEG straightforward implementation of the reverse play operation requires much higher network bandwidth compared with the proposed GOP structure. Table 4.3 compares number of bits to be sent over the network between

(a)

(b)

(c)

Fig. 4.6. Bandwidth requirement with respect to the reverse play operation for sending (a) Mother & Daughter, (b) Foreman and (c) Stefan.

the proposed and the conventional structures. The saving in the bits to be sent over the network is more than 40% for all the sequences. It is obvious that the proposed structure can achieve significant performance improvement in terms of the network traffic load.

Table 4.3: Detailed comparisons of bits to be sent over the network between the proposed and the conventional structures

Sequence	Number of bits to be sent over the network		Saving of bits to be sent over the network
	Conventional	Proposed	
Mother & daughter	27549	14416	47.6%
Foreman	45037	24632	46.3%
Stefan	111440	59794	45.3%

4.4 Experimental Results

Extensive simulations have been carried to test the robustness of the proposed structure. For comparison, we also show the results obtained with the conventional structure. In the experiments, we tested different video sequences, which provide a good representation of the content transmitted over the networks. All sequences were coded under the same conditions. Every I frame was followed by 10 P frames. The frame rate was 10 frames/s. The sequences were compressed using the same QP 20. Experiments were performed to transmit the video sequences over a channel that was uncorrelated in time. This is reasonable since burst errors can be randomized by interleaving. Due to the random nature of such a channel, 50 different runs of the experiments were conducted. In the rest of this work, we will use Mother & Daughter QCIF, Foreman QCIF and Stefan QCIF to show our experiment results.

Firstly, to clearly show the effect of the proposed GOP structure, in our experiments we specify the frames where slice loss happens beforehand. In Fig. 4.7, there are slice losses at frame 7, 19, 33, 46, 59 and 72. We select these frames so as to show the generality of the distribution of slice losses in all frames. Actually these frames are the 6th, 7th, 10th, 1st, 3rd and 5th P frame after each I frame of the sequence. Their positions are

Fig. 4.7. Slice losses at frame 7, 19, 33, 46, 59 and 72.

typical in each GOP. For instance, for slice loss in frame 46, which is the 1st P frame after I frame, the error propagates to all the following P frames by the conventional MPEG GOP structure. But by the proposed structure, the error propagation stops at the 6th P frame, say frame 51, because all the following P frames are encoded in a backward manner with the next I frame as the starting reference. Other error propagations in Fig. 4.7 can be analyzed in the same way. Table 4.4 summarizes the number of affected frames according to different distribution of slice losses. Summarily we state that the affected frames are much fewer by the proposed GOP structure than by the conventional GOP structure.

Table 4.4: Affected frames according to different distribution of slice losses

Frame Where Loss Happens		Frame 7	Frame 19	Frame 33	Frame 46	Frame 59	Frame 72
Position of the frame		IPPPPPPP PPPI	IPPPPPPP PPPI	IPPPPPPP PPPI	IPPPPPPP PPPI	IPPPPPPP PPPI	IPPPPPPP PPPI
MPEG GOP Structure	Affected Frames	7, 8, 9, 10, 11	19, 20, 21, 22	33	46, 47, 48, 49, 50, 51, 52, 53, 54, 55	59, 60, 61, 62, 63, 64, 65, 66	72, 73, 74, 75, 76, 77
	Number of Affected Frames	5	4	1	10	8	6
	Total Number	34					
Proposed GOP Structure	Affected Frames	7	18, 19	29, 30, 31, 32, 33	46, 47, 48, 49, 50	59, 60, 61	72
	Number of Affected Frames	1	2	5	5	3	1
	Total Number	17					

The end-to-end performance in terms of average PSNR is illustrated in Fig. 4.8. The results using our GOP structure with different delay requirements are compared with the performance of the conventional structure. Delay of 5 frames, 3 frames and 1 frame means three types of GOP structure we applied, which are shown in Fig. 4.9. In contrast, the proposed structure exhibits excellent performance and always performs better than the conventional structure over a wide range of packet loss rate (from 2% to 20%). The performance of the structure with delay of 5 frames is the best while the performance of the conventional MPEG structure performs worst. The robustness will increase with the number of delayed frames if only the number does not exceed 5, which is the half of the total number of the P frames between the nearest two I frames. Thus we can adjust our GOP structure according to different system requirements to get a tradeoff between the robustness and the delayed time.

Finally, the proposed structure is compared with a compliant structure that starts with an I-frame with compliant prediction directions, which is shown in the Fig. 4.10. As we

Fig. 4.8 Average PSNR comparison according to different delay requirements. (a) Mother & Daughter, (b) Foreman and (c) Stefan.

can see from Fig. 4.10 (b), the compliant structure puts more B frames in the beginning of the GOP to improve the robustness. Therefore, the distances between P (or I) frames are not equal. Note that this may somehow lead to a low compression efficiency due to

Fig. 4.9. GOP structures applied in the experiments with (a) delay of 5 frames, (b) delay of 3 frames and (c) delay of 1 frame.

long predicting distance between P (or I) frames. As a result, it is not good to make the distances between the P (or I) frames different too much from each other. From the experiments, we found that the structure in Fig. 4.10 (b) is a reasonable choice, which is used for the comparison. Note that the number below each frame in Fig. 4.10 represents the length of the error propagation if a lost slice belongs to this frame. We can simply calculate that the average length of the error propagation of the proposed structure, which is $34p$, is shorter than that of the structure in Fig. 4.10 (b), which is $42p$, where p is the average slice lost rate. This is verified by the experimental results for Foreman and Stefan in Fig. 4.11, where we can see that the proposed structure outplays the compliant GOP structure over a wide range of packet loss rate.

Fig. 4.10. GOP structures with B frames of (a) the proposed structure and (b) a compliant structure that starts with an I-frame with compliant prediction directions.

4.5 Conclusion

In this chapter, we proposed a novel GOP structure so that we can optimally insert I frame in a GOP to improve the robustness of the video transmission over error-prone channels. The simulation results indicated that our proposed GOP structure outperforms the conventional MPEG GOP structure in error-prone environments. On the other hand, it inevitably increases the delay and the complexity of the system. A tradeoff has been found in the chapter.

Another important nature of the proposed structure is the efficient reverse-play operation for the video streaming system. It is obvious that the proposed structure can achieve significant performance improvement in terms of the network traffic load.

(a)

(b)

Fig. 4.11. Average PSNR comparison between the proposed structure and the compliant structure for (a) Foreman and (b) Stefan.

It is worthy to note that the error-resilient GOP structure is not compliant with existing MPEG-2/MPEG-4 profiles. It could however straightforwardly be implemented in H.264/MPEG-4 AVC which has all the tools for flexible reference frame handling.

Chapter 5

Optimal MB Interleaving Scheme

Using Maximum Minimal Distance

5.1 Introduction

As we discussed in previous chapters, the video applications have to provide sufficient built-in robustness to ensure that the quality of the decoded video is not overly affected

by the channel unreliability. A traditional error control method uses FEC codes. However, adding redundant information compromises the compression rate.

1	2	1	2	1	2	1	2	1	2	1
2	1	2	1	2	1	2	1	2	1	2
3	4	3	4	3	4	3	4	3	4	3
4	3	4	3	4	3	4	3	4	3	4
5	6	5	6	5	6	5	6	5	6	5
6	5	6	5	6	5	6	5	6	5	6
7	8	7	8	7	8	7	8	7	8	7
8	7	8	7	8	7	8	7	8	7	8
9	9	9	9	9	9	9	9	9	9	9

(a)

8	9	7	8	9	7	8	9	7	8	9
5	6	4	5	6	4	5	6	4	5	6
2	3	1	2	3	1	2	3	1	2	3
9	7	8	9	7	8	9	7	8	9	7
6	4	5	6	4	5	6	4	5	6	4
3	1	2	3	1	2	3	1	2	3	1
7	8	9	7	8	9	7	8	9	7	8
4	5	6	4	5	6	4	5	6	4	5
1	2	3	1	2	3	1	2	3	1	2

(b)

Fig. 5.1. Partition of a QCIF frame into 9 slices when using (a) checker-board type of mapping and (b) this work. The number signifies the number of the slice that the MB belongs to.

To achieve robustness without much loss in coding efficiency, flexible MB ordering (FMO), a new ability to partition the picture into slice groups, has been developed [63], with each slice becoming an independently-decodable subset of a slice group. When used

effectively, FMO can significantly enhance robustness to data losses by managing the spatial relationship between the regions that are coded in each slice. A typical mapping, named checker-board type of mapping is illustrated in Fig. 5.1 (a), which has been demonstrated useful for concealment in video conferencing applications. Assume that, during transmission, the packet containing the information of slice 3 gets lost. Since every lost MB has several spatial neighbors that belong to other slices, an efficient concealment can be applied. Actually, FMO is the developed form of the block interleaving in [53]. In [80], an interleaving scheme called block shuffling for image transmission over Rayleigh fading channels is developed. The shuffling pattern is generated by using the algorithm described as follows: 1) Generate a sequence of pseudorandom numbers corresponding to all PBs (picture blocks); 2) Scan the PBs by the ascending order of these pseudorandom numbers. However, this scheme needs a large array to store the random numbers. The larger the picture is, the larger the array will be.

In this chapter, we propose a novel interleaving structure for robust video transmission. We optimally group MBs into slices considering the maximal minimal distance of these MBs. The proposed structure will also be demonstrated to improve the robustness of video transmission with little overhead, which is highly desirable for bandwidth-limited networks.

The rest of this chapter is organized as follows. Section 5.2 gives a description of the proposed scheme. Meanwhile, its robustness, coding efficiency and other relative properties are analyzed. Section 5.3 gives the simulation results and Section 5.4 concludes the chapter.

5.2 Optimal Interleaving Scheme

In order to formulate the interleaving problem as an optimization problem, we have to select an objective function. Ideally, for a given number of slices, we wish to come up with an interleaving scheme, which will minimize the expected distortion under all possible transmission scenarios. As well known, mean square error (MSE) is a very popular criterion for video quality evaluation. Complex theoretical model considering the effect of error propagation in terms of MSE have been proposed in [16]. However, such model involves complicated parameter estimation, which varies with complexities of both video content and video codec. Therefore, we will choose a much simpler criterion.

The basic idea of the proposed interleaving scheme is to group the MBs in a frame according to a certain predefined structure. This structure should be designed such that neighboring MBs are separated as far as possible. We do this because:

- 1) Successful concealment of missing a MB relies on the spatial autocorrelation of the MB and the neighboring ones. Whatever the choice of the concealment method is, it normally selects spatially adjacent MBs in the interpolation process;

- 2) The more information is available, the more effectively the concealment works. Generally, the existing concealment methods only use spatially adjacent MBs.

Therefore, in our following interleaving scheme we will try to separate the neighboring MBs as far as possible. Thus, because every lost MB will have several spatial neighbors that belong to other slices, an efficient concealment can be applied.

5.2.1 Optimal Interleaving Scheme

Now we want to partition a frame into P slices: $C = (S_1, S_2, \dots, S_P)$. The minimal distance between two MBs of the slice i is defined to be

$$d_{\min}^{(i)} = \min_{MB_x, MB_y \in S_i; x \neq y} d(MB_x, MB_y) \quad (5.1)$$

where $d(MB_x, MB_y)$ is the distance between MB_x and MB_y . $d_{\min}(C)$, which is the function of the partition C , is defined to be the minimum of all $d_{\min}^{(i)}$,

$$d_{\min}(C) = \min_{i=1,2,\dots,P} d_{\min}^{(i)} \quad (5.2)$$

Now we want to separate the neighboring MBs as far as possible, i.e., to maximize

$$d_{\min}(C), \quad \max_{|C|=P} d_{\min}(C) \quad (5.3)$$

The above issue can be considered as a maximum minimal distance partitioning problem of the two-dimensional lattice [81]. In this section we show an interleaving pattern to maximize $d_{\min}(C)$ for a QCIF picture.

Fig. 5.1 (b) shows how the picture can be partitioned into $P=9$ slices. The label assigned to the MB shows the index of the slice to which the MB belongs, e.g., if the MB is labeled “1”, it belongs to S_1 . The pattern of MBs denoted with 1 is repeated throughout the picture. The labeling can be described by a 2×2 periodicity matrix $V = [v_1 \mid v_2]$ [5], whose columns are the basis vectors v_1 and v_2 shown in Fig. 5.2 (b). Here in this

example, we have $V = \begin{bmatrix} 3 & 1 \\ 0 & 3 \end{bmatrix}$ (can be looked up in the table of the periodicity matrix and

1	2	1	2	1	2	1	2	1	2	1
2	1	2	1	2	1	2	1	2	1	2
3	4	3	4	3	4	3	4	3	4	3
4	3	4	3	4	3	4	3	4	3	4
5	6	5	6	5	6	5	6	5	6	5
6	5	6	5	6	5	6	5	6	5	6
7	8	7	8	7	8	7	8	7	8	7
8	7	8	7	8	7	8	7	8	7	8
9	9	9	9	9	9	9	9	9	9	9

(a)

8	9	7	8	9	7	8	9	7	8	9
5	6	4	5	6	4	5	6	4	5	6
2	3	1	2	3	1	2	3	1	2	3
9	7	8	9	7	8	9	7	8	9	7
6	4	5	6	4	5	6	4	5	6	4
3	1	2	3	1	2	3	1	2	3	1
7	8	9	7	8	9	7	8	9	7	8
4	5	6	4	5	6	4	5	6	4	5
1	2	3	1	2	3	1	2	3	1	2

(b)

Fig. 5.2. The loss patterns of (a) checker-board and (b) this work. Packet loss rate=10%.

associated values of $d_{\min}$ in [81]) since $v_1 = [3, 0]^T$ and $v_2 = [1, 3]^T$. With this matrix we can get $d_{\min} = 3$. For the other value of P , the periodicity matrix and the value of $d_{\min}$ will be different and the table of the periodicity matrix and associated values of $d_{\min}$ can be found in [81].

Note that this interleaving scheme is very effective for method of adaptive slice size, in which the slice size can be changed according to the different channel condition. Large slice sizes appear best at low loss rate while smaller slices perform better at higher loss rate. Therefore, we can adjust the value of P to get the corresponding maximal value of $d_{\min}$ to get the best interleaving pattern. Besides, it is easy for the implementation system to achieve the proposed interleaving scheme by memorizing the table of the periodicity matrix by the codec. As far as transmission is concerned, the interleaving pattern, i.e., the periodicity matrix, can be preset in both transmitter and receiver. It is not necessary to change the interleaving pattern during video transmission even if the video format changes. Therefore, there is no extra transmission overhead required for the proposed scheme.

With the above interleaving scheme, erroneous MBs will be most largely isolated if channel errors occur. This is important for effectively employing error concealment techniques, which usually recover the corrupted MBs by taking advantage of spatial correlation among neighboring MBs.

A well-known concealment algorithm, weighted pixel value averaging [82], is applied in this chapter. In this method, 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 MBs 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.

Table 5.1: The coding efficiency of the proposed interleaving scheme. QP=20 in the experiments.

Sequence	Bits/Pic without interleaving scheme	Bits/Pic with the optimal interleaving scheme	Overhead ratio
Stefan QCIF	36512	37066	1.5%
Foreman QCIF	19672	20154	2.5%
Coastguard QCIF	17888	18738	4.8%

5.2.2 Properties of the Proposed Interleaving Scheme

In the next section, we will demonstrate that the proposed interleaving algorithm is very effective in isolating corrupted MBs for robust video transmission over error-prone channels. In this subsection, we consider other important issue regarding this interleaving algorithm.

1) Coding efficiency. In the experiment of this chapter, the proposed interleaving method was only used on I frames. For P frames, it is not used in this chapter since the coding of the motion vectors will explore the redundancy of neighboring MBs. However, since I frame is the basis of all the subsequent P frames, any uncorrectable image errors will be delivered to the subsequent P frames. Moreover lost unit in P frame may be recovered well by both temporal and spatial error concealment, while in I frame, only spatial error concealment can be employed. As a result, the error immunity of I frame is very important to be enhanced to achieve good visual quality throughout the sequence.

In H.263, for I frame, the proposed interleaving scheme will not affect the compression ratio of the AC coefficients since the AC coefficients are encoded without any prediction. The DC coefficients, however, undergoes DPCM in video standards.

When the proposed scheme is performed, the relative locations of the neighboring MBs, correspondingly the DC coefficients, will change. Thus, the redundancy between consecutive DC coefficients will not exist, resulting in lower compression ratio of the DC coefficients. Some video sequences are examined to test the coding efficiency of the proposed interleaving scheme, which is shown in Table 5.1. Clearly, the proposed algorithm slightly decreases the compression ratio. Note that for H.264 AVC, directional intra prediction can achieve better compression. As intra prediction is allowed to be performed only within the same slice, the impact of the different patterns on compression performance will be nontrivial.

5.3 Simulation Results

In this section we will analyze typical scenarios in the video transmission over error-prone networks. The video is coded using H.263. We begin by considering packet loss rate 10%. The loss patterns of checker-board and this work are shown in Fig. 5.2. The corresponding error concealed images for Foreman and Coastguard are shown in Fig. 5.3 and Fig. 5.4, respectively. It is worth noting that foreground objects generally occupy a group of connected blocks. The optimal interleaving scheme has an advantage of lowering the probability that the connected important MBs are lost at the same time. It optimally disperses the loss through the whole picture to make the visual effect least objectionable to human visual system (HVS). As we can see from Fig. 5.3 (b), the result with the checker-board method, a lot of MBs at the same position, say, the face of “Foreman”, are erroneous, which makes the subjective quality of the picture quite bad. With the proposed scheme, only one MB in the face will be in error whichever slice is

Fig. 5.3 Error concealed images for Foreman QCIF when packet loss rate=10%. (a) No loss, PSNR= 37.5, (b) checker-board, PSNR=29.4, (c) this work, PSNR=29.8.

Fig. 5.4 Error concealed images for Coastguard QCIF when packet loss rate=10%. (a) No loss, PSNR= 32.7, (b) checker-board, PSNR=28.6, (c) this work, PSNR=29.3.

lost (see Fig. 5.3 (c)). Compared with the result in Fig. 5.3 (b), the result with the proposed method is improved a lot in terms of visual quality although the objective quality is similar (PSNR=29.4, 29.8, respectively for Fig. 5.3 (b) and (c)). The advantage of the proposed method can be further verified in Fig. 5.4, which shows the error concealed images for Coastguard. In Fig. 5.4 (b), we can see that too many erroneous MBs in the same place make the annoying effect to be highly noticeable. However, by dispersing the errors, the subjective quality is better with the proposed scheme (Fig. 5.4 (c)).

Next, we consider packet loss rate 20%. Foreman and Coastguard are again used to evaluate the performance. The loss patterns of checker-board and this work are shown in

Fig. 5.5. Note that here the cases with burst losses for the both approaches are presented that consecutive slices (or consecutive MBs) are lost at the same time. The corresponding error concealed images for Foreman and Coastguard are shown in Fig. 5.6 and Fig. 5.7, respectively. Again, it is confirmed that the proposed scheme is effective in isolating the corrupted MBs, thereby improving the error concealment performance and obtaining good visual quality. It is observed from this example that, with checker-board method, the channel errors corrupt two rows of consecutive MBs. Thus, as shown in Fig. 5.6 (b) and Fig. 5.7 (b), even using the very powerful concealment, say, weighted pixel value averaging [6], the corrupted MBs can not be adequately reconstructed. This is because part of the missing information cannot be recovered since there is no reliable information available for interpolation. On the other hand, with the proposed interleaving method, almost all the corrupted MBs have 3 correctly received neighboring MBs to help concealment, as shown in Fig. 5.5 (b). The reconstructed images in Fig. 5.6 (c) and Fig. 5.7 (c) with the proposed method clearly present much better visual quality than Fig. 5.6 (b) and Fig. 5.7 (b).

Experiments have shown that, in video applications with QCIF-sized pictures, and at loss rates up to 20%, the visual impact of the losses can be kept quite low with the proposed interleaving method.

5.4 Conclusion

In this chapter, we proposed an optimal interleaving scheme so that we can optimally group MBs in slices to improve the robustness of the video transmission over error-prone channels. Considering the maximal minimal distance of MBs in a picture, the interleaving

algorithm can effectively isolate corrupted MBs caused by channel errors, thereby enabling error concealment to achieve better performance. The simulation results indicated that our proposed scheme outperforms the checker-board method in error-prone environments. Rather, it exhibits low redundancy, low complexity, and very high tolerance over error-prone channels.

1	2	1	2	1	2	1	2	1	2	1
2	1	2	1	2	1	2	1	2	1	2
3	4	3	4	3	4	3	4	3	4	3
4	3	4	3	4	3	4	3	4	3	4
5	6	5	6	5	6	5	6	5	6	5
6	5	6	5	6	5	6	5	6	5	6
7	8	7	8	7	8	7	8	7	8	7
8	7	8	7	8	7	8	7	8	7	8
9	9	9	9	9	9	9	9	9	9	9

(a)

8	9	7	8	9	7	8	9	7	8	9
5	6	4	5	6	4	5	6	4	5	6
2	3	1	2	3	1	2	3	1	2	3
9	7	8	9	7	8	9	7	8	9	7
6	4	5	6	4	5	6	4	5	6	4
3	1	2	3	1	2	3	1	2	3	1
7	8	9	7	8	9	7	8	9	7	8
4	5	6	4	5	6	4	5	6	4	5
1	2	3	1	2	3	1	2	3	1	2

(b)

Fig. 5.5. The loss patterns of (a) checker-board and (b) this work. Packet loss rate=20%.

Fig. 5.6. Error concealed images for Foreman QCIF when packet loss rate=20%. (a) No loss, PSNR= 37.5, (b) checker-board, PSNR=23.6, (c) this work, PSNR=25.3.

Fig. 5.7. Error concealed images for Coastguard QCIF when packet loss rate=20%. (a) No loss, PSNR= 32.7, (b) checker-board, PSNR=23.9, (c) this work, PSNR=26.5.

Chapter 6

Resynchronization Marker

Insertion for Layered Video

6.1 Introduction

As we discussed in Chapter 3, resynchronization has been proved to be a very effective tool for error resilience. Besides the conventional approaches in H.263 and MPEG-4, there are still many literatures considering the insertion of resynchronization markers [83] [13] [67]. For instance, in [83], Jeong, Kang and Kim propose an optimal resynchronization marker positioning technique using a novel information measure, i.e., the difference between the image recovered by error concealment and the image without

error. Methods in [83] [13] [67] share the same nature that they are all designed at *picture level*, i.e., discussing positioning of markers in the bitstream of one picture. On the other hand, Luis and Han [79] develop resynchronization approaches to *GOP level*, assigning different slice sizes to the video frames based on their type and order of appearance within a GOP. However, all the above methods [83] [13] [67] [79] are discussed only for single-layer video. In [84], Yan, Wu, Li and Tao design a hierarchical enhancement layer bitstream structure with resynchronization markers and Header Extension Code (HEC) for FGS video. In this scheme, the resynchronization markers are inserted in a fixed mode (same number of resynchronization markers in each bitplane) without considering about the significance of different bitplanes and different channel condition.

In this chapter, we introduce an optimal resynchronization method for layered video, which can optimally insert resynchronization markers within the bitplanes of the enhancement-layer based on the significance of the bitplane and the time-varying channel condition to perform a graceful degradation of video streaming when transmitting over wireless channels. Note that the proposed framework can be applied to all the layered video and image coding methods, e.g., SNR scalability coding method, although this chapter focuses on FGS video coding scheme.

The rest of this chapter is organized as follows. In Section 6.2, we propose a resynchronization approach based on the different importance of each bitplane and channel condition. Section 6.3 gives the simulation results and Section 6.4 concludes the chapter.

Fig. 6.1. FGS structure at the encoder and decoder.

6.2 Resynchronization Approach for FGS

Video

In this section, we first introduce FGS and analyze the factors that affect the video quality, based upon which, the optimization problem will be formulated thereafter. Finally, a fast and effective algorithm will be described to solve this seemingly prohibitive problem.

6.2.1 FGS and its Characteristics

A scalable video codec can produce bit-streams with different rates and reconstructed quality. This paper focuses on one scalable coding method, FGS, the structure of which is shown in Fig. 6.1. FGS consists of a base-layer and an enhancement-layer. Horizontal dependencies in the base-layer are due to predictive coding (I-, P- and B-pictures) and vertical dependencies in the enhancement-layer correspond to successive coding of bit-

planes (BP) in a single frame. Motion compensated scheme is used in the base-layer, which is coded at bit-rate R_{BL} , which is almost certainly lower than the lowest bandwidth $R_{\min}$ of the time-varying networks. Using the bit-rate $(R_{\max} - R_{BL})$, where $R_{\max}$ is the highest bandwidth of the networks, the enhancement-layer is coded progressively, i.e., the more significant bit-planes are transmitted prior to the less significant bit-planes. Under this framework, the video can be encoded over the available bandwidth $[R_{\min}, R_{\max}]$ according to the time-varying condition of networks. Because the enhancement-layer is intra-coded, the coding efficiency from temporal redundancy exploitation of FGS is lower than that of the single layer coder.

FGS efficiently provides servers bit-rate adaptability to the network condition, such as the bandwidth variation and packet losses. The multiple bit-planes in the enhancement-layer are in the order of importance from the most significant bit-plane to the least significant bit-plane. The important characteristic of the FGS coding scheme makes the video data more resilient to IP packet-losses [85].

6.2.2 Factors Affecting Video Quality

The bitplanes in the enhancement-layer of FGS video are not equally important. This can be illustrated in Fig. 6.2, which shows the different effects on the quality of the video due to loss of different bitplane. Note that even if the lower layers are lost, the high layers may still be decoded and increase the quality of the reconstructed picture. For bitplanes 2, 3 and 4, apparently, the lower the number of lost bitplane is, the larger the effect on the video quality will be. Note that the reason the loss of the first bitplane has little effect on the video quality is that the amount of bits amid the first bitplane is dramatically small in

Fig. 6.2. The effect on the quality of the video due to loss of different bitplane.

contrast to the higher bitplanes. This will also be demonstrated in the following. Thus for each bitplane, the amount of bits included and the importance to the constructed video quality is different from others. Summarily, there are three factors to be considered when we want to insert resynchronization markers in each bitplane:

- 1) The channel condition. With more resynchronization markers, the bitstream will be more robust to channel errors while the coding efficiency will drop. It is supposed that there must exist a tradeoff between the bits used for source coding and those used for resynchronization markers according to certain channel condition.

2) The amount of bits in the bitplane. Generally, with the level of bitplane getting higher, the amount of bits in this bitplane soars. If we insert same number of resynchronization markers in all the bitplanes, the slices in the higher bitplanes will be more prone to be lost due to the large slice size when error happens.

3) The significance of the bitplane. For FGS video, the bitplanes in the enhancement-layer are not equally important. This is not only due to the different amount of bits in each bitplane, but also due to the position of the bitplane in the enhancement layer. Intuitively, if we only consider the latter factor, we should put more resynchronization markers in the lower bitplanes. Another important thing is how to evaluate the significance of each bitplane. In this chapter, the criterion to evaluate the caused distortion is defined as *decreased PSNR* due to the loss of the slices in each bitplane without consideration of the dependencies of distortion among bitplanes.

6.2.3 Optimization Problem

Below, we discuss the problem of optimally inserting resynchronization markers subject to an overall target bit rate R_{Budget} . In the following formulation, we do not involve channel coding, but we should mention that the video quality can be further enhanced using unequal loss protection [86]. We firstly assume that there are totally L bitplanes to be transmitted. Given an overall coding rate, $R_{Overall}$, we want to optimally insert resynchronization markers such that the total distortion is minimized, that is,

$$\text{Min } D(\vec{N}), \text{ subject to } R_{overall} \leq R_{Budget} \quad (6.1)$$

with $\vec{N} = (N_1, N_2, \dots, N_l, \dots, N_L)$ and N_l is the number of slices (also resynchronization markers) in bitplane l .

$R_{Overall}$ is defined as

$$R_{Overall} = R_s + R_{RM} \quad (6.2)$$

with R_s being the source bit rate and R_{RM} being the amount of bits consumed by resynchronization markers. The objective therefore of the optimization is to find $\vec{N}^*$, where

$$\vec{N}^* = \arg \min D(\vec{N}) \quad (6.3)$$

so that the overall rate does not exceed R_{Budget} while minimizing the overall expected distortion, which can be expressed as

$$D(\vec{N}) = \sum_{l=1}^L \sum_{s_l=1}^{N_l} \Delta PSNR_l(s_l) P_l(s_l) + D_{RM}(\vec{N}) \quad (6.4)$$

There are two parts of distortion in the right side of the above equation. The first part of distortion is due to the loss of source bits because of channel errors and the second part depends on the number of bits consumed by inserted resynchronization markers.

In the first part, $\Delta PSNR_l(s_l)$ is defined as *decreased PSNR* due to loss of s_l slices in bitplane l and $P_l(s_l)$ is the probability that s_l slices are lost in bitplane l . $\Delta PSNR_l(s_l)$ can be obtained at the encoder by calculating the difference between the image with loss and the original image. We calculate $P_l(s_l)$ in the following way:

$$P_l(s_l) = \binom{N_l}{s_l} SER_l^{s_l} (1 - SER_l)^{1-s_l} \quad (6.5)$$

where SER_l is the slice error probability, i.e., the probability that one slice in bitplane l will be lost. For a channel modeled with i.i.d. bit errors, if we use BER as bit error rate, then we can calculate SER_l as

$$SER_l = 1 - (1 - BER)^{B_l} \quad (6.6)$$

with B_l being the average number of bits in each slice in bitplane l . Note that in this chapter, we assume the existence of a channel estimator that indicates the bit error rate.

The second part is the distortion caused by the consumption of bits by inserting resynchronization markers in all the L bitplanes. Note that for synchronization markers that we add to a stream, the same amount of source data needs to be removed to cater for the bandwidth requirement $R_{overall} \leq R_{Budget}$. We will not transmit this portion of source data in the highest bitplane L since it is the least important. Thus, we can express the second part of distortion as

$$D_{RM}(\vec{N}) = \Delta PSNR_L \left(r \sum_{l=1}^L N_l / B_L \right) \quad (6.7)$$

with r being the number of bits for one resynchronization marker.

Now, by solving the equations (6.1), (6.2) and (6.4), we can optimally insert resynchronization markers in each bitplane according to the amount of bits in each bitplane, the importance of each bitplane and the time-varying channel condition. However, this seems like an extremely difficult task especially for a large value of L . For example, if each layer has M MBs, the computational complexity will be $O(M^L)$, which is highly prohibitive. To circumvent this problem, we develop a simple and effective local hill-climbing algorithm, which is also adopted to assign FEC codes for layered image in [87]. We will give a brief description of it in the following.

Table 6.1: Pseudo-code of resynchronization marker assignment algorithm. M is the number of MBs in each bitplane and T_l is the number of bits in bitplane l . The variables $\vec{N}_{opt}$, $\vec{N}_{last}$, and $\vec{N}_{temp}$ are vectors that store redundancy assignments.

```

 $\vec{N}_{opt} = (1, 1, 1, \dots, 1);$ 
while ( $\vec{N}_{opt} \neq \vec{N}_{last}$ ) {
     $\vec{N}_{last} = \vec{N}_{opt};$ 
    for ( $l = 1; l \leq L; l++$ ) {
        for ( $q = -Q; q \leq Q; q++$ ) {
             $\vec{N}_{temp} = \vec{N}_{last};$ 
             $\vec{N}_{temp,l} = \vec{N}_{temp,l} + q;$ 
            if ( $\vec{N}_{temp,l} < 0$  or  $\vec{N}_{temp,l} > M$ )
                continue to next  $q$ ;
            if ( $q > 0$ )
                for ( $i = l + 1; i \leq L; i++$ )
                     $\vec{N}_{temp,i} = \left\lceil T_i / \max(T_l / \vec{N}_{temp,l}, T_i / \vec{N}_{temp,i}) \right\rceil$ 
            else
                for ( $i = 1; i \leq l - 1; i++$ )
                     $\vec{N}_{temp,i} = \left\lceil T_i / \min(T_l / \vec{N}_{temp,l}, T_i / \vec{N}_{temp,i}) \right\rceil$ 
            calculate  $D(\vec{N}_{temp})$ ;
            if ( $D(\vec{N}_{temp}) < D(\vec{N}_{opt})$ ) {
                 $\vec{N}_{opt} = \vec{N}_{temp};$ 
                 $D(\vec{N}_{opt}) = D(\vec{N}_{temp});$ 
            }
        }
    }
}
 $\vec{N}^* = \vec{N}_{opt};$ 

```

6.2.4 Hill-climbing Algorithm

First, we initialize the number of resynchronization markers as $(1, 1, 1, \dots, 1)$. For each iteration, we examine $2QL$ possible assignments to find certain $\vec{N}$ that can reduce

Table 6.2: The average slice size in each bitplane for three-bitplane scalability for Foreman CIF and Coastguard CIF (in MBs).

Sequence	Biplane Number	Bit Error Rate						
		10^{-6}	$10^{-5.5}$	10^{-5}	$10^{-4.5}$	10^{-4}	$10^{-3.5}$	10^{-3}
Foreman (CIF)	BP1	396.00	396.00	396.00	362.34	285.45	151.80	88.77
	BP2	143.55	90.09	47.74	28.71	17.93	12.10	11.44
	BP3	34.43	20.35	11.77	11.00	11.00	11.00	11.00
Coastguard (CIF)	BP1	396.00	396.00	396.00	219.78	143.55	81.84	47.96
	BP2	71.28	45.32	24.42	12.87	11.88	11.33	11.22
	BP3	22.66	11.55	11.11	11.00	11.00	11.00	11.00

$D(\vec{N})$, where Q is the search distance (maximal number of resynchronization markers that can be added to or subtracted from each bitplane). We calculate $D(\vec{N})$ after adding or subtracting 1 to Q resynchronization markers to each bitplane while satisfying our constraint $B_l \leq B_{l+1}$. The searching process is repeated until the best $\vec{N}$ minimizing $D(\vec{N})$ is found. The algorithm pseudo-code is given in Table 6.1. Our algorithm finds a local minimum, which is close to the global minimum with tolerable computation [87].

6.3 Experimental Results

Extensive simulations have been carried to test the performance of the proposed scheme, which is named as local optimal method. For comparison, we also show the results obtained with the scheme of fixed synchronization insertion method proposed in [84] and another straightforward scheme, which maximizes the number of decodable bits with feedback from the channel. We denote these two schemes as conventional method and suboptimal method in the rest of the chapter.

Fig. 6.3. Two-bitplane scalability results for Foreman.

In the experiments, we tested different video sequences, which provide a good presentation of the content transmitted over the wireless channels. All sequences were coded by MPEG-4 codec. Every I frame was followed by 29 P frames. The frame rate is 10 frames/s. The base-layer was compressed using the same QP 20. Experiments were performed to transmit enhancement layer of the video sequences over a channel that was uncorrelated in time. This is reasonable since burst errors can be randomized by interleaving. Due to the random nature of such a channel, 50 different runs of the experiments were conducted. In the rest of this work, we will use Foreman (CIF) and

Fig. 6.4. Two-bitplane scalability results for Coastguard.

Coastguard (CIF) to show our two sets of experiment results: two- and three-bitplane scalability.

In the conventional method, one slice consists of three rows of MBs, i.e., 66 MBs, for the first enhancement layer of Foreman (CIF), and 44 MBs for the first enhancement layer of Coastguard (CIF). For other enhancement layers, one slice consists of one row of MBs, i.e., 22 MBs [84]. In our proposed scheme, to reduce computational complexity, we select a minimal unit, 11 MBs, which means that each slice can and only can consist of some units of MBs. In terms of computational efficiency, the algorithm, which is implemented with C language, typically takes about 0.05 s on an Intel Pentium 4 CPU

Fig. 6.5. Three-bitplane scalability results for Foreman.

2.40GHz. We show the slice size within each bitplane for three-bitplane scalability in Table 6.2. It indicates the average number of MBs in one slice for each bitplane for 100 frames. We can find that as bit error rate rises, the slice size increasingly gets smaller. The reason why the slice size of the first bitplane is much larger than others is that the amount of bits in this bitplane is far smaller than that of others. The end-to-end performance (average PSNR) for two- and three-bitplane scalability is illustrated in Fig. 6.3-Fig. 6.6, respectively. The results using our optimal algorithm are compared with the performance of the conventional solution and the suboptimal one. In contrast, the proposed solution exhibits excellent performance and always performs at or above the

Fig. 6.6. Three-bitplane scalability results for Coastguard.

performance of the other two solutions over a wide range of bit error rate. For instance, for two-bitplane scalability, PSNR value is improved around 1 dB while for three-bitplane case, up to 2 dB PSNR improvement is achieved.

6.4 Conclusion

In this chapter, we proposed a resynchronization scheme for layered video so that we can optimally insert resynchronization markers in each bitplane according to three factors: the amount of bits in each bitplane, the importance of each bitplane and the time-varying channel condition. The simulation results indicated that our proposed scheme encodes the

video sequences with higher quality as compared to that of the conventional approach over error-prone channels.

Chapter 7

GOP-based Channel Rate

Allocation Using Genetic Algorithm for Scalable Video Streaming

7.1 Introduction

With the development of broadband wireless networks, such as IEEE 802.11b wireless LAN, delivering video over wireless networks has recently gained increasing attention. Although broadband wireless networks can transmit video data at high bit rates, there are still major challenges existing, such as fluctuations in channel quality and high bit-error

rates compared with wired links [88]. Transmission errors, together with lossy source coding techniques, lead to the distortion of the video sequences at the decoder. Hence, proper allocation of system resources to minimize the distortion is required to transmit video efficiently. UEP, which is based on the priority encoding transmission (PET) [89], has been proven to be very promising to resolve this problem by taking advantage of the differential sensitivities of the output bit-streams of video encoder.

UEP can mainly be classified into 2 categories according to the consideration of different aspects of bit-stream sensitivities:

1) Different layers of scalable coding. Because the different layers are not equally important, an obvious way of protecting progressive bit-stream is to add more protection to the layers that impact the quality more. UEP for scalable coding has been addressed by many researchers [90]-[92] [45] [46] [87] [86] for different types of scalable coding. For example, Mohr, et al. [87] apply UEP on Set Partitioning in Hierarchical Trees (SPIHT) coder and find an algorithm for optimizing the amount of protection bits used to protect progressive data. van der Schaar and Radha [86] study the impact of applying UEP between base- and enhancement-layer of fine-granularity-scalability (FGS) coding, as well as the impact of applying fine-grained loss protection (FGLP) to the enhancement-layer.

2) Different types of the frames in a GOP. This type of UEP scheme considers the different importance of different types of frames, like in MPEG-2, I-, P- or B-frames. In [93], a UEP scheme is proposed by considering the bit error sensitivity of the three types of frames so that the dependency among I-, P- and B-frames is exploited. The disadvantage of this kind of UEP is the ignorance of the temporal dependency of P-

frames in a GOP since it treats all P-frames equally. On the contrary, [94]-[95] [78] all explore the sensitivity of succeeding frames (including I- and P-frames) in order to minimize the mean distortion over the transmitted sequence.

To the best of our knowledge, these two aspects have never been jointly considered for channel rate allocation in previous literatures. In this chapter, we introduce a new UEP method, which not only considers differential importance of different frames in a GOP, but also of different layers of each frame. It can properly allocate channel rates to the frames in a GOP based on their importance, and in each frame, unequal amounts of protection are allocated to the progressive bit-stream of scalable video. However, searching for the channel rates is not an easy approach, especially when the length of the GOP or the number of layers in each frame is large. Hence, we use a more efficient algorithm, GA [96], in order to achieve fast channel rate allocation. Note that hill-climbing method [93] and local search algorithms [97] are ever applied to solve the problem of searching channel rates according to importance of different parts in the bit-stream. However, there exist a lot of limitations in using these methods. For instance, the channel rates, e.g., $r_1, r_2, \dots, r_n$, that need to be decided, have to be non-decreasing or non-increasing, i.e., $r_1 \leq r_2 \leq \dots \leq r_n$ or $r_1 \geq r_2 \geq \dots \geq r_n$. Unfortunately, this is not always true, which will be demonstrated in this chapter. Therefore, we apply GA to resolve the optimization problem formulated in this chapter.

The rest of the chapter is organized as follows. In Section 7.2, we firstly present an overview of our UEP algorithm jointly considering the different importance of the frames in a GOP and the different importance of the layers in each frame. After that, an optimization problem is formulated, an analytical distortion model is built and GA is

Fig. 7.1. System overview. Different gray colors denote the degree of importance: The darker, the more important.

briefly introduced to solve the optimization problem. The simulation results are given in Section 7.3 and the conclusion is drawn in Section 7.4.

7.2 Proposed Channel Rate Allocation

Approach

7.2.1 Proposed Framework

The MPEG video international standard specifies the coded representation of the video data. In order to achieve the highly efficient compression and meet the conflicting requirements of random access, three picture types have been defined: I-, P-, and B-frame. Due to the application of predictive coding techniques, the effect of channel errors on the video can be extremely severe when compressed video data is transmitted over error-prone channels. In order to provide the video data with some measure of reliability while communicating over these channels, it is necessary to exert some form of error control. The FEC codes are designed to protect data against channel errors. Additionally, scalable video coding is well developed due to the capability of reconstructing lower

Fig. 7.2. A frame is partitioned into two slice groups. The shaded macroblocks belong to slice group 1 and the white macroblocks to slice group 2.

resolution or lower quality signals from partial bit-streams. This allows for simple solutions in adaptation to network and terminal capabilities.

From the temporal dependency existing in a GOP, which is horizontally shown in Fig. 7.1, we can see that the earlier an error occurs in a GOP, the more frames will be corrupted. Here we ignore B-frames for simplicity but without loss of universality. To minimize the impact of the transmission error, an appropriate choice of channel rates for the frames in a GOP is necessary when the global transmission rate remains constant for a GOP. That is, we should add more channel bits to the former frames of a GOP while adding fewer to the latter ones. Furthermore, in each frame, the lower layers are more important than the higher ones since the higher layers can only be decoded based on the lower layers (vertically demonstrated in Fig. 7.1). And therefore, we should intuitively add more protection to the lower layers than the higher ones in each frame. The problem remained is: what is the appropriate channel rate allocation for the frames in a GOP, and also for the layers in each frame, to provide the best video quality under the constraint of a constant transmission rate? To solve this problem, we first introduce our proposed framework.

Fig. 7.3. A GOP is partitioned into two sub-bitstreams. The shaded slices belong to sub-bitstream 1 and the white slices to sub-bitstream 2.

Out of consideration for data packetization before video transmission, a frame can be partitioned into one or several slices. Some partitioning ways can be found in [63]. In this chapter, considering two contradictory aspects of transmission system: the source coding efficiency and the error concealment performance, we use a simple but error-resilient partitioning method, which is shown in Fig. 7.2. Here two slice groups are generated and each slice group may be partitioned into one or more slices. Obviously, when losing one of the two slice groups, the lost MBs have some neighboring MBs, which can be used to conceal the lost information. If we partition all the frames in a GOP in the same way (see one simplified example in Fig. 7.3, where one frame with four rows of MBs is only partitioned into 2 slices, shaded and white), in the temporal direction we call all the slices in the same spatial position as a sub-bitstream. If the GOP size is G , we number the slices in this sub-bitstream as the 1st slice to the G th slice according to the sequential number of the frames that they belong to. The structure of packetization for one sub-bitstream is shown in Fig. 7.4.

Fig. 7.4. Structure of channel rate allocation for a sub-bitstream.

For each sub-bitstream, we packet the t th slice (including the scalable layers) with the length $B(t)$ in bytes into m_t segments. For the l th segment, the length is $K_{t,l}$ and the width is $w_{t,l}$. Here

$$w_{t,l} = \lfloor b_{t,l} / K_{t,l} \rfloor + \alpha \quad (7.1)$$

where $b_{t,l}$ denotes the number of bytes in the l th segment and $B(t) = \sum_{l=1}^{m_t} b_{t,l}$. If the

number of bytes of the l th segment is divisible by $K_{t,l}$, $\alpha = 0$, or else $\alpha = 1$, when we will fill the remaining space of the square $K_{t,l} \times w_{t,l}$ with the data from the next segment.

Then we add FEC codes (Reed–Solomon codes are used in the chapter) with the length $N_t - K_{t,l}$ to each column. It is important to note that we here use segment, which can represent a scalable layer or not. In this chapter we assume one segment is one layer in the scalable video coding. Finally, the L_t columns, where

$$L_t = \sum_{l=1}^{m_t} w_{t,l}, \quad (7.2)$$

are horizontally partitioned into N_t packets at last. We do all the same to the other slices in this sub-bitstream.

7.2.2 Problem Formulation

Below, we discuss the problem of allocating channel rates subject to an overall target bit rate for a sub-bitstream illustrated in Fig. 7.4. In this chapter, we mainly discuss the problem of allocating channel rates to coded bit-streams. Therefore we don't consider source distortion here since it is independent of the channel coding [16]. Given an overall bit rate, R_{Budget} , we want to allocate channel bits such that the channel distortion, $D_c(\vec{r})$, for this sub-bitstream is minimized, that is,

$$\min D_c(\vec{r}), \text{ subject to } \sum_{t=1}^G N_t \times L_t \leq R_{budget} \quad (7.3)$$

The objective therefore of the optimization is to find the channel rate

$$\vec{r} = [r_1, r_2, \dots, r_t, \dots, r_G] \quad (7.4)$$

where r_t is the channel rate for the t th slice and it can be denoted as

$$r_t = [r_{t,1}, r_{t,2}, \dots, r_{t,l}, \dots, r_{t,m_t}] \quad (7.5)$$

with $r_{t,l} = K_{t,l} / N_t$ being the individual channel rate for the l th segment of this slice.

Based on the degree of the importance of each segment as analyzed, we have the following constraints:

$$r_{1,1} \leq r_{2,1} \leq \dots \leq r_{G,1} \quad (7.6)$$

$$r_{t,1} \leq r_{t,2} \leq \dots \leq r_{t,m_t}, \quad t = 1, 2, \dots, G \quad (7.7)$$

The constraint (7.6) demonstrates the dependence of the 1st segments (or base-layers) of all the slices that the base-layers of the former slices are more important than that of the latter ones. The constraint (7.7) tells us that in each slice, the lower scalable layers make more sense for the reconstruction.

Summarily, the overall bit rate does not exceed R_{Budget} while minimizing the overall expected distortion for a sub-bitstream, which can be written as

$$D_c(\vec{r}) = \sum_{t=1}^G D_c(\vec{r}, t) \quad (7.8)$$

where $D_c(\vec{r}, t)$ is the channel distortion of the t th slice.

7.2.3 Analytical Model for Channel Distortion

In this chapter, the mean square error (MSE) is used as the criterion to measure the channel distortion. In each slice, we define distortion per segment as the differential improvement due to the inclusion of this segment in the reconstruction. Therefore, in the absence of the channel errors, only the distortion for the 1st segment (base-layer) would be positive and the distortion for all the other segments would be negative since inclusion of these segments improves the quality of the video. Therefore we express the distortion $D_c(\vec{r}, t)$ as

$$D(\vec{r}, t) = \Delta D_{t,1}(\vec{r}) + \sum_{l=2}^{m_t} \Delta D_{t,l}(r_{t,l}) \times P_{Dec}(r_{t,l}) \quad (7.9)$$

where $\Delta D_{t,1}(\vec{r})$ is the expected distortion of the base-layer, $\Delta D_{t,l}(r_{t,l})$ ($2 \leq l \leq m_t$) is the differential improvement due to the inclusion of the l th segment in the reconstruction and $P_{Dec}(r_{t,l})$ denotes the probability that the l th segment is decodable, which is

$$P_{Dec}(r_{t,l}) = \sum_{m=0}^{N_t - K_{t,l}} P(m, N_t) \quad (7.10)$$

with $P(m, N_t)$, $m = 0, 1, \dots, N_t$, being the probability that m packets are lost when transmitting N_t packets. In this chapter, we assume the existence of a channel estimator that indicates this probability. This estimator could be formulated as any distribution of expected packet loss rate and a two-state Markov model is applied as it approximates the packet loss behavior fairly well.

Next let's determine the distortion of the 1st layer (the 1st segment) for each slice due to the error propagation. It has been shown [76] that if an error happens in the 1st layer of the 1st slice, the expected variance of the propagated error signal of the 1st layer of the t th slice is given by

$$\delta_v^2(t) = \delta_{u0}^2 (1 - P_{Dec}(r_{1,1})) \frac{1}{1 + \gamma(t-1)} \quad (7.11)$$

where δ_{u0}^2 can be treated as a constant value that does not depend on other model parameters. $1 - P_{Dec}(r_{1,1})$ is the residual error rate of the 1st layer of the 1st slice and γ is defined as the leakage which describes the efficiency of explicit and/or implicit loop filtering to remove the introduced error. Details about δ_{u0}^2 and γ can be found in [76].

Similarly, if the error happens in the 1st layer of the 2nd slice, the expected variance of the propagated error signal of the 1st layer of the t th slice is given by

$$\delta_v^2(t) = \delta_{u0}^2 (1 - P_{Dec}(r_{2,1})) \frac{1}{1 + \gamma(t-2)} \quad (7.12)$$

with $1 - P_{Dec}(r_{2,1})$ the residual error rate of the 1st layer of the 2nd slice.

If the error happens in the 1st layer of the i th slice ($i \leq t$), the expected variance of the propagated error signal of the 1st layer of the t th slice is given by

$$\delta_v^2(t) = \delta_{u0}^2 (1 - P_{Dec}(r_{i,1})) \frac{1}{1 + \gamma(t-i)} \quad (7.13)$$

with $1 - P_{Dec}(r_{i,1})$ the residual error rate of the 1st layer of the i th slice.

Thus the total expected channel distortion of the 1st layer of the t th slice in this GOP is

$$\Delta D_{t,1}(\vec{r}) = \sum_{i=1}^t \delta_{u0}^2 (1 - P_{Dec}(r_{i,1})) \frac{1}{1 + \gamma(t-i)} \quad (7.14)$$

7.2.4 Model of Wireless Packet-Erasure Channel

This scheme models the wireless network channel as a packet-erasure channel at the IP level. The process which leads to packet losses over wireless packet-erasure channel is quite complex. In this chapter, we assume the existence of a channel estimator that indicating the probability that a particular number of packets are lost, given the total number of packets to be transmitted. This estimator could be formulated as any distribution of expected packet loss rate, such as uniform, binomial, exponential, Zipf, Poisson, etc. However, among such distributions, a Markov model approximates the wireless channel's packet loss behavior fairly well.

In Markov model, hidden channel states $S = \{0, 1, 2, \dots\} \equiv N$ form a Markov chain with the initial state X_0 and the transitional probabilities $P(X_i | X_{i-1})$. The probability of a packet erasure is represented by a conditional density $P(\text{erasure} | X_i)$. The number of states N and the values of the erasure probabilities $P(\text{erasure} | X_i)$ can be chosen in a

number of ways. In this chapter, we apply a simple and analytically tractable Markov model with two states, i.e., $N = 2$. The two states of the model are denoted as 0 when packets are received correctly and timely and 1 when packets are lost. The model is fully described by transitional probabilities $P(1|0)$ from state 0 to state 1 and $P(0|1)$ from state 1 to state 0. Since these parameters are not very intuitive [98], we use the average loss probability $P_1 = \frac{P(1|0)}{P(0|1) + P(1|0)}$ and the average burst length $L_1 = \frac{1}{P(0|1)}$, which is the average number of consecutively lost packets [98].

Now, let us calculate the block error density function $P(m, N)$. In the Markov model, the event of a loss frees the memory of the loss process and restarts it. The distribution of error-free gaps can describe such a model. Assuming that a gap of length v is the event that a packet is missing after $v - 1$ packets are received correctly, the gap density function $g(v) = P(0^{v-1}1|1)$ represents the probability of a gap length v . The gap distribution function $G(v) = P(0^{v-1}|1)$ gives the probability of a gap length, which is not smaller than v . $g(v)$ and $G(v)$ can be calculated using transitional probabilities $P(1|0)$ and $P(0|1)$.

$$g(v) = \begin{cases} 1 - P(0|1) & \text{for } v = 1, \\ P(0|1)(1 - P(1|0))^{v-2} P(1|0) & \text{for } v > 1, \end{cases} \quad (7.15)$$

$$G(v) = \begin{cases} 1 & \text{for } v = 1, \\ P(0|1)(1 - P(1|0))^{v-2} & \text{for } v > 1. \end{cases} \quad (7.16)$$

To facilitate calculation of $P(m, N)$, now we use $R(m, n)$ as the probability that $m - 1$ packets will be lost within the next $n - 1$ packets, which follow a lost packet.

$$R(m, n) = \begin{cases} G(n) & \text{for } m = 1, \\ \sum_{v=1}^{n-m+1} g(v) R(m-1, n-v) & \text{for } 2 \leq m \leq N. \end{cases} \quad (7.17)$$

At last, the probability $P(m, N)$ that m packets will be lost within N packets can be calculated as

$$P(m, N) = \begin{cases} \sum_{v=1}^{n-m+1} P_1 G(v) R(m, n-v+1) & \text{for } 1 \leq m \leq N, \\ 1 - \sum_{m=1}^n P(m, N) & \text{for } m = 0, \end{cases} \quad (7.18)$$

where P_1 is the average packet loss probability.

7.2.5 Application of GA for Fast Channel Rate Allocation

Searching for the globally optimal assignment for the optimizing problem of (7.3), (7.6) and (7.7) seems to be computationally prohibitive in a practical system, particularly in the case of large length of GOP or large number of scalable layers in each frame. Hence, we need more efficient algorithms than explicit enumeration. For some difficult optimization problems, we do not have efficient algorithm to find the optimal solution within a reasonable short time. In practice, a good solution obtained within a reasonable time is much more appreciated than the global optimal solution obtained taking excessive time and/or computational resources. In these situations, GAs or simulated annealing, which make use of randomness to escape from local optimal solution, can be applied. In this chapter, we use GAs to solve the formulated problem as they do not require the search space to be unimodal, i.e., no local optima points, but just global optimum.

Fig. 7.5. Flowchart of GA to solve the optimization problem.

GAs attempt to mimic the natural evolution of species in order to solve difficult optimization problems. In GAs, every individual string in the population represents an instance of a solution to the problem being solved. The solution is suitably coded so that evolutionary computation operators, such as crossover, mutation, reproduction, etc., can be applied. Fitness of an individual population member (i.e. a solution instance) reflects how good the solution is to the problem. The structure of the GA we apply in this chapter is shown in Fig. 7.5.

Step 1: Initialization.

In order to initialize the population we need values for: 1) the number of bits for a solution candidate $\vec{r} = [r_1, r_2, \dots, r_t, \dots, r_G]$ with $r_t = [r_{t,1}, r_{t,2}, \dots, r_{t,l}, \dots, r_{t,m_t}]$. If we use l bits for each $r_{t,l}$, the length of a solution candidate will be $l \times \sum_{t=1}^G m_t$. 2) S --the total number of solution candidates $\vec{r}^1, \vec{r}^2, \dots, \vec{r}^i, \dots, \vec{r}^S$. Random initialization is used in this chapter although there are several other alternatives. The selection of S influences the computation cost and the diversity of the population.

Step 2: Calculation of $D_c(\vec{r}^*)$.

This step is to evaluate the fitness of every candidate solution in the population, i.e., to calculate $D_c(\vec{r}^i)$, $i = 1, 2, \dots, S$.

Step 3: Reproduction.

Reproduction operator copies solution strings from the current generation into a mating pool taking into consideration the objective values $D_c(\vec{r}^i)$ of individual solution string $\vec{r}^i$, $i = 1, 2, \dots, S$. Strings with higher fitness values will likely be represented in higher number in the mating pool, thereby passing on their genetic information into future generations. Reproduction uses the roulette wheel selection method [17] although some alternatives can be considered. In this approach, each string is assigned with a portion of the wheel's perimeter in proportion to its fitness level. Then the wheel is spun S number of times to select S strings into the mating pool. In this chapter we use a random number generator to simulate the operation of the roulette wheel.

Step 4: Crossover.

Crossover is applied to the ordered mating pool generated by the above step. It is performed between 2 randomly selected parent strings in the mating pool and results in 2

offspring. This operation exchanges partial information in each string, and therefore the offspring have distinct parts of both parents. Here the population is S and it will repeat $S/2$ times to get S offspring. Note that probability p_c may be used to determine if there will be crossover between 2 parent strings. As we'll perform crossover $S/2$ times, we can generate a random vector with $S/2$ elements each between 0 and 1. If the generated random number is smaller than p_c , we perform the crossover operation. Otherwise, no crossover is performed. If no crossover is performed, the selected parents are passed on as offspring to be processed by the next step: mutation operator. After determining whether crossover will take place, One-point crossover (although there are other alternatives) is performed as follows:

An integer position k , $1 \leq k \leq L-1$, is randomly selected along the string, where L is the length of the string. In our case, $L = l \times \sum_{t=1}^G m_t$ and there are $l \times \sum_{t=1}^G m_t - 1$ potential one-point crossover points. Two new strings (offspring) are generated by breaking up the parent strings at position k and exchanging the information.

Step 5: Mutation.

The mutation will introduce a degree of diversity to the population, prevent a premature convergence, and help to sample unexplored regions of the search space. An appropriate mutation operation will randomly change the bit values in the generated offspring strings from 0 to 1 or from 1 to 0. The probability of mutation operation, p_m , is small. Usually it is gradually reduced with increase in the number of generations. In the early stages, it is necessary to sample the whole search space while in the later stages, it is only a fine searching with small mutation not to disturb the high quality solutions. Here

there are S strings in the population and each string is $L = l \times \sum_{t=1}^G m_t$ bits long. Thus there

are $l \times \sum_{t=1}^G m_t \times S$ bits in total in the generated offspring. The bit position for mutation is

selected randomly according to the mutation probability. We can generate $l \times \sum_{t=1}^G m_t \times S$

random numbers between 0 and 1. If a generated value is smaller or equal to the mutation probability, we change the bit value in the solution string at the position.

Step 6 will be carried in the same manner with Step 2.

In general, steps 1 and 2 are performed just once in the beginning and steps 3, 4, 5 and 6 are repeated until a termination condition is satisfied. Here we choose completion of a given number of generations as the termination condition.

So far we considered the problems with simple constraints of the form of upper and lower bounds. For the problem in this chapter, which has some constraints, the main difficulty is that the GA often yields infeasible offspring (i.e. solutions that violate the constraints). We use penalty function approach to convert the constraint problem of (7.3), (7.6) and (7.7) into an unconstraint problem with some penalties. To do so, we rewrite the constraint in (7.3) as

$$g(\vec{r}) = R_{budget} - \sum_{t=1}^G N_t \times L_t \geq 0 \quad (7.19)$$

Similarly we rewrite (7.6) and (7.7) as

$$k_t = r_{t,1} - r_{t-1,1} \geq 0, \quad t = 2, 3, \dots, G \quad (7.20)$$

$$h_{t,l} = r_{t,l} - r_{t,l-1} \geq 0, \quad l = 2, 3, \dots, m_t, \quad t = 1, 2, \dots, G \quad (7.21)$$

Therefore the penalty function can be defined as

$$P_{\text{penalty}}(\lambda_1, \lambda_2, \lambda_3, \bar{r}) = \lambda_1 [\min\{0, g(\bar{r})\}]^2 + \lambda_2 \sum_{t=2}^G [\min\{0, k_t\}]^2 + \lambda_3 \sum_{t=1}^G \sum_{l=2}^{m_t} [\min\{0, h_{t,l}\}]^2 \quad (7.22)$$

where the parameters λ_1 , λ_2 and λ_3 are to be chosen to reflect the degree of constraint violation. For example, if any of (7.19)-(7.21) is not satisfied, the penalty term will definitely get large, which is not what we want. In our experiment, we set λ_1 , λ_2 and λ_3 to small values initially and gradually increase them. Thus they allow the GA to search feasible and infeasible regions initially and in the end with large values, we force the GA to give valid solutions.

Now by solving the following problem

$$\min D_c(\bar{r}) + P_{\text{penalty}}(\lambda_1, \lambda_2, \lambda_3, \bar{r}) \quad (7.23)$$

we can get the result for the constraint problem of (7.3), (7.6) and (7.7).

7.3 Experimental Results

In this section, we show the simulation results of channel bits allocation in Subsection 7.3.1 and the performance of the proposed scheme in Subsection 7.3.2. In the experiments, the test sequences were coded by MPEG-2 at frequency 15Hz and the format is IPPP.... The SNR scalability coding [99] is used to achieve the scalable video streaming. Intra and inter frame error concealments introduced in [63] are applied to conceal the errors. 4 sub-bitstreams are generated for each GOP. In each sub-bitstream, we may set $N_1 = N_2 = \dots = N_G$ to make the packet size L_t , $t = 1, 2, \dots, G$, flexible or vice

Fig. 7.6. Selection of the number of generation when applying GA.

versa. To decrease the complexity of the channel coding, in this chapter, we restrict

$$N_1 = N_2 = \dots = N_G = 100.$$

7.3.1 Channel Rate Allocation Using GA

We first try to select appropriate values for the number of generations and the size of population used in the GA. We use Foreman QCIF with 12 frames in a GOP and 4 SNR scalable layers in each frame in the experiment. For the two-state Markov channel [98], the average packet loss rate is 10% and the average burst length is 9.57. In Fig. 7.6, we change the number of generations (i.e., process of production, crossover, mutation and calculation of distortion, more details can be found in [96]) from 1 to 200 and find that

the best fitness will not change much any longer after the number of generations equals 50, which is used in the rest of this section. Thus it will need only 50 iterations to get the final allocation result. In terms of computational efficiency, the algorithm, which is implemented with C language, typically takes about 0.15 s on an Intel Pentium 4 CPU 2.40GHz. We change the size of population from 26 to 200 and discover that the result converges after it equals 100 (see Fig. 7.7). Therefore, in our following experiment, we will use 100 as the size of the population. The probabilities of crossover and mutation, p_c and p_m , are set to 0.65 and 0.004, which come from experience and secure a good result.

Fig. 7.8 presents the channel rate allocation for Foreman QCIF when the average packet loss rate is equal to 10% or 20%, respectively. The GOP length is 12 and the number of SNR scalable layers (the base-layer inclusive) in each frame is 4. The source bit rate is 150kbps and the overall bit rate is 200kbps. It can be seen that generally more parity symbols are added to the important frames than the less important ones. For instance, when the packet loss rate is 20%, the channel rate of the base-layer of I-frame is as small as 0.2. The channel rate increases as the frame number increases until the channel rate for the base-layer of the last P-frame in the GOP reaches over 0.6. Additionally, in each frame, more protection is added to the lower scalable layers. We can discover from the figure with packet loss rate 20% that most channel rates for the base-layers of all the frames in the GOP are smaller than 0.6, but as the layer number increases, the channel rates rise as well. Finally the channel rates for the highest layers reach almost 1, when little protection is added. Note that for the 2nd and the 3rd layers of I-frame, the channel rate is higher than that of some layers of P-frames. That means the

Fig. 7.7. Selection of the size of population when applying GA.

number of source bits for these layers of I-frame is much larger than that of P-frames. If the channel rate for these two layers of I-frame is too small, the number of rest protection bits for the following P-frames will be extremely small and the P-frames will be highly vulnerable to the channel errors. Therefore the system would not lower the channel rate for these layers of I-frames.

On the other hand, when comparing the two allocation patterns with different packet loss rates, we discover that the distribution patterns of channel rates are quite similar. From the 1st frame to the last one in a GOP, also from the 1st layer to the 4th layer in each frame, the channel rate changes in the same trend, sometimes even equals although the

Fig. 7.8. Channel rate allocation for a GOP with 12 frames and 4 SNR scalable layers in each frame for Foreman QCIF.

loss rates are different. The difference is, as the loss rate increases, more protection bits will be transferred from the higher layers to the lower ones, especially to the base-layers. This demonstrates that the lower layers are more sensitive to the errors when the channel condition gets even worse.

7.3.2 Performance Comparison

Due to the random nature of the two-state Markov channel, 100 different runs of the experiments were conducted to transmit video sequences with different packet-loss pattern. In the experiments, the test sequences were coded by MPEG-2 at frequency 15Hz and the format is IPPP.... Intra and inter frame error concealments introduced in [63] are applied to conceal the errors. 4 sub-bitstreams are generated for each GOP. In each sub-bitstream, we restrict $N_1 = N_2 = \dots = N_G = 100$. Here we use Foreman QCIF, Coastguard QCIF and Stefan QCIF to show our experiment results. The source bit rate and the overall bit rate are 150kbps and 200kbps for Foreman and Coastguard, 300kbps and

Fig. 7.9. PSNR comparison of the proposed UEP scheme against other two schemes.

400kbps for Stefan. The performance (average PSNR) over a range of average packet loss rate (from 0 to 30%) is illustrated in Fig. 7.9. The results using our UEP algorithm are compared with the performance of the following schemes:

EEP: equal channel rates are allocated to the layers of each frame of a GOP without consideration of the importance of the segments of the bitstream. Instead of the relationship among the channel rates in (7.6) and (7.7), here we have

$$r_{1,1} = r_{2,1} = \dots = r_{G,1} \quad (7.24)$$

$$r_{t,1} = r_{t,2} = \dots = r_{t,m_t}, \quad t = 1, 2, \dots, G \quad (7.25)$$

UEP [87] [100]: unequal channel rates are allocated to the layers of a frame without consideration of the position of the frame in the GOP. It is important to note that the basic concept, except the detailed implementation, of this method is typically used in the previous literatures [90] [87] [100] [101]. The main different of this method from the proposed one is that channel rate allocation is found only in one dimension. That is, for the t th frame in a GOP, we want to minimize the expected distortion $D_c(t)$ of the frame, $\min D_c(t)$, subject to $R_{S+C} \leq R_B$ (7.26)

where R_{S+C} is the total bit rate and R_B is the budget bit rate for the frame. The calculation of $D_c(t)$ can be found in [87] and [100], respectively.

In contrast, the proposed solution exhibits excellent performance over a very wide range of average packet loss rate and performs above the performance of other solutions, especially in poor channel conditions. An improvement of over 3 dB in quality is achieved when comparing with the EEP method and an improvement up to 1.8 dB is achieved when comparing with UEP [87] [100].

Next we evaluate the effects of a mismatch in packet loss rate between the assumed value at the time of channel rate allocating and the actual value. To do so, we have channel rate allocation at different assumed loss rates and transmit the data over the two-state Markov channel with different values of packet loss rate. Result is presented in Fig. 7.10 for Coastguard QCIF. On the average, the best performance is achieved when the packet loss rate used at the encoder matches exactly the channel condition, which confirms the accuracy of the above results. Additionally, we discover that the impact of the mismatch is worse at higher loss rates. To compensate for the possible feedback

Fig. 7.10. Performance of the proposed method with mismatch.

delays of the channel, it is preferable to assume worse network conditions when having the channel rate allocation.

7.4 Conclusion

In this chapter, we propose a novel UEP scheme. By jointly considering the temporal dependency of the frames in a GOP and the quality dependency of the scalable layers in each frame, the proposed method can properly allocate channel rates between the frames and in each frame, between the scalable layers based on their importance. We apply SNR scalability in our experiment although other sorts of scalable video coding could also be

used. In addition, due to the complexity of the two-dimensional channel rate allocation, the conventional method can hardly be used to get the allocation pattern. Therefore we propose a GA scheme to quickly solve the constraint optimization problem.

Chapter 8

Conclusion and Recommendations

This chapter summarizes the main developments and results of the thesis in Section 8.1, and indicates possible directions for further research in Section 8.2 as well.

8.1 Conclusion

This thesis is focused on error control techniques for video over error-prone channels. A general review of the state-of-the-art techniques is given and several proposed error control schemes, including error resilient and error protection methods, are presented.

1) A literature review of the error control schemes is presented.

Various techniques have been described for performing error control in video communication. Depending on channel error characteristics and system configuration and requirements, some techniques are more effective than others. The error-resilience

methods are implemented in the encoder. They reduce the error effect by introducing additional redundancy to the encoded video. The error concealment methods are based on the correlation of neighboring frames and pixels. The corrupted or lost area of compressed video can be recovered by the area that is spatially or temporally neighboring. The error protection is implemented at the transport layer to provide error detection and error correction in video transmission.

It is important to note that some techniques may be either too complicated for cost-effective implementation or introduce unacceptable processing delay for real-time applications. Aside from the delay and complexity issues, one important criterion for comparing different schemes is the required redundancy in source and/or channel coders to achieve the same degree of robustness. A fair comparison is difficult to obtain, however, because these techniques are usually developed for very different transport environments.

2) A content-based resynchronization approach is presented to improve the subjective video quality.

Using MB-based segmentation results, we propose a framework for the positioning of resynchronization markers such that the image quality of foreground can be improved at the expense of sacrificing the unimportant background. In the framework, we put the resynchronization markers just before the foreground. Therefore we can have better protection on the areas of interest, i.e., foreground, resulting in better subjective video quality.

The technique is useful for bit errors occurring in the wireless networks while for the packet losses some other scheme has to be proposed. Also, more effective

foreground/background segmentation and resynchronization schemes for non-head-and-shoulder video need further exploration.

3) An error-resilient GOP structure is developed to combat the channel errors.

A novel GOP structure is proposed for robust transmission of MPEG video bitstream. By selecting the optimal position of the I frame in a GOP, robustness can be achieved without reducing any coding efficiency. Normally the middle position of a GOP will be optimal. However sometimes for the delay-sensitive applications, like video telephony, due to the delay, we have to find a tradeoff between the delay and robustness to make the caused delay acceptable according to different system requirements.

Another advantage of the proposed GOP structure is also analyzed: compared with the conventional GOP structure, it provides reverse-play operation for MPEG video streaming with much less requirement on the network bandwidth.

The method can bring significant improvement in video quality without incurring significant additional complexity either at the encoder or decoder (just some delay and extra memory), and it can be incorporated in current video coding standards (possibly requiring some syntax change).

4) An optimal MB interleaving scheme based on maximal minimal distance is presented.

We propose a novel interleaving scheme for robust transmission of video sequences. By optimally grouping MBs into slices, robustness can be achieved with negligible reduction in coding efficiency. Considering the maximal minimal distance of MBs in a picture, the interleaving algorithm can effectively isolate corrupted MBs caused by channel errors, thereby enabling error concealment to achieve better performance. With

the H.264 codec utilizing intra-prediction, the proposed interleaving scheme is likely to incur coding efficiency loss. Therefore, a better approach would have to explicitly consider the trade-off between the coding efficiency and concealment difficulty.

5) An optimal resynchronization approach for layered video is proposed.

We proposed a resynchronization scheme for layered video so that we can optimally insert resynchronization markers in each bitplane according to three factors: the amount of bits in each bitplane, the importance of each bitplane and the time-varying channel condition. The effectiveness of this scheme lies in reducing loss of information based on significance of each bitplane of FGS video streaming. The proposed method is useful for bit errors occurring in the wireless networks while for the packet losses some other scheme has to be proposed.

6) GOP-based channel rate allocation approach is presented for layered coding.

We address the problem of UEP for scalable video transmission over wireless packet-erasure channel. By jointly considering the temporal dependency of the frames in a GOP and the quality dependency of the scalable layers in each frame, the proposed method can properly allocate channel rates between the frames and in each frame, between the scalable layers based on their importance to provide a graceful degradation of video quality as packet loss rate varies. We use GA to quickly get the allocation pattern, which is hard to get with other conventional methods, like hill-climbing method.

The proposed method has some limitations. First it is useful only when the underlying network's packet size is small compared to the average size of a video frame. For relatively low bit rate, a compressed video frame has a size comparable to the packet size, so that unequal error protection within a frame does not make much sense. Therefore

considering UEP at GOP level would be more promising in this case. Secondly developing an even faster and simpler method for deriving the optimal channel code rate is another task.

8.2 Recommendations for Future Research Work

We believe that emerging wireless/mobile networks will have the following characteristics.

- 1) Large bandwidth and delay variation;
- 2) Two distinct error states: a good stable state and a burst error state.

From these characteristics, we can conclude that the essential error control in future should have the following features:

- 1) Extraordinary adaptability to burst error state;
- 2) Rate control in the two-state error-prone environment.

One important issue needs to be considered is how to efficiently detect the state transition from a good stable state to a burst error state. The other issue is how to adapt the delivery system to the state change over heterogeneous radio access networks (RANs). For error control method such as error resilience and error protection in the burst error state, the essential element is rate adaptation to introduce data redundancy.

Cross-layer design of heterogeneous wireless Internet video systems is a relatively new and active field of research, in which many issues need further examination. Optimally allocating resources in this heterogeneous setting presents many challenges and opportunities. To solve the cross-layer optimization problems for video transmission, several components such as: 1)

adaptive modulation and channel coding; 2) adaptive retransmission; and 3) adaptive source rate control need to be jointly optimized to achieve better performance. Moreover, efficient end-to-end QoS support for multicast video transmission systems is an area that still requires considerable work.

Researchers interested in MDC have a wealth of promising directions as well. Coding structures that are likely to receive increasing attention are those using efficient prediction structures that eliminate mismatch, those using new Class C predictors that allow flexible tradeoffs between mismatch and redundancy, and those that can efficiently generate more than two descriptions. To combine scalable coding with MDC to offer both rate scalability and error resilience is another fairly new research direction. This can be achieved either by creating MD coders in which each description is scalable or by designing layered coders in which each layer is MD with a different amount of redundancy [117].

In the future, there are also a number of important research issues about joint source and channel coding that deserve significant attention.

1) Currently, one of the most active fields of research is cross-layer design of heterogeneous wireless video systems, which we believe need further investigation. In one scheme of this field, a proxy is ever deployed at the edge between the wired and wireless network to improve the overall video quality [102]. Another scheme has focused on end-to-end error control methods to increase the robustness of the whole system by analyzing the unique characteristics of all the network components [103] [104].

2) Efficient joint source and channel coding for multiuser and multicast video transmission systems is another active research area. Important issues such as fairness as

well as distributed versus centralized source allocation is necessary to be studied. The study of source-distortion framework in some new coding paradigms such as distributed video system and sensor networks, is another worthwhile research direction. At the same time, the definition of cost and distortion has to be modified appropriately based on different applications [114].

Author's Publications

Journal Publications

- 1) T. Fang and L. P. Chau, "GOP-based Channel Rate Allocation Using Genetic Algorithm for Scalable Video Streaming over Error-prone Networks", *IEEE Transactions on Image Processing*, vol. 15, no. 6, pp. 1323-1330, Jun. 2006.
- 2) T. Fang and L. P. Chau, "Efficient Content-based Resynchronization Approach for Wireless Video", *IEEE Transactions on Multimedia*, vol. 7, no. 6, pp. 1021-1027, Dec. 2005.
- 3) T. Fang and L. P. Chau, "An Error Resilient GOP Structure for Robust Video Transmission", *IEEE Transactions on Multimedia*, vol. 7, no. 6, pp. 1131-1138, Dec. 2005.
- 4) T. Fang and L. P. Chau, "Content-based Resynchronization for Robust Video Transmission", *IEEE Transactions on Broadcasting*, vol. 50, no. 4, pp. 390-395, Dec. 2004.

Conference Publications

- 5) T. Fang and L. P. Chau, "A Novel Unequal Error Protection Approach for Error Resilient Video Transmission", *IEEE International Symposium on Circuits and Systems*, Kobe, Japan, May 2005.
- 6) T. Fang and L. P. Chau, "Optimal Resynchronization for Layered Video over Wireless Channel", *IEEE International Symposium on Circuits and Systems*, Kobe, Japan, May 2005.
- 7) T. Fang and L. P. Chau, "A Novel Interleaving Algorithm for Robust Video Transmission", *IEEE International Symposium on Circuits and Systems*, Kobe, Japan, May 2005.
- 8) T. Fang and L. P. Chau, "Robust Group-of-Picture Architecture for Video Transmission over Error-prone Channels", *IEEE International Symposium on Circuits and Systems*, Kobe, Japan, May 2005.
- 9) L. P. Chau and T. Fang, "An Efficient Resynchronization Technique for Perceptual Quality Enhancement for Robust Video Transmission", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Montreal, Canada, May 2004.
- 10) T. Fang and L. P. Chau, "A Novel Resynchronization Marker Positioning Approach for Robust Video Transmission", *IEEE International Symposium on Circuits and Systems*, Vancouver, Canada, May 2004.

11) T. Fang and L. P. Chau, "Fine Granular Scalability Video over Wireless Communication Networks", *Fourth International Conference on Information, Communications & Signal Processing and Fourth Pacific-Rim Conference on Multimedia*, Singapore, Dec. 2003.

Bibliography

- [1] T. J. Ferguson and J. H. Rabinowitz, "Self-synchronizing Huffman codes," *IEEE Trans. Inform. Theory*, vol. IT-30, no. 4, pp. 687-693, Jul. 1984.
- [2] W.-M. Lam and A. R. Reibman, "Self-synchronizing variable length codes for image transmission," in *Proc. IEEE ICASSP*, San Francisco, CA, USA, vol. 3, pp. 477-480, Mar. 1992.
- [3] "Video codec for audiovisual services at px64 kbits," International Telecommunications Union, Geneva, Switzerland, ITU-T Recommendation H.261, 1993.
- [4] "Video coding for low bitrate communication," International Telecommunications Union, Geneva, Switzerland, ITU-T Recommendation H.263, 1998.
- [5] D. W. Redmill and N. G. Kingsbury, "The EREC: an error resilient technique for coding variable-length blocks of data," *IEEE Trans. Image Processing*, vol. 5, pp. 565-574, Apr. 1996.
- [6] J. Wen and J. D. Villasenor, "A class of reversible variable length codes for robust image and video coding," *Proc. 1997 IEEE Int. Conf. Image Processing*, vol. 2, Oct. 1997, pp. 65-68.

-
- [7] Description of Error Resilient Core Experiments, ISO/IEC JTC1/SC29/WG11 N1383, Nov. 1996.
- [8] M. Budagavi and R. Talluri, “Wireless video communications,” *Mobile Communications Handbook*, Boca Raton: CRC Press LLC, 1999.
- [9] P. Haskell and D. Messerschmitt, “Resynchronization of motion compensated video affected by ATM cell loss,” in *Proc. IEEE ICASSP*, San Francisco, Ca, USA, vol. 3, pp. 545-548, Mar. 1992.
- [10] J. Liao and J. Villasenor, “Adaptive intra update for video coding over noisy channels,” in *Proc. IEEE ICIP*, Lausanne, Switzerland, vol. 3, pp. 763-766, Sept. 1996.
- [11] T. Wiegand, Personal Communication, July 1998.
- [12] S. Wenger and G. Côté, “Using RFC2429 and H.263+ at low to medium bit-rates for low-latency applications,” in *Proc. Packet Video Workshop (PVW)*, New York, NY, USA, Apr. 1999.
- [13] G. Côté, 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, Jun. 2000.
- [14] J. Y. Liao and J. Villasenor, “Adaptive intra block update for robust transmission of H.263,” *IEEE Trans. Circuit Syst. Video Technol.*, Vol. 10, pp. 30—35, Feb. 2000.
- [15] S. Dogan, A. Cellatoglu, M. Uyguroglu, A. H. Sadka and A. M. Kondo, “Error-resilient video transcoding for robust Internetwork communications using GPRS,” *IEEE Trans. Circuit Syst. Video Technol.*, Vol. 12, pp. 453—464, Jun. 2002.

-
- [16] Z. He, J. Cai, and C. W. Chen, “Joint source channel rate-distortion analysis for adaptive mode selection and rate control in wireless video coding”, *IEEE Trans. Circuit Syst. Video Technol.*, Vol. 12, pp. 511-523, Jun. 2002.
- [17] B. Girod and N. Farber, “Wireless video,” *Compressed Video Over Networks*, Nov. 1999.
- [18] S. Wenger, “Video redundancy coding in H.263+,” in *Proc. AVSPN*, Aberdeen, UK, Sep. 1997.
- [19] S. Wenger, G. Knorr, J. Ott, and F. Kossentini, “Error resilience support in H.263+”, *IEEE Trans. Circuit Syst. Video Technol.*, vol. 8, no. 6, pp. 867-877, Nov. 1998.
- [20] N. Franchi *et al.*, “Multiple video coding for scalable and robust transmission over IP,” presented at the *Packet Video Conf.*, Nantes, France, 2003.
- [21] C. Kim and S. Lee, “Multiple description coding of motion fields for robust video transmission,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 999–1010, Sep. 2001.
- [22] D. Chung and Y. Wang, “Multiple description image coding using signal decomposition and reconstruction based on lapped orthogonal transforms,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 895–908, Sep. 1999.
- [23] S. Cho and W. A. Pearlman, “A full-featured, error resilient, scalable wavelet video codec based on the set partitioning in hierarchical trees (SPIHT) algorithm,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 157–170, Mar. 2002.
- [24] I. V. Bajic and J. W. Woods, “Domain-based multiple description coding of images and video,” *IEEE Trans. Image Process.*, vol. 12, pp. 1211–1225, Oct. 2003.

-
- [25] V. A. Vaishampayan, "Design of multiple description scalar quantizer," *IEEE Trans. Inform. Theory*, vol. 39, pp. 821–834, May 1993.
- [26] V. A. Vaishampayan and J. Domaszewicz, "Design of entropy-constrained multiple description scalar quantizer," *IEEE Trans. Inform. Theory*, vol. 40, pp. 245–250, Jan. 1994.
- [27] J.-C. Batllo and V. Vaishampayan, "Asymptotic performance of multiple description transform codes," *IEEE Trans. Inform. Theory*, vol. 43, pp. 703–707, Mar. 1997.
- [28] M. Srinivasan and R. Chellappa *et al.*, "Multiple description subband coding," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, 1998, pp. 684–688.
- [29] S. D. Servetto *et al.*, "Multiple description lattice vector quantization," in *Proc. IEEE Data Compression Conf.*, 1998, pp. 13–22.
- [30] H. Jafarkhani and V. Tarokh, "Multiple description trellis coded quantization," *IEEE Trans. Commun.*, vol. 47, pp. 799–803, Jun. 1999.
- [31] M. Fleming and M. Effros, "Generalized multiple description vector quantization," in *Proc. IEEE Data Compression Conf.*, 1999, pp. 3–12.
- [32] C. Tian and S. Hemami, "Universal multiple description scalar quantization: Analysis and design," in *Proc. IEEE Data Compression Conf.*, 2003, pp. 183–192.
- [33] Y. Wang *et al.*, "Multiple description coding using pairwise correlating transforms," *IEEE Trans. Image Processing*, vol. 10, pp. 351–366, Oct. 2001.
- [34] V. K. Goyal and J. Kovacevic, "Generalized multiple description coding with correlating transforms," *IEEE Trans. Inform. Theory*, vol. 47, pp. 2199–2224, Sep. 2001.
- [35] S. H. Lee, P. J. Lee, and R. Ansari, "Cell loss detection and recovery in variable rate video," in *Proc. 3rd Int. Workshop Packet Video*, Morriston, Mar. 1990.

-
- [36] K. Challapali, X. Lebegue, J. S. Lim, W. H. Paik, R. Saint Girons, E. Petajan, V. Sathe, P. A. Snopko, and J. Zdepski, "The grand alliance system for US HDTV," *Proc. IEEE*, vol. 83, pp. 158–174, Feb. 1995.
- [37] T. Kinoshita, T. Nakahashi, and M. Maruyama, "Variable bit rate HDTV CODEC with ATM cell loss compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, pp. 230–237, Jun. 1993.
- [38] V. Parthasarathy, J. W. Modestino, and K. S. Vastola, "Design of a transport coding scheme for high quality video over ATM networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 2, pp. 358–376, Apr. 1997.
- [39] J.-Y. Cochenne, "Method for the correction of cell losses for low bit-rate signals transport with the AAL type 1," ITU-T SG15 Doc. AVC-538, July 1993.
- [40] E. Ayanoglu, R. Pancha, and A. R. Reibman, and S. Talwar, "Forward error control for MPEG-2 video transport in a wireless ATM LAN," *ACM/Baltzer Mobile Networks Applicat.*, vol. 1, no. 3, pp. 245–258, Dec. 1996.
- [41] G. Cheung and A. Zakhor, "Joint source/channel coding of scalable video over noisy channels," in *Proc. ICIP*, 1996, pp. 767–770.
- [42] K. Ramchandran, A. Ortega, M. Uz, and M. Vetterli, "Multiresolution broadcast for digital HDTV using joint source/channel coding," *IEEE J. Select. Areas Commun.*, vol. 11, pp. 6–23, Jan. 1993.
- [43] M. Srinivasan and R. Chellappa, "Joint source-channel subband coding of images," in *Proc. ICASSP*, 1997, pp. 2925–2928.

-
- [44] N. Tanabe and N. Farvardin, "Subband image coding using entropy-coded quantization over noisy channels," *IEEE J. Select. Areas Commun.*, vol. 10, pp. 926–943, June 1992.
- [45] M. Srinivasan and R. Chellappa, "Adaptive source-channel subband video coding for wireless channels," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 1830–1839, Dec. 1998.
- [46] L. P. Kondi, F. Ishtiaq and A. K. Katsaggelos, "Joint source-channel coding for motion-compensated DCT-based SNR scalable video," *IEEE Trans. Image Processing.*, vol. 11, pp.1043-1052, Sep. 2002.
- [47] J. W. Modestino, D. G. Daut, and A. L. Vickers, "Combined source-channel coding of images using the block cosine transform," *IEEE Trans. Commun.*, vol. COM-29, pp. 1261–1274, Sept. 1981.
- [48] V. A. Vaishampayan and N. Farvardin, "Optimal block cosine transform image coding for noisy channels," *IEEE Trans. Commun.*, vol. 38, pp. 327–336, Mar. 1990.
- [49] M. W. Marcellin, P. Sriram, and K.-L. Tong, "Transform coding of monochrome and color images using trellis coded quantization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, pp. 270–276, Aug. 1993.
- [50] R. E. Blahut. Theory and Practice of Error Control Codes. Addison Wesley, Reading, MA, 1983.
- [51] S. Aign, "Error concealment for MPEG-2 video," in *Signal Recovery Techniques for Image and Video Compression and Transmission*, Eds. Norwell, MA: Kluwer, 1998,
- [52] M. C. Hong, L. Kondi, H. Scwab, and A. K. Katsaggelos, "Video error concealment techniques," *Signal Processing: Image Communications*, vol. 14, pp. 437-492, 1999.

-
- [53] Q.-F. Zhu, Y. Wang, and L. Shaw, "Coding and cell loss recovery for DCT-based packet video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, pp. 248–258, Jun. 1993.
- [54] T. Stockhammer, H. Jenkac, and C. Weiss, "Feedback and error protection strategies for wireless progressive video transmission," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 6, pp. 465–482, Jun. 2002.
- [55] D. Turaga and T. Chen, "Model-based error concealment for wireless video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 6, pp. 483–495, Jun. 2002.
- [56] H. Sun and W. Kwok, "Concealment of damaged block transform coded images using projections onto convex sets," *IEEE Trans. Image Processing.*, vol. 4, pp. 470–477, Apr. 1995.
- [57] G.-S. Yu, M. M.-K. Liu, and M. W. Marcellin, "POCS-based error concealment for packet video using multiframe overlap information," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 6, pp. 422–434, Aug. 1998.
- [58] H. Sun, K. Challapali, and J. Zdepski, "Error concealment in digital simulcast AD-HDTV decoder," *IEEE Trans. Consumer Electronics*, vol. 38, pp. 108–117, Aug. 1992.
- [59] W.-M. Lam, A. R. Reibman, and B. Liu, "Recovery of lost or erroneously received motion vectors," in *Proc. ICASSP'93*, Minneapolis, MN, pp. V417–V420.
- [60] R. Talluri, "Error-resilient video coding in the ISO MPEG-4 standard," *IEEE Commun. Mag.*, vol. 1, no. 1, pp. 112–119, June 1998.
- [61] B. Yan and K. Ng, "A survey on the techniques for the transport of MPEG-4 video over wireless networks," *IEEE Trans. Consumer Electronics*, vol. 48, no. 4, pp. 863–873, Nov. 2002.

-
- [62] T. Stockhammer, M. M. Hannuksela, and T. Wiegand, "H.264/AVC in wireless environments," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 657–673, July 2003.
- [63] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, July 2003.
- [64] S. Wenger, M. Hannuksela, T. Stockhammer, M. Westerlund, and D. Signer, "RTP payload format for H.264 video," *Internet Eng. Task Force*, IETF draft-ietf-avt-rtp-h264-03.txt, Oct. 2003.
- [65] I. Moccagatta, S. Soudagar, J. Liang and H. Chen, "Error-resilient coding in JPEG-2000 and MPEG-4", *IEEE J. Select. Areas Commun*, vol. 18, pp. 899-914, Jun. 2000.
- [66] K. Yoo, "Adaptive resynchronization marker positioning method for error resilient video transmission," *IEE Electronics Letters*, vol. 34, pp. 2084-2085, Oct. 1998.
- [67] K. H. Yang, D. W. Kang and A. F. Faryar, "Efficient intra refreshment and Synchronization Algorithms for robust transmission of video over wireless networks," *IEEE ICIP*, Oct. 2001, vol. 1, pp. 938 –941.
- [68] M. van der Schaar and Y.-T Lin, "Content-based selective enhancement for streaming video," *IEEE ICIP*, vol. 2, pp. 977 –980, Oct. 2001.
- [69] D. Chai, and K. N. Ngan, "Face segmentation using skin-color map in videophone application," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 551-564, Jun. 1999.
- [70] S. Gao and G. Tu, "Robust H.263+ video transmission using partial backward decodable bitstream (PBDBS)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, pp. 182-187, Feb. 2003.

-
- [71] D. LeGall, "MPEG: A video compression standard for multimedia applications," *Commun, ACM*, vol. 34, no. 4, pp. 46-58, 1991.
- [72] C. H. Fu, Y. L. Chan, W. C. Siu, "Efficient reverse-play algorithm for MPEG video streaming with VCR functionality," *ICICS-PCM*, Dec. 2003.
- [73] M. S. Chen and D. D. Kandlur, "Downloading and stream conversion: supporting interactive playout of videos in a client station," in *Proc. 2nd Int. IEEE Conf. Multimedia Computing and Systems*, pp. 73-80, 1995.
- [74] S. J. Wee, "Reversing motion vector fields," in *Proc. IEEE International on Conference Image Processing 1998 (ICIP98)*, pp. 209-212, Oct. 1998.
- [75] S. J. Wee and B. Vasudev, "Compressed-domain reverse play of MPEG video streams," in *Proc. SPIE Conf. Multimedia Syst. and Appl.*, pp. 237-248, Nov. 1998.
- [76] K. Stuhlmüller, N. Färber, M. Link and B. Girod, "Analysis of video transmission over lossy channels," *IEEE J. Select. Areas Commun*, vol. 18, pp. 1012-1032, Jun. 2000.
- [77] G. Reyes, A. R. Reibman, S. Chang and J. C.-I. Chuang, "Error-resilient transcoding for video over wireless channels," *IEEE J. Select. Areas Commun*, vol. 18, pp. 1063-1074, Jun. 2000.
- [78] X. K. Yang, *et al.*, "Unqual loss protection for robust transmission of motion compensated video over the internet," *Signal Processing: Image Communication* 18, pp. 157-167, 2003.
- [79] L. O. -Barbosa and T. Han, "On the use of Frame-based slice size for the robust transmission of MPEG video over ATM networks," *IEEE Trans. Broadcasting*, vol. 46, pp. 134-143, Jun. 2000.

-
- [80] K. I. Chan, J. Lu and J. C.-I. Chuang, "Block shuffling and adaptive interleaving for still image transmission over rayleigh fading channels," *IEEE Trans. Veh. Technol.*, vol. 48, pp. 1002-1011, May. 1999.
- [81] I. V. Bajic and J. W. Woods, "Maximum minimal distance partitioning of the Z^2 lattice," *IEEE Trans. Inform. Theory*, vol. 49, pp. 981-992, Apr. 2003.
- [82] 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.
- [83] J. -H. Jeong, H. -S. Kang, J. -K. Kim, "Optimal resynchronization marker positioning method using a novel information measure," *Signal Processing: Image Communication* 17, pp. 799-806, 2002.
- [84] R. Yan, F. Wu, S. Li and R. Tao, "Error resilience methods for FGS video enhancement bitstream," *IEEE PCM*, Dec. 2000.
- [85] W. P. Li, "Overview of Fine Granularity Scalability in MPEG-4 Video Standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 301-317, Mar 2001.
- [86] M. van der Schaar and H. Radha, "Unequal packet loss resilience for fine-granular-scalability video," *IEEE Trans. Multimedia*, vol. 3, pp. 381-393, Dec. 2001.
- [87] E. Mohr, E. A. Riskin and R. E. Ladner, "Unequal loss protection: graceful degradation of image quality over packet erasure channels through forward error correction," *IEEE J. Select. Areas Comm.*, vol. 18, pp. 819-828, Jun. 2000.
- [88] Majumder, *et al*, " Multicast and unicast real-time video streaming over wireless LANs," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 524-534, Jun. 2002.

-
- [89] A. Albanese, J. Blömer, J. Edmonds, M. Luby, and M. Sudan, "Priority encoding transmission", *IEEE Trans. Inform. Theory*, vol. 42, pp. 1737-1744, Nov. 1996.
- [90] J. Kim, R. M. Mersereau and Y. Altunbasak, "A multiple-substream unequal error-protection and error-concealment algorithm for SPIHT-coded video bitstreams," *IEEE Trans. Image Processing*, vol. 13, pp. 1547-1553, Dec. 2004.
- [91] G. Cheung and A. Zakhor, "Bit allocation for joint source/channel coding of scalable video," *IEEE Trans. Image Processing*, vol. 9, pp. 340-356, Mar. 2000.
- [92] U. Horn, K. W. Stuhlmüller, M. Link, and B. Girod, "Robust internet video transmission based on scalable coding and unequal error protection," *Signal Process. Image Commun.*, vol. 15, pp. 77-94, Sep. 1999.
- [93] C. Huang and S. Liang, "Unequal error protection for MPEG-2 video transmission over wireless channels", *Signal Process. Image Commun.*, vol. 19, pp. 67-79, Jan. 2004.
- [94] F. Marx and J. Farah, "A novel approach to achieve unequal error protection for video transmission over 3G wireless networks", *Signal Process. Image Commun.*, vol. 19, pp. 313-323, Apr. 2004.
- [95] T. Fang and L.-P. Chau, "A novel unequal error protection approach for error resilient video transmission," *Proc. 2005 IEEE Int. Symp. Circuits and Systems*, May 2005.
- [96] D. E. Goldberg, "Genetic algorithms in search, optimization, and machine learning", Addison Wesley, Dec. 1988.
- [97] V. M. Stanković, R. Hamzaoui and Z. Xiong, "Real-time error protection of embedded codes for packet erasure and fading channels," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, pp. 1064-1072, Aug. 2004.

-
- [98] E. O. Elliott, "A model of the switched telephone network for data communications," *Bell. Syst. Techn. J.*, pp. 89-109, Jan. 1965.
- [99] B. G. Haskell, A. Puri, and A. N. Netravali, *Digital Video: An Introduction to MPEG-2*. London, U.K.: Chapman & Hall, 1997, Digital Multimedia Standards Series.
- [100] Puri and Ramchandran, "Multiple description coding using forward error correction codes," *Proc. of 33rd Asilomar Conference on Signals and Systems*, vol. 1, pp. 342-346, Oct. 1999.
- [101] L. Cheng, W. Zhang, and L. Chen, "Rate-distortion optimized unequal loss protection for FGS compressed video," *IEEE Trans. Broadcasting*, vol. 50, pp. 126-131, Jun. 2004.
- [102] G. Cheung, W. Tan, and T. Yoshimura, "Rate-distortion optimized application-level retransmission using streaming agent for video streaming over 3G wireless network," in *Proc. ICIP*, Rochester, NY, 2002, pp. I 529-I 532.
- [103] J. Chakareski and P. A. Chou, "Application layer error correction coding for rate-distortion optimized streaming to wireless clients," presented at the *ICASSP*, Orlando, FL, 2002.
- [104] F. Zhai, Y. Eisenberg, T. N. Pappas, R. Berry, and A. K. Katsaggelos, "Rate-distortion optimized product code forward error correction for video transmission over IP-based wireless networks," presented at the *ICASSP*, QC, Canada, May 2004.
- [105] K. Brayer and S. Natarajan, "An investigation of ARQ and hybrid FEC-ARQ on an experimental high latitude meteor burst channel", *IEEE Transactions on Communications*, Vol. 37, pp. 1239-1242, Nov. 1989.

-
- [106] A. Shiozaki, K. Okuno, K. Suzuki and T. Segawa, “A hybrid ARQ scheme with adaptive forward error correction for satellite communication”, *IEEE Transactions on Communications*, Vol. 39, pp. 482—484, Apr. 1991.
- [107] S. Kallel, “Efficient hybrid ARQ protocols with adaptive forward error correction”, *IEEE Transactions on Communications*, Vol. 42, pp. 281—289, Apr. 1994.
- [108] Coding of Audio-Visual Objects: Visual (MPEG-4 Video), Committee Draft ISO/IEC 14496-2, Oct. 1997.
- [109] S. Fukunaga, T. Nakai, and H. Inoue, “Error resilient video coding by dynamic replacing of reference pictures,” in *Proc. IEEE Global Telecommunications Conf. (GLOBECOM)*, London, U.K., Nov. 1996, vol. 3, pp. 1503–1508.
- [110] Y. Tomita, T. Kimura, and T. Ichikawa, “Error resilient modified inter-frame coding system for limited reference picture memories,” in *Proc. Int. Picture Coding Symp. (PCS)*, Berlin, Germany, Sept. 1997.
- [111] W. Tan and A. Zakhor, “Multicast transmission of scalable video using receiver-driven hierarchical FEC,” in *Proc. Packet Video*, New York, Apr. 1999.
- [112] D. Wu, Y. T. Hou, B. Li, W. Zhu, Y.-Q. Zhang, and H. J. Chao, “An end-to-end approach for optimal mode selection in Internet video communication: theory and application,” *IEEE J. Select. Areas Commun.*, vol. 18, pp. 977–995, June 2000.
- [113] M. Etoh, and T. yoshimura, “Advances in wireless video delivery,” *Proc. IEEE*, vol. 93, pp. 111–122, Jan. 2005.
- [114] A. K. Katsaggelos, *et al.*, “Advances in efficient resource allocation for packet-based real-time video transmission,” *Proc. IEEE*, vol. 93, pp. 135–147, Jan. 2005.

-
- [115] D. Wu, Y. T. Hou, and Y. Q. Zhang, "Transporting real-time video over the Internet: challenges and approaches," *Proc. IEEE*, vol. 88, pp. 1855–1875, Dec. 2000.
- [116] D. Wu, Y. T. Hou, and Y. Q. Zhang, "Scalable video coding and transport over broad-band wireless networks," *Proc. IEEE*, vol. 89, pp. 6–20, Jan. 2001.
- [117] Y. Wang, A. R. Reibman, and S. Lin, "Multiple description coding for video delivery," *Proc. IEEE*, vol. 93, pp. 57–70, Jan. 2005.
- [118] B. Girod, and N. Farber, "Feedback-based error control for mobile video transmission," *Proc. IEEE*, vol. 87, pp. 1707–1723, Oct. 1999.
- [119] J. D. Villasenor, Y. Q. Zhang, and J. Wen, "Robust video coding algorithms and systems," *Proc. IEEE*, vol. 87, pp. 1724–1733, Oct. 1999.
- [120] Y. Wang, and Q. Zhu, "Error control and concealment for video communication: a review," *Proc. IEEE*, vol. 86, pp. 974–997, Oct. 1999.
- [121] Y. Wang, *et al.*, "Error resilient video coding techniques," *IEEE Signal Processing Magazine*, pp. 61–82, Jul. 2000.
- [122] I-C, Lee, *et al.*, "On the throughput of multicasting with incremental forward error correction," *IEEE Trans. on Information Theory*, vol. 51, pp. 900–918, Mar. 2005.
- [123] J. Reichel, M. Wien and H. Schwarz, "Scalable video model 3.0," ISO/IEC JTC 1/SC 29/WG 11 N6716, Oct. 2004.
- [124] A. Jerbi, *et al.*, "Error-resilient region-of-interest video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, pp. 1175–1181, Sept. 2005.