LAYERED SCALABLE AND LOW COMPLEXITY VIDEO ENCODING: NEW APPROACHES AND THEORETIC ANALYSIS

A Thesis

Submitted to the Faculty

of

Purdue University

by

Yuxin Liu

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

August 2004

To my parents; and in memory of my grandparents.

ACKNOWLEDGMENTS

Toward finishing up this document, I have so many names in my mind that I would like to acknowledge. They filled my journey with so much fun and learning. Even though my words of thanks are short, my gratitude is long and deep.

First of all, I would like to thank my major advisor, Professor Edward J. Delp. I thank him for having my dream come true. I had so much respect for him before I came to Purdue, and it had been amazing for me to become a student of his and become a member of the VIPER lab. I am grateful for his guidance and support. Also, I have been so much enjoying talking with him, even though we do not always stick to strictly academic topics.

I would like to thank my co-advisor Professor Paul Salama. I am grateful for his many insightful and detailed suggestions. He is more like a friend to me than as a teacher.

I would also like to acknowledge my other committee members, Professor Ness B. Shroff and Professor Michael D. Zoltowski, for their support. I got an "Excellent" comment in my final exam paper for EE538 I took from Professor Zoltowski, while I got my only "B" from EE547 I took from Professor Shroff. Regardless of this, I have been so much enjoying their classes and grateful for their imparting their knowledge to me.

I am grateful to organizations which support research at the University level. In particular, my research work has been funded by an Indiana Twenty-First Century Research and Technology Fund grant and a PRF Research Grant from Purdue.

I would like to thank Dr. Josep Prades-Nebot, a professor at the Universidad Politécnica de Valencia, Valencia, Spain. I had been lucky to work with him at the end of 2003 while he was a visiting scholar in VIPER. He always has wonderful ideas. This dissertation would not have been possible without his collaboration.

I would like to thank Dr. Christine I. Podilchuk for her help and collaboration. I thank her for her offering me the chance working in the prestigious Bell Labs as a summer intern and meeting with the people I had long admired. I am grateful for having had numerous conversations with her on academic problems.

I would like to thank Dr. Gregory W. Cook for his great work on the theoretic analysis of scalable video coding. His work not only motivated my work on the theoretic analysis of leaky prediction, but, more importantly, opened a door for me to explore the area of rate distortion theory.

I would like to thank Professor Luis Torres, a professor at the Technical University of Catalonia, Barcelona, Spain, for his support and friendship. He has been visiting VIPER three times during the time I have been at Purdue. I thank him for his encouraging words to me: "Be a nice girl and finish your Ph.D. this year! ..."

I would like to thank my fellow officemates: Eugene T. Lin, Cuneyt M. Taskiran, Hyung Cook Kim, Jinwha Yang, and Jennifer L. Talavage, and my "pseudo" officemates: Rafael Villoria, Aravind K. Mikkilineni, Anthony F. Martone, and Michael

Igarta, for their great help and inspiration in my academic research and their sincere friendship.

I would like to give my special thanks to Eugene, for his assistance and consultation on my software programming and a lot of others we have been through together in the years I have been in VIPER; to Cuneyt, for his great humor and wonderful ideas that he has in many issues which have inspired me; to Hyung Cook, for his kind heart - I have been asking for help from him on software installation for so many times, and he never complained; to Mike and Aravind, for their extraordinary patience in helping me to debug programs executed under UNIX.

I would like to thank Limin Liu and Zhen Li, my friends and colleagues in VIPER, for their friendship and collaboration.

I would like to thank Yajie Sun, Sahng-Gyu Park, Hakeem Ogunleye, Lauren A. Christopher, and Oriol Guitart. I had been enjoying the time working with them in VIPER. It was sad to see they leave (it is so great to have Oriol back to Purdue, even though I will not be here), and I wish the best for their future.

I would like to thank Yan Huang and Bin Ni for their help in getting this document done in an appropriate format. Also, they have brought so much fun to my life.

I would like to thank Rongmei Zhang, my three-year's roommate at Purdue. It had been hard to stay away from my family especially in a different country. With her true friendship, I felt home at Purdue. I would like to give special thanks to her

for her encouragement she gave to me on that rainy day I was notified I failed my qualification exam.

I have dedicated this document to my mother and father. They have supported me throughout these many years while I have been pursuing my academic career. I thank them for giving me the chance to come to this world. I enjoy it.

Finally, I thank my husband, Hui Wayne Peng, for his support and love. He is my best friend and confidante. I have been so lucky to have him at my side whose support, understanding, encouragement, care, and true love make all the difference in the world.

TABLE OF CONTENTS

Page

Page

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

ABSTRACT

Liu, Yuxin. Ph.D., Purdue University, August, 2004. Layered Scalable and Low Complexity Video Encoding: New Approaches and Theoretic Analysis. Major Professors: Edward J. Delp and Paul Salama.

Transmission of digital video signals over current data networks demands efficient, reliable, and adaptable video coding techniques due to the heterogeneous nature of current wired and wireless networks. In this dissertation, we focus on the scalable video coding structure, in particular leaky prediction layered video coding (LPLC), and low complexity video encoding. Scalable video coding facilitates channel adaptive and error resilient performances. Low complexity video encoding shifts the computational complexity from the encoder to the decoder, which addresses applications with scarce resource at the encoder.

We highlight a deficiency inherent in LPLC, namely that the enhancement layer cannot always "enhance" the rate distortion performance. We develop a general framework that applies to both LPLC and a multiple description coding scheme using motion compensation, and use this framework to confirm the existence of the deficiency. We propose an enhanced LPLC based on maximum-likelihood estimation to address the deficiency in LPLC.

We further develop theoretic analysis of LPLC with respect to the leaky factor. We obtain two sets of rate distortion functions in closed form for LPLC, through the

use of rate distortion theory and the use of a quantization noise model. Theoretical results of both closed form expressions are evaluated, which conform with the operational results.

We describe a low complexity video encoding technique, which is developed for applications where resources are scarce at the video encoder whereas resources at the decoder are relatively abundant. We develop a low complexity video encoding approach that uses new B-frame direct coding modes. Experimental results have shown that our approach has a competitive rate distortion performance compared to the conventional high complexity video encoding approach.

We also discuss the reliable transmission of digital video over an error-prone environment. We present a thorough evaluation of a joint source-channel video coding technique over wireless networks. We obtain reliable video transmission using adaptive packetization and a two-layer rate-distortion optimization scheme.

1. INTRODUCTION

1.1 Overview

Demands for multimedia communications have been growing significantly over the past decade. Digital video lies at the core of most multimedia applications. Particularly, the compression and transmission of digital video over heterogeneous networks has become an area of active research.

A typical video communication system is composed of five components: source encoder and decoder, channel encoder and decoder, and the transmission channel, as shown in Fig. 1.1.

The transmission of video signals is constrained by resource availability at the transmitter, varying channel bandwidth, and the inevitable transmission errors. Applications such as video streaming also require timely transmission. Therefore, considering the heterogeneous nature of current wired and wireless networks and the various characteristics of receivers, digital video transmission demands efficient, reliable, and adaptable video coding techniques [1].

Video source coding, or video compression, aims to minimize the distortion at a given target data rate, or to minimize the data rate required for obtaining a target distortion. Video compression mainly tries to remove two kinds of redundancies

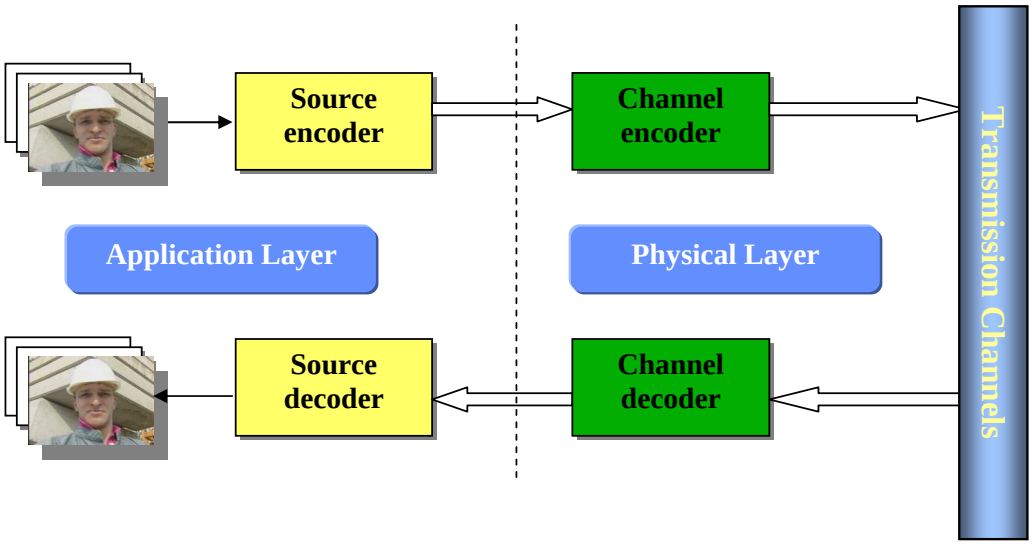


Fig. 1.1. A typical video communication system

that exist in a video sequence: spatial redundancy and temporal redundancy. In the literature, two-dimensional (2D) orthogonal transforms, such as the 2D discrete cosine transform (DCT) or the 2D discrete wavelet transform (DWT), are used to remove spatial redundancy in video sequences and obtain energy compaction. Motion estimation (ME) and motion compensation (MC) have been widely used to remove temporal redundancy in video signals. All current video coding standards have utilized the hybrid 2D orthogonal transform and motion compensation techniques.

For a given video sequence, each frame is usually partitioned into non-overlapping macroblocks of size 16×16 pixels. A macroblock is further divided into four 8×8 blocks, or even 8×4, 4×8, 4×4 blocks. Essentially, each macroblock can be coded using one of two modes: intra-coding mode or inter-coding mode.

The intra-coding of an arbitrary macroblock only uses the macroblock itself and the surrounding macroblocks in the same frame. Hence, the intra-coding mode is implemented in a same manner as the coding of a 2D still image. A typical implementation procedure of intra-coding is shown in Fig. 1.2. In contrast, the inter-coding of a macroblock uses motion estimation and compensation techniques. Basically, the motion estimation process finds the best match of the current macroblock, under a specific distortion metric, in the reference frame. The motion vectors, which denote the location of the best match area relative to the current macroblock position, are thus obtained. The motion compensation uses this best match as a prediction of the current macroblock. The prediction error frame (PEF), the difference between the original macroblock and its prediction, is then coded and transmitted together

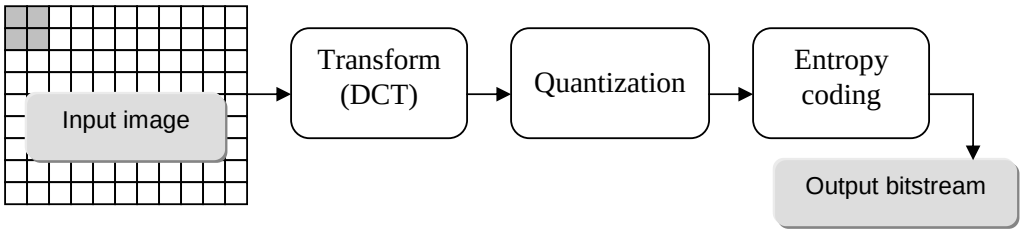
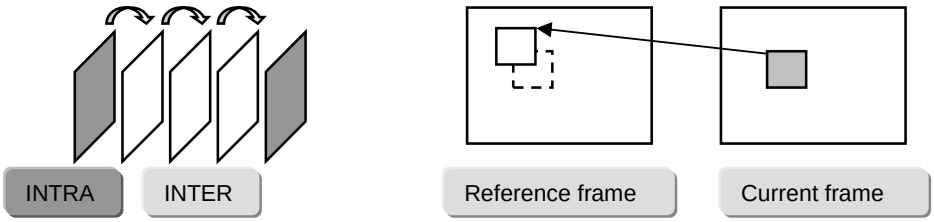


Fig. 1.2. Transform source coding



(a) INTRA/INTER mode (b) Motion estimation

Fig. 1.3. Motion estimation/motion compensation

with the coded motion vectors. Inter-coding using motion estimation/compensation is shown in Fig. 1.3.

Frames where all macroblocks are constrained to be intra-coded are referred to as I frames. Frames where macroblocks are inter-coded by using past reference frames are P frames, whereas a B frame is a frame where the motion compensated prediction is obtained from past and/or future reference frames. Video coding methods using three-dimensional (3D) orthogonal transforms, such as the 3D wavelet transform, have been developed where spatial and temporal redundancies are both removed by the use of the 3D transform [2].

In the following we will briefly address some specific topics in video coding which form a fundamental basis for our work.

1.1.1 Scalable Video Coding

Scalable video coding is designed to facilitate channel-adaptive and error resilient performances in heterogeneous error prone channels. Two types of scalabilities are exploited in state-of-the-art scalable video coding approaches, nested scalability and parallel scalability.

In nested scalability different levels of the bitstream are decoded in a fixed sequential order [3]. Nested scalability modes include rate scalability, SNR (signal-to-noise ratio) scalability, temporal scalability, spatial scalability, or content scalability.

An instantiation of nested scalable video coding is the layered scalable video coding structure, where a multilayered representation is generated for a video sequence [4, 5]. The lower layers (base layer) provide a coarse representation of the original video sequence, while the higher layers (enhancement layer) include refinement information for the video sequence. In this dissertation, we focus on the layered rate scalable video coding structure, and simply refer to it as layered video coding. Fine granularity scalability (FGS) is a specific layered video coding structure which possesses fully rate scalability over a wide range of data rates [4].

Layered coding is desired for error resilient video streaming over heterogeneous networks with changing bandwidth mainly because: (1) It can be adapted to varying channel bandwidth by simply discarding the higher layer(s), or, by truncating

the bitstream when coded using FGS; (2) It allows one to protect parts of the bitstream differently, i.e., to use unequal error protection (UEP). For error resilient video transmission in an error-prone environment, error protection can be used on the base layer since it carries more significant information. This achieves a trade-off between coding efficiency and robustness. For example, the base layer bitstream could be protected by Forward Error Correction (FEC) coding, or transmitted using an error-recovery capable network protocol such as TCP (Transmission Control Protocol) [6, 7]. The enhancement layer however still remains vulnerable to errors. Due to the potential incompleteness or destruction of the enhancement layer, traditional layered coding schemes usually do not incorporate the enhancement layer into the motion compensation loop at the encoder to prevent error drift at the decoder. This results in poor coding efficiency, when compared to non-scalable coding, since the high-quality reconstruction offered by both the enhancement layer and the base layer is not exploited by the motion compensation operation.

A fully rate scalable video codec, which is used in this dissertation, is SAMCoW (Scalable Adaptive Motion Compensated Wavelet) [5, 8–14]. It has two main features: (i) the use of adaptive motion estimation/compensation to reduce temporal redundancy [15, 16]; and (ii) a modified embedded zero tree (EZW) wavelet image compression scheme, known as Color Embedded Zero tree Wavelet (CEZW) to encode the intra frames and the PEFs [17, 18]. SAMCoW falls into the category of the conventional layered coding structure since only one motion compensation loop is implemented and any embedded bitstream beyond the base layer is excluded from

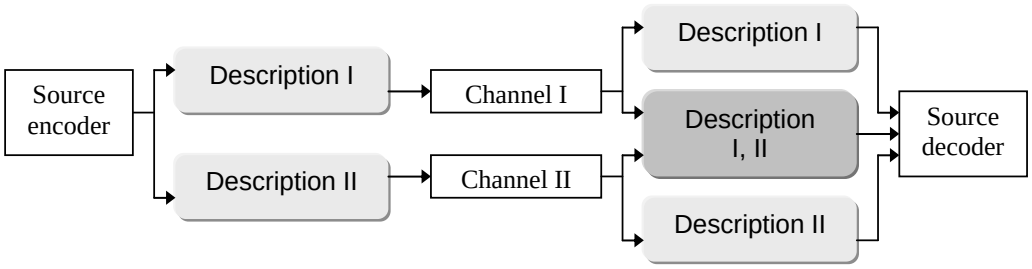


Fig. 1.4. Video coding and transmission by Multiple Description Coding (MDC)

the motion compensation loop. If R_B denotes the data rate allocated to the base layer, and R_T denotes the encoding data rate for the overall bitstream, SAMCoW allows drift-free decoding at any data rate between R_B and R_T if no error or truncation occurs to the bitstream.

In parallel scalability schemes, parallel and equally important but mutually refinable descriptions of a given video sequence are generated. Parallel scalability is inherent in Multiple Description Coding (MDC) approaches [19]. MDC is desirable for error resilient video streaming since each description can be independently used to reconstruct the original video signal regardless of the availability of any other description. In general, two descriptions are used, and the decoder reconstructs the encoded video from both descriptions if they are available, or from either description. The coding and transmission procedure of MDC with two descriptions is shown in Fig. 1.4.

MDC is a data partitioning scheme, since it partitions the bitstream into independent descriptions and thus prevents error propagation from one description to the

other. MDC introduces redundancy to the bitstream since a piece of information is partitioned into several independent descriptions. Hence, the total data rate is inevitably larger than that represented by a single description. MDC may be regarded as a joint source and channel coding method in which the key problem is to design judicious descriptions of a given video signal and thus efficiently allocate the total data rate among different descriptions.

1.1.2 Leaky Prediction Layered Video Coding (LPLC)

As discussed in the previous subsection, conventional layered scalable video coding results in poor coding efficiency due to the exclusion of the enhancement layer from the motion compensation operation. To circumvent this coding inefficiency, leaky prediction layered video coding (LPLC) [20–22] includes a scaled version of the enhancement layer within the motion compensation loop to improve the coding efficiency while maintaining graceful error resilience performance. A framework of the LPLC encoder is shown in Fig. 1.5, where the enhancement layer is coded in the FGS manner [20]. LPLC has attracted much attention in the literature recently due to its performance in handling the trade-off between coding efficiency and error drift. It provides a flexible coding structure, by utilizing a leaky factor, having a value between 0 and 1, to scale the enhancement layer before it is incorporated into the motion compensation loop. When the leaky factor, α, is 0, the enhancement layer is completely excluded from the motion compensation loop, resulting in a codec that has the least coding efficiency and the best error resilience performance. If, however,

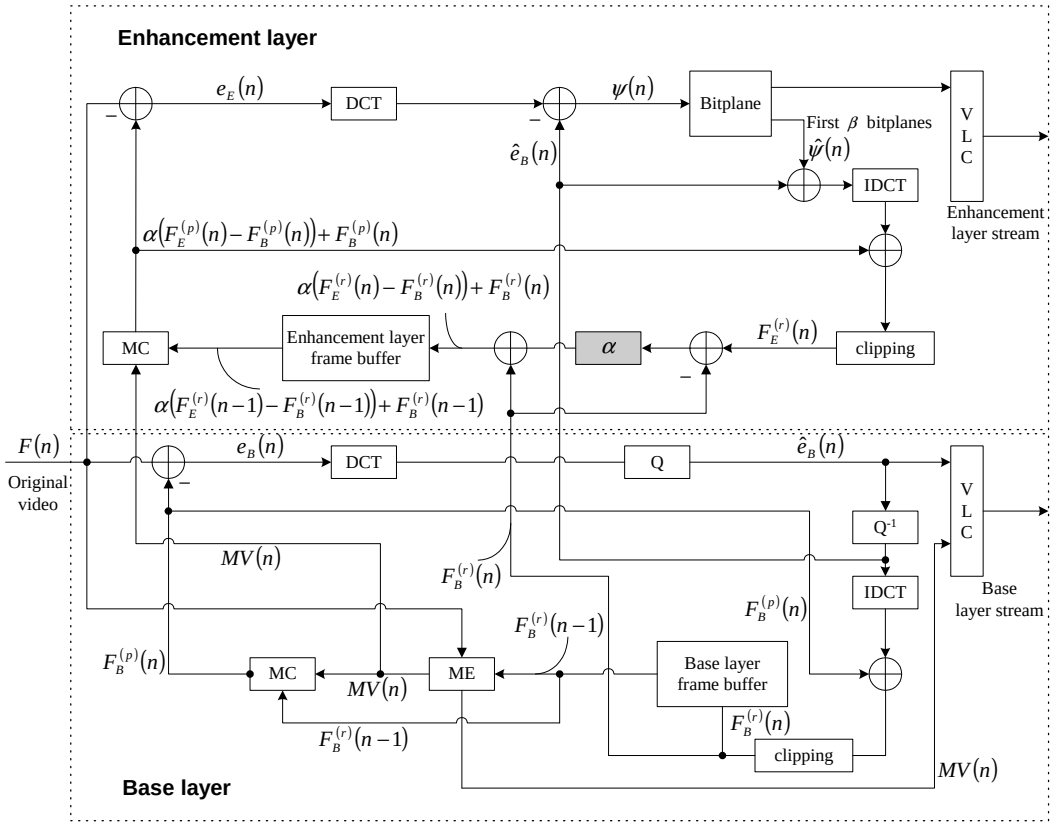


Fig. 1.5. A framework of the leaky prediction layered video coding (LPLC) encoder

$\alpha = 1$, then the codec has the best coding efficiency and the least error resilience. For intermediate values of α, the codec in essence trades off coding efficiency for error resilience.

Alternative approaches to improving the coding efficiency of conventional layered coding while mitigating potential error drift include the work presented in [23], [24], and [25]. These approaches usually couple a mode-adaptive scalable coding scheme with a drift-management system. A predefined mode may either exclude the enhancement layer from the motion compensation loop, or completely incorporate it

within the motion compensation loop, or linearly combine the two layers within the motion compensation loop. From the LPLC perspective, to adaptively select such a coding mode is equivalent to adaptively adjusting the leaky factor. Hence LPLC provides a more general framework, of which many state-of-the-art error resilient layered coding schemes are special instantiations. In addition, LPLC is easy to implement and incorporate into most layered coding structure.

1.1.3 Rate Distortion Optimization

According to Shannon's separation principle, for end-to-end delivery, the design of source coding and channel coding are separate if the following two conditions are satisfied [26]:

- The source data can be grouped into blocks of an arbitrarily long block length;
- Arbitrarily high computational complexity and delay can be tolerated.

However, the above conditions might not be met in practice and the theory is not applicable to a real video communication system. Joint source and channel coding schemes have hence been developed, which are usually optimized in an operational manner. The overall expected distortion of an arbitrary received video signal, denoted as $E(D)$, contains two elements: the distortion caused by video compression due to source quantization, which can be predicted at the encoder, and the distortion caused by channel errors, which is random and thus impossible to obtain a precise prediction at the encoder. If the overall data rate for the video communication sys-

tem is fixed as R_{budget}, the joint source and channel coding optimization problem can be formulated as follows

$$\min E(D), \quad \text{subject to } R_{\text{source}} + R_{\text{channel}} \leq R_{\text{budget}}. \tag{1.1}$$

The peak signal-to-noise ratio (PSNR) is often used as a metric to evaluate the decoded quality of an arbitrary video frame

$$\text{PSNR} = 10 \log_{10} \frac{255^2}{\text{MSE}}, \tag{1.2}$$

where MSE denotes the mean square error (MSE), which is used as a distortion measure and obtained as follows

$$\text{MSE} = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left| X(i,j) - \hat{X}(i,j) \right|^2, \tag{1.3}$$

where $M \times N$ denotes the frame size of a given video sequence, and $X(i,j)$ and $\hat{X}(i,j)$ denote the pixel intensity located in (i,j) of the original video frame and the reconstructed video frame, respectively.

The problem formulated in (1.1) is a rate-distortion optimization problem with a data rate budget constraint. A theoretical solution to this problem requires precise modeling of the source video signal and the channel loss behavior to obtain the bounds between the achievable and unachievable rate-distortion regions, as discussed in Shannon's work [26]. Nevertheless, these models are hard to obtain for complex signals such as video and the bounds are not practically constructive. Therefore, operational rate-distortion optimization is usually used in the literature, where the solution resorts to the numerical optimization methods [27, 28]. Generally, a specific

video communication system is built, and a rate-distortion optimization scheme is then used to adjust the system parameters to search for the best operating points. Compared to the theoretic results, these operating points are achievable associated with a specific system implementation, and the boundary between the achievable and unachievable regions resides in the convex hull of the set of the operating points.

Two efficient tools are often used to solve operational rate-distortion optimization problems, namely the Lagrangian multiplier optimization and dynamic programming. A simple example of Lagrangian optimization is to minimize the Lagrangian cost function $J(\lambda) = D + \lambda R$, where R denotes the allocated data rate and D denotes the distortion at a given Lagrangian multiplier λ. As illustrated in Fig. 1.6, the optimization of the Lagrangian cost is equivalent to finding the point at which the line with absolute slope λ is tangent to the convex hull of the rate-distortion characteristic. Bisection search is usually used to find the correct λ [29]. In contrast, a dynamic programming method usually first builds a tree or trellis structure with weighted cost associated with each branch of the tree or trellis and then finds the optimum path with the minimum cost.

The Lagrangian optimization technique is well suited to solving the optimization problems where the cost functions are continuous and differentiable. Practically, the operational points usually contribute a set with a discrete allocation of elements. If the convex hull is densely populated, we can still use the Lagrangian method as long as the gap between its solution and the optimal solution is sufficiently small. If the operating rate-distortion points are sparsely distributed and the optimal point

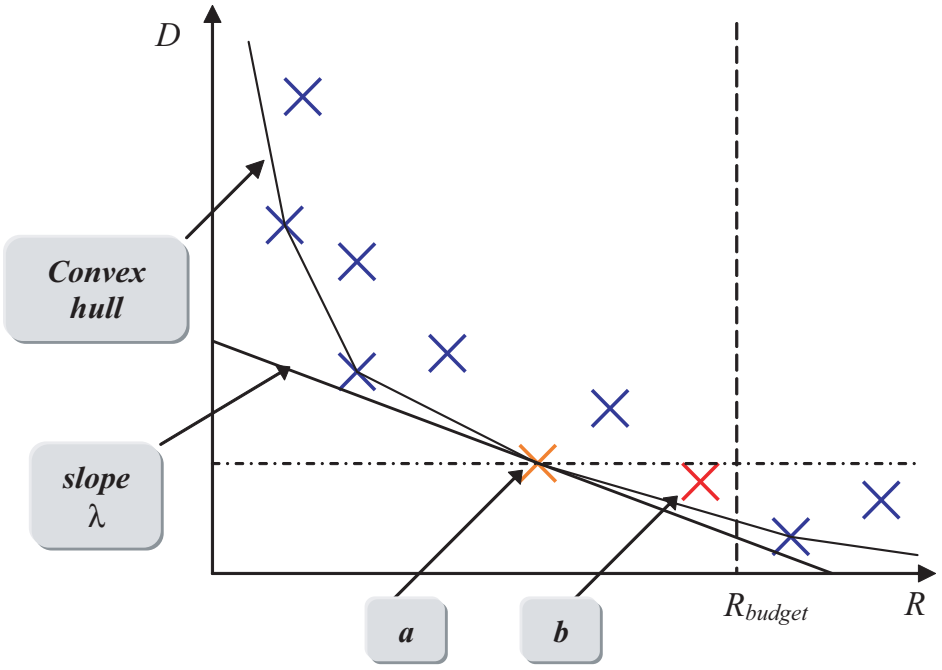


Fig. 1.6. Operational rate-distortion optimization by Lagrangian multiplier optimization and dynamic programming

does not reside in the convex hull, as indicated by the point "b" in Fig. 1.6, the Lagrangian solution is not optimal (which instead results in the solution indicated by the point "a" in the figure). In this case, we have to resort to other optimization methods such as dynamic programming.

Dynamic programming is an optimization strategy to thoroughly search the entire data space for the optimal solution in an efficient way. Two conditions have to be satisfied when using dynamic programming to a specific optimization problem: (i) overlapping subproblems, where the solution space associated with each subproblem is sufficiently small, and (ii) optimal structure, meaning that the solutions to the subproblems should also contribute to the final optimal solution. Dynamic program-

ming takes advantage of the above two features to search for the optimal solution in a bottom-up manner where the subproblems are first solved and the solutions are stored in a table for look-up when needed.

Dynamic programming is time-consuming, and the computational complexity grows exponentially in terms of the dimension of the problem space, compared to the Lagrangian method where the complexity grows linearly. Hence, the Lagrangian method is often used when the data space can be assumed as densely allocated, or a Lagrangian solution is used as an initial value for dynamic programming. Also, greedy algorithms are often used to substitute dynamic programming, where the computational complexity is reduced but the global optimal solution is approximated by a local solution. A greedy algorithm also applies to an optimization problem that has an optimal structure. Nevertheless, the greedy algorithm always tries to select the best solution at the current stage, namely the "greedy-choice", and this is often referred to as the greedy-choice property. Only when all the greedy-choices are part of a global solution can a greedy algorithm achieve the final optimal solution.

1.1.4 Low Complexity Video Encoding

Current video coding standards are highly asymmetrical. Encoding is typically 5-10 times more complex than decoding. This is due to the use of inter-frame predictive coding, which is desirable for consumer electronics (CE) applications including DVD (Digital Versatile Disk) and DTV (Digital Television), video streaming, and video-on-demand (VOD), as it can result in high compression ratios.

Fig. 1.7. A video surveillance system with distributed cameras

New video coding applications have emerged such as wireless sensor networks, wireless PC cameras, and mobile video cameras for video surveillance as shown in Fig. 1.7. A common characteristic of these applications is that resources for memory, computation, and energy are scarce at the video encoder whereas resources at the decoder can be relatively abundant. Since conventional video coding approaches cannot meet the requirements of these new emerging applications, it is necessary to develop alternative video coding methods that have low complexity at the encoder but allow high complexity at the decoder. We refer to such video coding approaches as low complexity video encoding approaches.

Low complexity video encoding requires that the computational complexity be shifted from the encoder to the decoder. One way to implement this is to replace an inter-frame encoder-decoder, which uses both inter-frame encoding and inter-frame

decoding as in the case of the conventional video coding, with an intra-frame encoder but inter-frame decoder. This implies that the implementation of motion estimation is shifted from the encoder to the decoder, and the motion vectors are only used by the decoder.

In conventional video source coding, the encoder usually uses extra information, statistical or syntactic, to encode the source. This extra information is known as "side information." Side information may be supplied to the encoder, or guessed or measured by the encoder. A classical example of side information in conventional video coding is motion information symbolized by motion vectors. Hence, the side information is known to both the encoder and the decoder for conventional video coding. In contrast, low complexity video encoding may require that the side information be known only to the decoder. Therefore, how to obtain the side information at the decoder, instead of at the encoder, is the most essential problem to be addressed for low complexity video encoding approaches.

Distributed source coding has provided a coding paradigm that allows side information to be used only at the decoder. Given two arbitrary source symbols, one source may serve as the side information to the other. Using distributed source coding, the two source symbols can be encoded independently but decoded jointly. The ideal situation for distributed source coding is to achieve a coding efficiency the same as that of using joint encoding and joint decoding, i.e., the same as that of conventional source coding. A comparison between conventional source coding and distributed source coding is given in Table 1.1.

Table 1.1
Source coding paradigms: distributed source coding vs. conventional source coding

Distributed source coding	**Conventional source coding**
Low complexity in the encoder	High complexity in the encoder
Side information is only known to the decoder	Side information is known to both encoder and decoder
Applications: Well-suited for wireless mobile terminals	*Applications*: Well-suited for broadcasting and CE applications
Theoretical basis: Slepian-Wolf Theorem Wyner-Ziv Theorems	*Theoretical basis*: Channel Coding Theorem Rate-Distortion Theory

1.1.5 Error Resilient Video Coding

To combat lossy channel errors, strategies have been developed in the application layer of a video communication system [30]. The strategies can be classified into three categories: error resilience, error control coding, and error concealment. Error resilience refers to schemes that introduce error resilient elements at the stage of video compression to mitigate error propagation. As previously discussed, video compression removes redundancies that exist in the video sequence. Since differential coding and motion compensation are widely used to improve the coding efficiency, a large amount of dependency exists across different segments of the encoded bitstream, which makes the bitstream very vulnerable to the bit errors. By using error resilience schemes such as forced intra mode, slice structure, or data partitioning, the dependency is reduced across different information bits, thus preventing error propagation from one portion of the bitstream to the other. Inevitably, the use of error resilience will degrade the coding efficiency.

Error control coding schemes refer to error protection by using channel coding in the application layer. As shown in Fig. 1.1, channel coding is usually implemented in the physical layer. Contrary to source coding, channel coding inserts redundant information into the bitstream, which helps detect and correct errors due to the channel noise corruption. When video signals are transmitted over packet networks or wireless channels, serious corruption might occur to the bitstream, which is caused by the burst packet loss or burst bit errors due to network congestion or channel

fading. Hence, it is wise to introduce channel coding in the application layer to obtain better error protection. Forward Error Correction (FEC) generates redundant parity data to the bitstream, and has been widely used in the application layer for error detection and correction. In particular, Reed-Solomon (RS) coding is an ideal FEC scheme since RS codes are maximum distance separable codes which are well suited for error protection against burst errors.

Essentially, the combination of error resilience and error control coding falls into the category of joint source and channel coding design, which requires two types of optimized data rate allocation: (i) the optimized data rate allocation between the source coding and the channel coding, and (ii) the optimized allocation of the data rate among source code elements to introduce an appropriate amount of error resilience into the bitstream.

Compared to the former two schemes which actively place error protection at the encoder side, error concealment is a passive scheme for error recovery at the decoding stage. In the literature, the temporal-replacement method is often used, where the motion vectors of current lost block are interpolated from the motion vectors of the neighboring blocks, if available, and the motion compensated macroblock using the interpolated motion vectors is taken to replace the lost macroblock. Other error concealment approaches include the work in [31] and [32, 33].

We compare error resilience and error control coding schemes in Table 1.2, where UEP denotes unequal error protection, indicating that different amounts of error protection are applied to different portions of the encoded bitstream. As opposed to

Table 1.2
Comparison of error resilience and error control coding schemes

Error resilience	**Error control coding**
Advantages: - Standard compliant: no additional software needed at the client - No additional delay - Work better when channel errors are burst and channel conditions are not known as *a priori* - Result in graceful degradation when channel conditions getting worse	*Advantages*: - Detect/correct errors - Yield better performance when physical layer FEC is not sufficient - Able to take advantage of UEP
Disadvantages: - Lessen error propagation, but not correct errors - Make compression performance go down	*Disadvantages*: - Additional bandwidth - Additional delay - Additional software needed at the client

UEP, equal error protection (EEP) represents the error protection where all segments of the bitstream are afforded equal protection. UEP allows video transmission with priority over current data networks that have no QoS (quality of service) guaranteed. For bitstreams generated by an encoder that uses the DCT and motion compensation techniques, three kinds of information play the most significant role in the decoded video quality. These are the header information, the motion vectors, and the DC (the transform coefficient located in $(0, 0)$) and low frequency DCT coefficients. How to protect these kinds of information is more critical than protecting the rest part of the bitstream. Considering the total data rate constraint, UEP can exploit the available data rate in a more efficient way and achieve a better error protection performance.

In addition to the strategies developed in the application layer, appropriate transmission protocols can be used to protect video bitstreams from being corrupted by channel errors. Cross-layer design has newly emerged as an approach to jointly design video compression, channel coding, and network transmission protocols to realize the efficient and robust transmission of digital videos [34].

We summarize the strategies for robust video transmission in Fig 1.8.

1.1.6 Video Coding Standard - JVT/H.264/AVC

Video coding standards allow inter-operability between different manufacturers and various equipment worldwide [35]. The scope of video coding standardization only defines the syntax and decoder, which has two advantages: (i) allowing op-

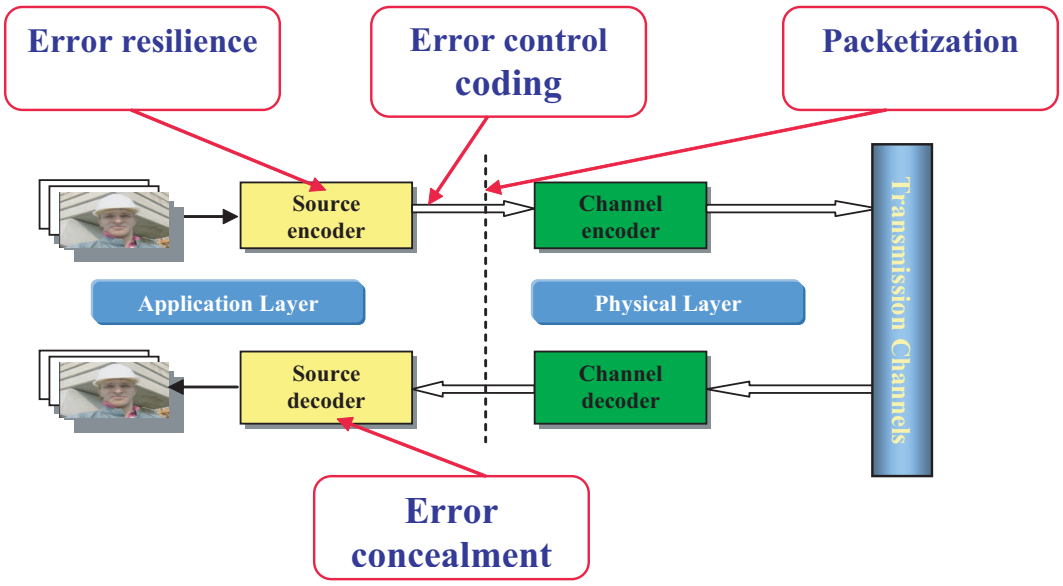


Fig. 1.8. Strategies of error resilient video coding

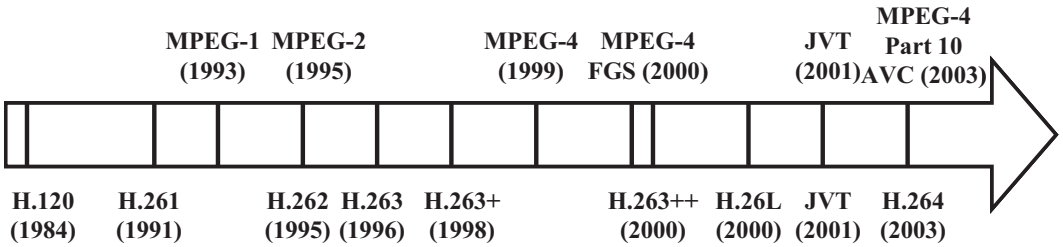


Fig. 1.9. Timeline of video coding standards (MPEG-2/H.262 and MPEG-4 Part 10 AVC/H.264 were joint projects)

timization beyond the obvious; (ii) allowing computational complexity reduction concerning the implementation.

Two organizations have strongly influenced the development of video coding standards: the Video Coding Experts Group (VCEG) of ITU-T (Telecommunication Standardization Sector of International Telecommunications Union) and the Moving Picture Experts Group (MPEG) of ISO/IEC (International Organization for Standardization). Video coding standards developed by ITU-T are designated by the label "H.26x", and standards developed by MPEG are designated by the label "MPEG-x." The timeline of video coding standards is given in Fig. 1.9.

H.26L was a standard developed by the ITU-T VCEG, with the first test model released in August 1999. In December 2001, the Joint Video Team (JVT) was formed between the ITU-T VCEG and the ISO/IEC MPEG to work on H.26L as a joint project that is similar to the establishment and development of MPEG-2. The JVT project was finalized in December 2003, and the new standard was approved as H.264, as a new ITU-T recommendation, and also as MPEG-4 Part 10 AVC

(Advanced Video Coding), as a new part of ISO/IEC MPEG-4. It is referred to as H.264/AVC in short [36, 37]. Documents and verification codecs for H.26L were offered by the Test Model Long Term (TML), and most recent released documents and verification codecs for H.264/AVC are available online [38].

Many video coding standards contain different configurations of capabilities, which are specified by the so-called "profiles" and "levels." A *profile* is a set of algorithmic features, while a *level* is a degree of capability. H.264/AVC currently has three profiles: (i) Baseline, which addresses a broad range of applications, in particular those requiring low latency; (ii) Main, which adds features such as interlace, B-Slices, and CABAC (context-based adaptive binary arithmetic coding) [39]; (iii) Extended, referred to as the streaming profile. H.264/AVC has as many as 14 levels, built to match popular international production and emission formats. The video format can range from QCIF to the format for digital cinema applications.

As in the previous video coding standards, H.26L/H.264 specifies a hybrid video coding implementation using block-based transform coding and motion estimation, as illustrated in Fig. 1.10. H.26L/H.264 however contains more features in its video coding layer (VCL) that further improve the coding efficiency at all data rates, and fulfill several tasks in the network abstraction layer (NAL) that improve the error resilience performance of the bitstream [40–42].

One novel feature that is available in H.26L/H.264 is in-the-loop deblocking filtering [43], as shown in Fig. 1.10. Block-based video coding produces artifacts known as blocking artifacts. These can originate from both the block-based motion com-

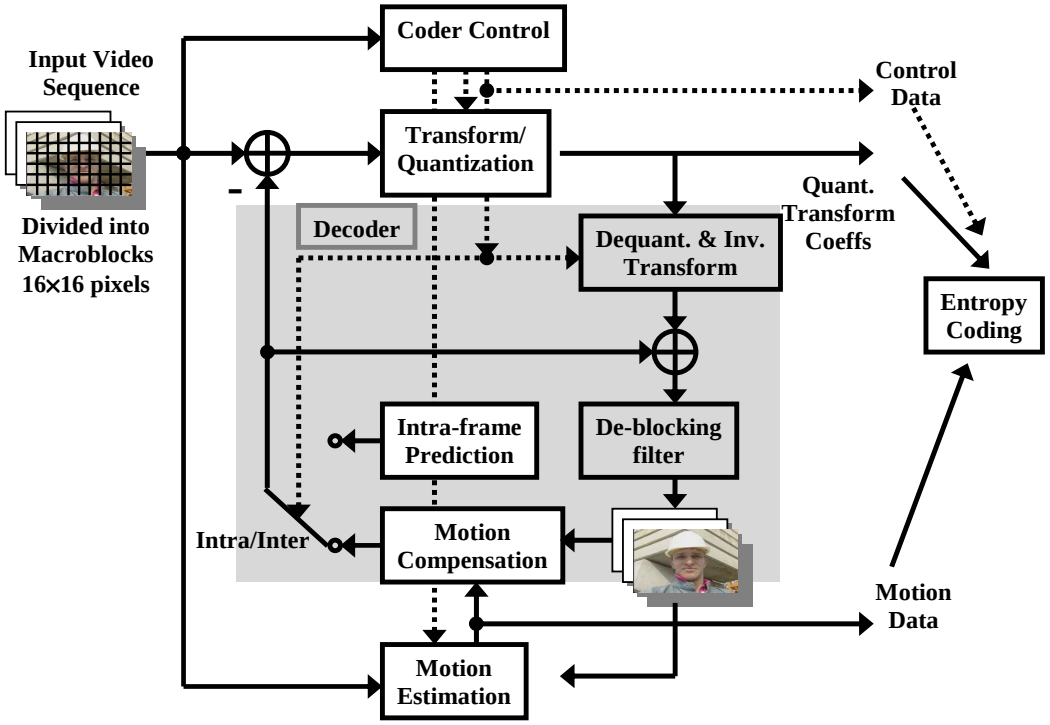


Fig. 1.10. Basic macroblock coding structure in H.26L/H.264

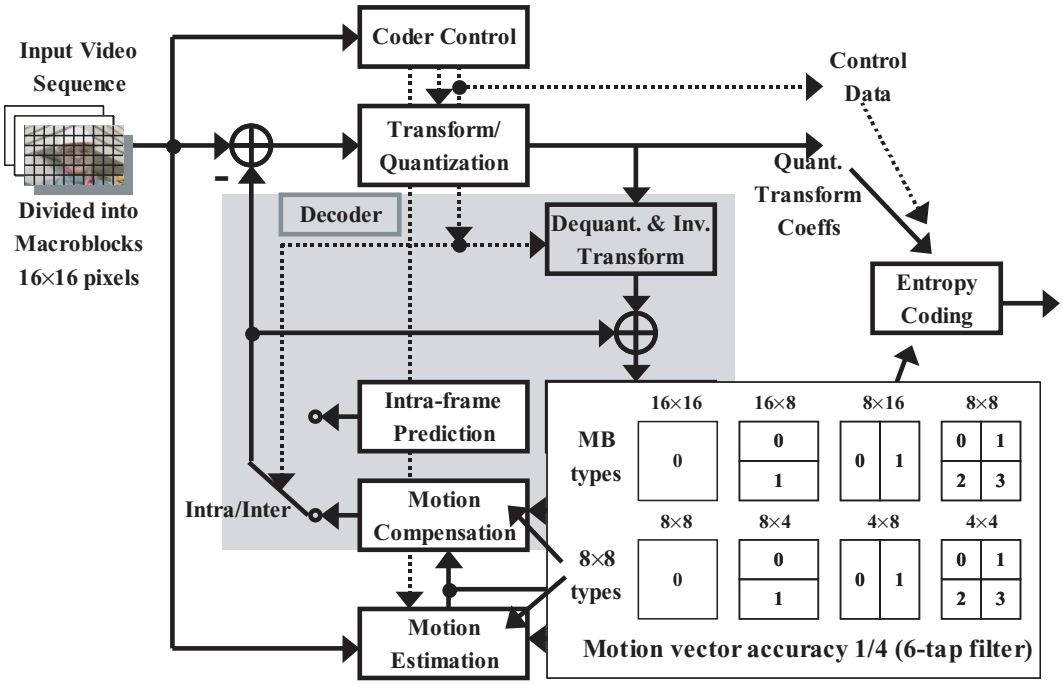


Fig. 1.11. Motion compensation accuracy in H.26L/H.264

pensated prediction and the block-based PEF coding stages. The use of deblocking filtering improves the resulting video quality, both objectively and subjectively. The deblocking filter in the H.264/AVC design within the motion compensated prediction loop further enables the improvement in quality to be used in inter-picture prediction, hence improving the ability to predict other pictures as well.

H.26L/H.264 uses seven block shapes for motion prediction, obtains motion vectors up to 1/4 or 1/8 pel resolution, and utilizes multiple reference frames to implement multihypothesis motion prediction for each frame, as illustrated in Fig. 1.11 and Fig. 1.12 [44, 45].

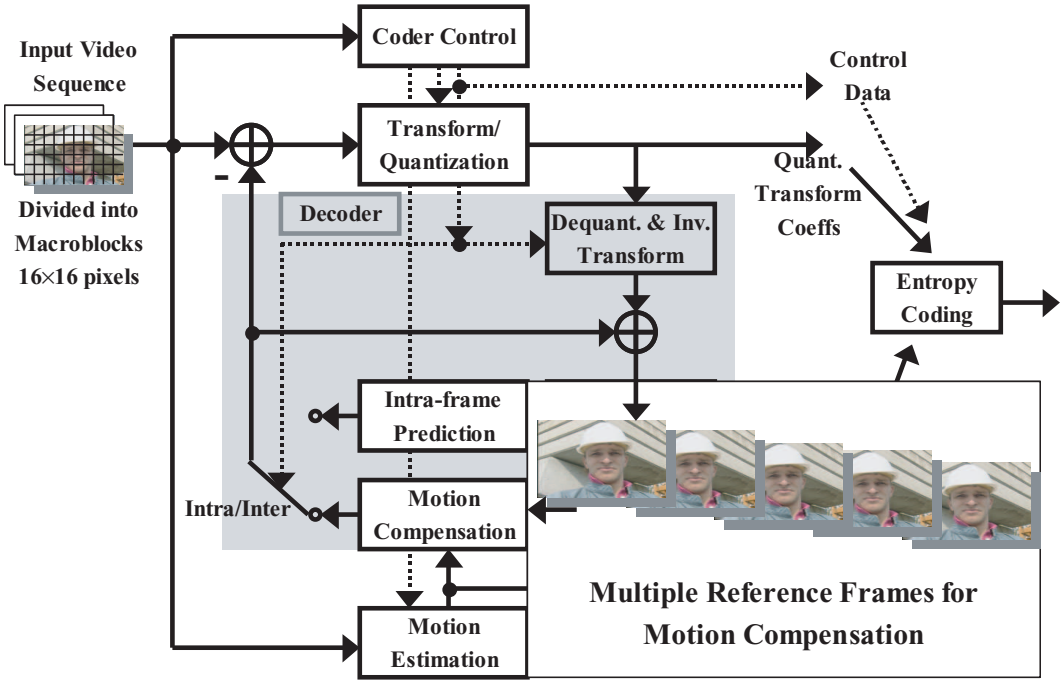


Fig. 1.12. Multiple reference frames in H.26L/H.264

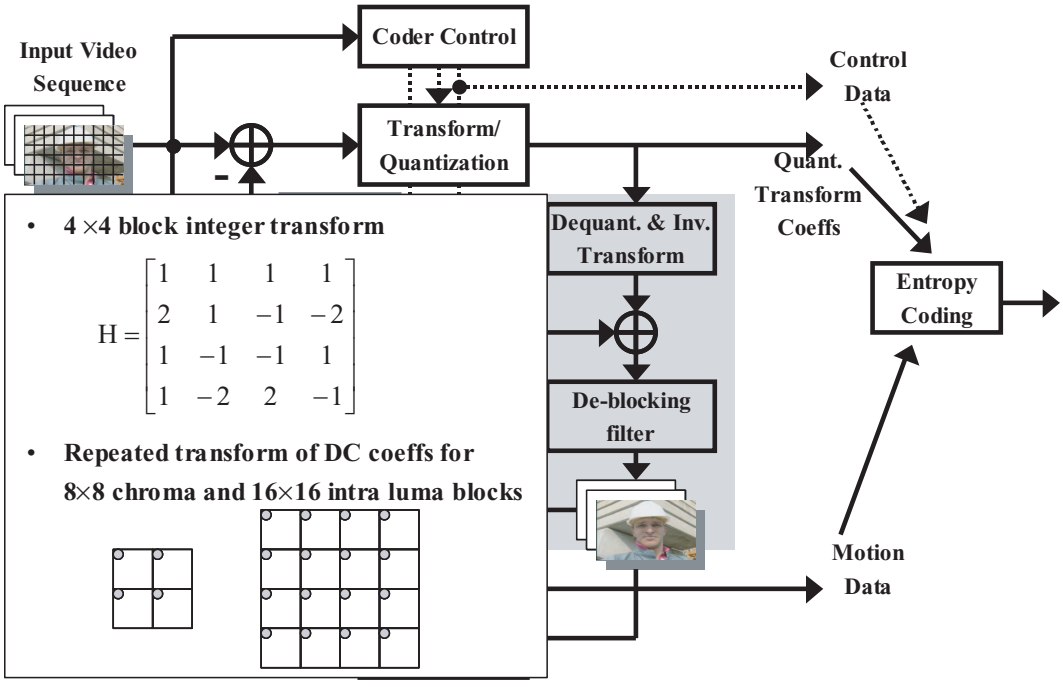


Fig. 1.13. Transform coding in H.26L/H.264

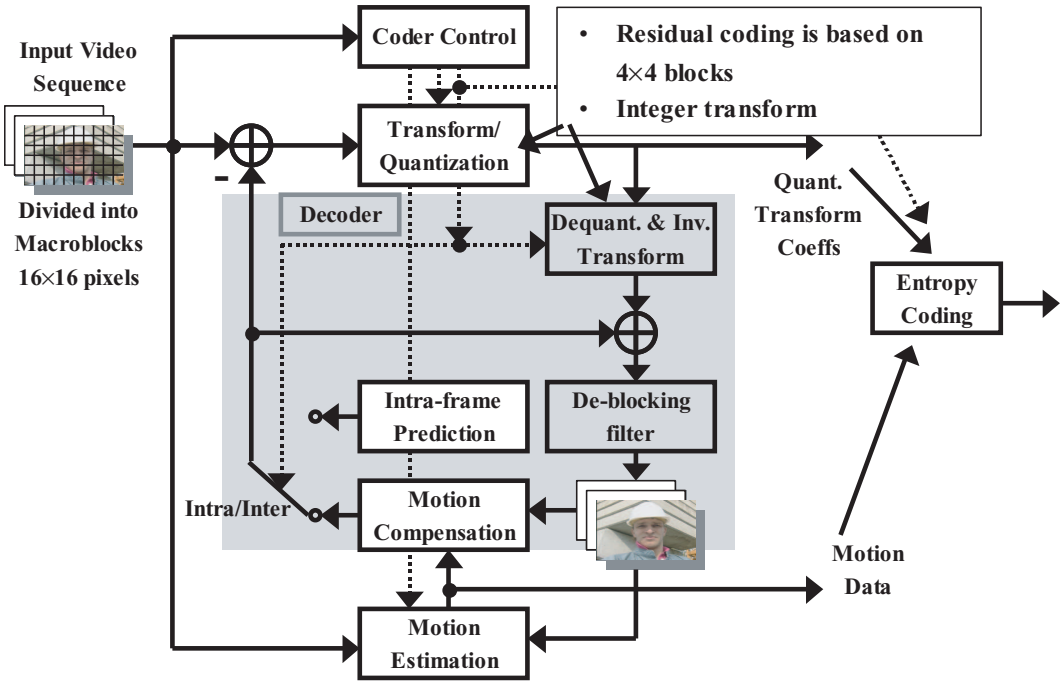


Fig. 1.14. Residual coding in H.26L/H.264

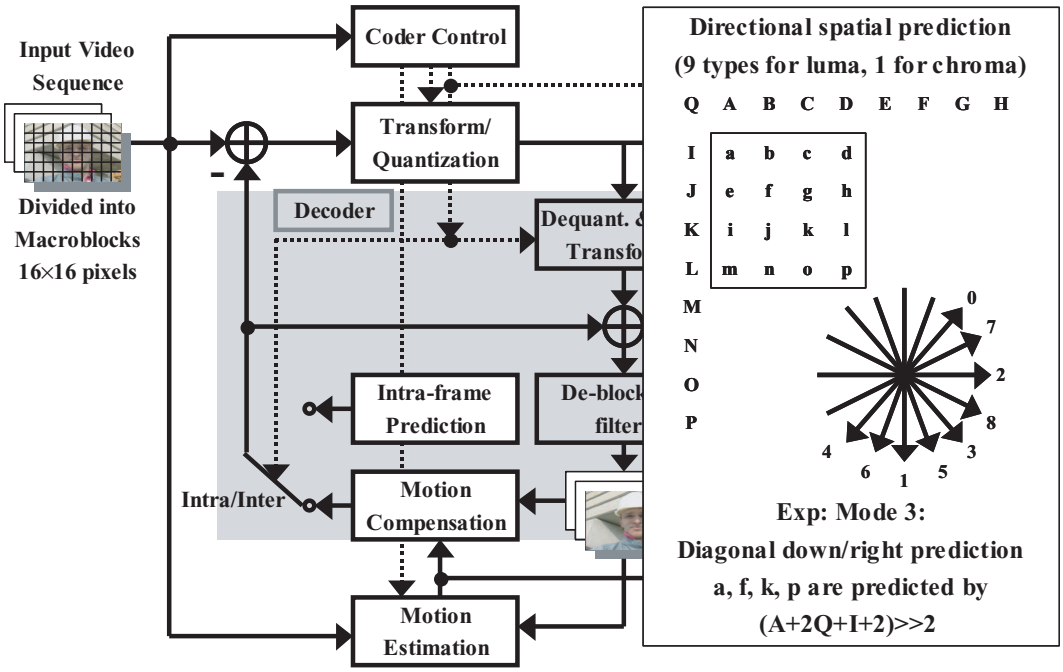


Fig. 1.15. Intra prediction in H.26L/H.264

H.26L/H.264 also uses a 4×4 integer transform with DCT-like properties to eliminate the rounding mismatch problem in the inverse transform, as illustrated in Fig. 1.13 and Fig. 1.14 [46].

H.26L/H.264 exploits directional spatial prediction for intra coding and proposes a new technique of extrapolating the edges of the previously-decoded blocks of the current picture to predict the current intra-coded blocks, as illustrated in Fig. 1.15.

As shown in Fig. 1.16, in addition to the conventional universal variable length coding (UVLC), two new entropy coding modes are designed in H.26L/H.264, the context-based adaptive binary arithmetic coding (CABAC) and the context adaptive variable length coding (CAVLC) [39].

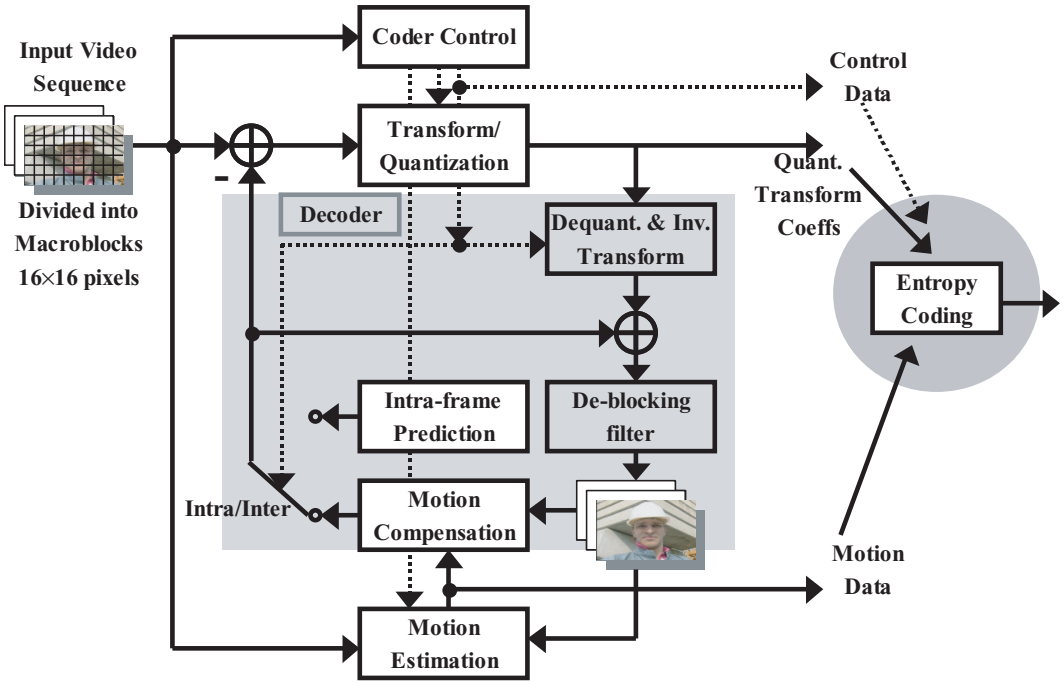


Fig. 1.16. Entropy coding in H.26L/H.264

In the verification model of H.26L/H.264, Lagrangian rate-distortion optimization is recommended in both motion vector selection and macroblock coding mode decision. Closed-form relations between the Lagrangian parameter and the chosen quantization parameter are formulated. Strategies for the rate control and efficient decoding of H.264/AVC bitstreams are addressed in [47–50].

1.2 Organization of The Dissertation

The dissertation is organized as follows: We address an enhancement of LPLC with respect to its rate distortion performance in Chapter 2, where we highlight a deficiency of LPLC, and propose a general framework that applies to both LPLC and a multiple description coding scheme using motion compensation to further confirm the existence of the deficiency. We further propose an enhanced LPLC based on maximum-likelihood estimation to address the previously specified deficiency in LPLC.

We develop the theoretic analysis of LPLC in Chapter 3, and present two different approaches, one using rate distortion theory and another using a quantization noise model. We derive two sets of rate distortion functions in closed form for LPLC, and evaluate the theoretical results in comparison to the operational results.

In Chapter 4, we describe the schemes that introduce nested scalability in the parallel scalable coding structure, and propose a coding structure characterized as "dual-leaky", which combines nested scalability and parallel scalability under one framework.

We address low complexity video encoding in Chapter 5, where we propose a low complexity video encoding approach using new B-frame direct modes. In particular, we design three B-frame direct modes and nine coding modes for encoding macroblocks in B frames. We demonstrate that our approach provides a competitive rate distortion performance compared to that of the high complexity video encoding approach.

We also address reliable video transmission in Chapter 6 and Chapter 7. In Chapter 6, we present a thorough evaluation of the joint source-channel video coding methodology over wireless networks, and investigate new metrics other than the common PSNR to evaluate video distortion caused by the combination of source compression and channel errors. In Chapter 7, we achieve reliable video transmission using adaptive packetization and a two-layer rate-distortion optimization scheme.

Conclusions and future work are provided in Chapter 8.

2. AN ENHANCEMENT OF LEAKY PREDICTION LAYERED VIDEO CODING

2.1 Introduction

In this chapter, we focus on leaky prediction layered video coding (LPLC). LPLC includes a scaled version of the enhancement layer within the motion compensation loop to improve the coding efficiency while maintaining graceful recovery in the presence of error drift. However, there exists a deficiency inherent in the LPLC structure, namely that the reconstructed video quality from both the enhancement layer and the base layer cannot be guaranteed to be always superior to that of using the base layer alone, even when no drift occurs. In this chapter: (1) We highlight this deficiency using a formulation that describes LPLC, and (2) propose a general framework that applies to both LPLC and a multiple description coding scheme using motion compensation. We use this framework to further confirm the existence of the deficiency in LPLC. (3) Furthermore, we propose an enhanced LPLC based on maximum-likelihood estimation to address the previously specified deficiency in LPLC. We then show how our new method performs compared to LPLC [51, 52].

Layered video coding has a nested structure whereby different levels of the bit-stream are decoded in a fixed sequential order. Fine granularity scalability (FGS)

is a specific layered scalable coding structure, which possesses fully rate (or SNR) scalability over a wide range of data rates [4, 5]. Layered coding is desired for error resilient video streaming over heterogeneous networks with changing bandwidth mainly because: (1) It can be adapted to varying channel bandwidth by simply discarding the higher layer(s), or, by truncating the bitstream when coded using FGS; (2) It allows one to protect parts of the bitstream differently, i.e., to use unequal error protection (UEP).

For error resilient video transmission in an error-prone environment, error protection can be used for the base layer since it carries more significant information. This achieves a trade-off between coding efficiency and robustness. For example, the base layer bitstream could be protected by Forward Error Correction (FEC) coding, or transmitted using an error-recovery capable network protocol such as TCP [6, 7]. The enhancement layer however still remains vulnerable to errors. Due to the potential incompleteness or destruction of the enhancement layer, traditional layered coding schemes usually do not incorporate the enhancement layer into the motion compensation (MC) loop at the encoder to prevent error drift at the decoder. This results in poor coding efficiency, when compared to non-scalable coding, since the high-quality reconstruction offered by both the enhancement layer and the base layer is not exploited by the MC operation.

To circumvent this coding inefficiency, leaky prediction layered video coding (LPLC) [20–22] includes a scaled version of the enhancement layer within the MC loop to improve the coding efficiency while maintaining graceful error resilience per-

formance. LPLC has attracted much attention in the literature recently due to its performance in handling the trade-off between coding efficiency and error drift. It provides a flexible coding structure, by utilizing a leaky factor, having a value between 0 and 1, to scale the enhancement layer before it is incorporated into the motion compensation loop. When the leaky factor, α, is 0, the enhancement layer is completely excluded from the motion compensation loop, and LPLC thus becomes the conventional layered video coding structure. This results in a codec that has the least coding efficiency and the best error resilience performance. If, however, $\alpha = 1$, then the codec has the best coding efficiency[1] and the least error resilience. For intermediate values of α, the codec in essences trades off between coding efficiency and error resilience.

Alternative approaches to improving the coding efficiency of the conventional layered coding while mitigating potential error drift include the work presented in [23], [24], and [25]. These approaches usually couple a mode-adaptive scalable coding scheme with a drift-management system. A predefined mode may either exclude the enhancement layer from the MC loop, or completely incorporate it within the MC loop, or linearly combine the two layers within the MC loop. From the LPLC perspective, to adaptively select such a coding mode is equivalent to adaptively adjusting the leaky factor. Hence LPLC provides a more general framework, of which many state-of-the-art error resilient layered coding schemes are special instantiations.

[1]The discussion of the coding efficiency of LPLC for different leaky factors will be explored in-depth in this chapter.

In addition, LPLC is easy to implement and incorporate into most layered coding structure.

In this chapter, we describe a deficiency in terms of the coding efficiency inherent in the LPLC structure, namely that it cannot guarantee that the decoded video quality obtained from both the enhancement layer and the base layer will always be superior to that offered by the base layer alone. In other words, the enhancement layer does not always "enhance" the performance. We will analytically and experimentally demonstrate this deficiency, and confirm it by addressing the similarity between LPLC and a multiple description coding (MDC) scheme, namely MDMC [53]. We further establish a general framework that applies to both LPLC and MDMC. Using this framework, we utilize maximum-likelihood (ML) estimation, originally developed for MDC, and propose a new approach we refer to as ML-LPLC. It will be shown that ML-LPLC always achieves a superior performance beyond, or at least as good as, the better of the reconstructions obtained using the base layer alone or using both the enhancement layer and the base layer when both layers are available at the decoder.

We implemented LPLC and ML-LPLC using two video codecs: a hybrid fully rate scalable codec using the wavelet transform and motion compensation [5], and the ITU-T H.26L version TML9.4 [54]. The experimental results from both implementations show that ML-LPLC achieves a gain of up to 1 dB in average PSNR.

In Section 2.2, we review the structure of layered coding as well as LPLC. The deficiency of LPLC is addressed in Section 2.3. We also present the similarity between

LPLC and MDMC in the section, and establish a general framework applied to both schemes. We discuss our ML-LPLC approach in Section 2.4. The experimental results and further discussions are given in Section 2.5 and Section 2.6 concludes the chapter.

2.2 Overview of LPLC

Consider a two-layer scalable coding structure. Let $F(n)$ denote the original nth frame of a video sequence, and $F_B^{(p)}(n)$ the predicted image using motion compensation based on the base layer of the reference frame. The predicted error frame (PEF) between $F(n)$ and $F_B^{(p)}(n)$, $e_B(n)$, is then obtained as

$$e_B(n) = F(n) - F_B^{(p)}(n). \tag{2.1}$$

The base layer of frame n contains the quantized version of $e_B(n)$

$$\hat{e}_B(n) = \text{Quant}\left\{e_B(n)\right\}, \tag{2.2}$$

where Quant$\{\cdot\}$ denotes the quantization/dequantization operation[2]. In general, in addition to $\hat{e}_B$, the motion vectors are also associated with the base layer since they are critical in a video coding technique that uses motion estimation/compensation.

The reconstruction using the base layer, denoted as $F_B^{(r)}$, is

$$F_B^{(r)}(n) = F_B^{(p)}(n) + \hat{e}_B(n). \tag{2.3}$$

[2]It is to be noted that our analysis does not differentiate whether a signal is represented in the spatial domain, or in the frequency (e.g., DCT) domain. This is because, discounting computational precision, the orthogonal transforms used in video/image compression standards are reversible.

At the encoder, $F_B^{(r)}$ is stored in a frame buffer for the encoding of the next frame. The motion compensated image derived from the previously buffered frame is

$$F_B^{(p)}(n) = \mathrm{MC}_{MV(n)} \left\{ F_B^{(r)}(n-1) \right\}, \tag{2.4}$$

where $\mathrm{MC}_{MV(n)}\{\cdot\}$ indicates motion compensation using the set of motion vectors $MV(n)$.

Due to the use of error resilient strategies such as UEP, the base layer is usually assumed to stay intact when transmitted over error-prone channels, and thus the reconstruction using the base layer at the decoder side is identical to $F_B^{(r)}$ as indicated in (2.3).

The enhancement layer in conventional layered coding is the quantized version of the residue between the original image and the reconstructed base layer

$$\begin{aligned} \psi_E(n) &= F(n) - F_B^{(r)}(n) \\ &= F(n) - F_B^{(p)}(n) - \hat{e}_B(n) \\ &= e_B(n) - \hat{e}_B(n), \end{aligned} \tag{2.5}$$

$$\hat{\psi}_E(n) = \mathrm{Quant}\left\{ \psi_E(n) \right\}, \tag{2.6}$$

where ψ_E denotes the residue and $\hat{\psi}_E$ denotes its quantized version.

In contrast to the reconstruction using the base layer alone, the reconstruction using both the base layer and the enhancement layer might have different video

qualities at the encoder and the decoder. The reconstruction using both layers at the encoder, denoted as $F_E^{(r,enc)}$, or just as $F_E^{(r)}$, is

$$\begin{aligned} F_E^{(r)}(n) &= F_B^{(r)}(n) + \hat{\psi}_E(n) \\ &= F_B^{(p)}(n) + \hat{e}_B(n) + \hat{\psi}_E(n). \end{aligned} \tag{2.7}$$

The decoder, on the other hand, might receive a different version of $\hat{\psi}_E$ due to channel errors or the truncation of the enhancement layer to adapt to the channel bandwidth. Thus, a different version of $F_E^{(r)}$, denoted by $F_E^{(r,dec)}$, is reconstructed at the decoder, which is given by

$$\begin{aligned} F_E^{(r,dec)}(n) &= F_B^{(r)}(n) + \breve{\psi}_E(n) \\ &= F_B^{(p)}(n) + \hat{e}_B(n) + \breve{\psi}_E(n), \end{aligned} \tag{2.8}$$

where $\breve{\psi}_E$ denotes the received version of $\hat{\psi}_E$.

Since both the base layer and the enhancement layer are obtained using the buffered reconstructed base layer $F_B^{(r)}(n-1)$ when encoding the nth frame, the inconsistency between $\hat{\psi}_E(n)$ and $\breve{\psi}_E(n)$ will not affect the frames following frame n, i.e., there is no error drift as long as $\hat{e}_B$ is received correctly. Nevertheless, the coding efficiency is degraded when compared to non-scalable coding, since the enhancement layer, which is able to produce superior quality reconstruction together with the base layer, is excluded from the MC loop.

LPLC exploits the same coding procedure to encode and reconstruct the base layer, as presented in equations (2.1)-(2.4), but obtains the enhancement layer in a different way. In addition to $F_B^{(r)}$, LPLC also buffers $F_E^{(r)}$, the reconstruction

by both the enhancement layer and the base layer, and then obtains the following reconstruction using the two buffered frames

$$\Gamma_E^{(r)}(n) \triangleq F_B^{(r)}(n) + \alpha \left(F_E^{(r)}(n) - F_B^{(r)}(n) \right), \tag{2.9}$$

where $\alpha \in [0, 1]$ indicates the leaky factor introduced by LPLC. When $\alpha = 1$, $\Gamma_E^{(r)}$ equals $F_E^{(r)}$. Generally, the quality of $F_E^{(r)}$ should be superior to $F_B^{(r)}$[3]. LPLC exploits α to scale the gain achieved by the enhancement layer, and hence the quality of the reconstruction $\Gamma_E^{(r)}$ should be located somewhere between that of $F_B^{(r)}$ and that of $F_E^{(r)}$. LPLC further obtains a motion compensated image $\Gamma_E^{(p)}$ as

$$\begin{aligned} \Gamma_E^{(p)}(n) &= \mathrm{MC}_{MV(n)} \left\{ \Gamma_E^{(r)}(n-1) \right\} \\ &= F_B^{(p)}(n) + \alpha \left(F_E^{(p)}(n) - F_B^{(p)}(n) \right), \end{aligned} \tag{2.10}$$

where

$$F_E^{(p)}(n) = \mathrm{MC}_{MV(n)} \left\{ F_E^{(r)}(n-1) \right\}. \tag{2.11}$$

In a similar way as indicated in (2.5), LPLC obtains the following residue

$$\psi_E(n) = F(n) - \Gamma_E^{(p)}(n) - \hat{e}_B(n), \tag{2.12}$$

and utilizes its quantized version, $\hat{\psi}_E$, in the enhancement layer, where

$$\hat{\psi}_E(n) = \mathrm{Quant} \left\{ \psi_E(n) \right\}. \tag{2.13}$$

LPLC then reconstructs the enhancement layer at the encoder using

$$F_E^{(r)}(n) = \Gamma_E^{(p)}(n) + \hat{e}_B(n) + \hat{\psi}_E(n), \tag{2.14}$$

[3]We will discuss the superiority of $F_E^{(r)}$ in more detail in Section 2.3.

and at the decoder using

$$F_E^{(r,dec)}(n) = \Gamma_E^{(p,dec)}(n) + \hat{e}_B(n) + \breve{\psi}_E(n), \tag{2.15}$$

where $\breve{\psi}_E$ denotes the received residue at the decoder, and $\Gamma_E^{(p,dec)}(n)$ is the motion compensated image, obtained at the decoder, of the reconstruction $\Gamma_E^{(r,dec)}(n-1)$ where

$$\Gamma_E^{(r,dec)}(n) = F_B^{(r)}(n) + \alpha \left(F_E^{(r,dec)}(n) - F_B^{(r)}(n) \right). \tag{2.16}$$

When $\alpha = 0$, both $\Gamma_E^{(r)}$ in (2.9) and $\Gamma_E^{(r,dec)}$ in (2.16) become $F_B^{(r)}$, and consequently $\Gamma_E^{(p)}$ and $\Gamma_E^{(p,dec)}$ become $F_B^{(p)}$. In this case, LPLC reduces to the conventional layered coding structure, resulting in no error drift but poor coding efficiency. When $\alpha > 0$, an improvement in coding efficiency is achieved using the better quality $\Gamma_E^{(r)}$ instead of $F_B^{(r)}$. In terms of error resilience, it turns out that the error drift caused by one frame degrades exponentially with time when $0 < \alpha < 1$ [20]. The smaller α, the faster the degradation. Therefore, a significant advantage of LPLC is the trade-off between coding efficiency and error resilience facilitated by α.

2.3 Further Analysis of LPLC

In this section, we first point out a deficiency in terms of coding efficiency inherent in LPLC, namely that the reconstructed quality using both the enhancement layer and the base layer in LPLC cannot be guaranteed to be always superior to that using the base layer alone. Moreover, we address the similarity between LPLC and

a motion compensated MDC scheme, known as MDMC, and use it to confirm the existence of the deficiency in LPLC.

2.3.1 A Deficiency in LPLC

First, we define e_E to denote the difference between the original image and the motion compensated image $\Gamma_E^{(p)}$ given in (2.10)

$$e_E(n) = F(n) - \Gamma_E^{(p)}(n). \tag{2.17}$$

The residue in (2.12) then becomes

$$\psi_E(n) = e_E(n) - \hat{e}_B(n). \tag{2.18}$$

Hence the encoded residue by the enhancement layer in LPLC, $\hat{\psi}_E$, is in fact a quantized version of the *mismatch* between the two PEFs, e_E and $\hat{e}_B$.

Moreover,

$$\begin{aligned}
\psi_E(n) &= (e_E(n) - e_B(n)) + (e_B(n) - \hat{e}_B(n)) \\
&= \left(F(n) - F_B^{(p)}(n) - \alpha\left(F_E^{(p)}(n) - F_B^{(p)}(n)\right) - \left(F(n) - F_B^{(p)}(n)\right)\right) \\
&\quad + \left(F(n) - F_B^{(r)}(n)\right) \\
&= \alpha\left(F_B^{(p)}(n) - F_E^{(p)}(n)\right) + \left(F(n) - F_B^{(r)}(n)\right).
\end{aligned}$$

Thus,

$$\begin{aligned}
\psi_E(n) &= \alpha \mathrm{MC}_{MV(n)}\left\{F_B^{(r)}(n-1) - F_E^{(r)}(n-1)\right\} \\
&\quad + \left(F(n) - F_B^{(r)}(n)\right).
\end{aligned} \tag{2.19}$$

If $\alpha = 0$, (2.19) reduces to (2.5), i.e., the *mismatch* ψ_E becomes the residue between the original image and the reconstruction using the base layer only. Hence, any approximation of ψ_E has additional information regarding the original signal to that provided by the reconstructed base layer $F_B^{(r)}$, and the reconstruction $F_E^{(r)}$ achieved by both layers, as given in (2.14), in the error-free case, should always be superior to $F_B^{(r)}$.

If $\alpha > 0$, however, ψ_E includes another term as indicated in (2.19): the motion compensated image of the difference between the two buffered reconstructions for the reference frame, $F_B^{(r)}(n-1) - F_E^{(r)}(n-1)$, scaled by the leaky factor. The larger the leaky factor, the more significant role this difference plays in the *mismatch* ψ_E. It is possible that the first term in ψ_E dominates the value of the *mismatch*, as a result of a large α (e.g., 1). Thus, a coarsely quantized version of the *mismatch*, carried by the enhancement layer in LPLC, cannot be guaranteed to always provide more information to the decoder than knowledge of the base layer. In other words, it is not guaranteed the enhancement layer in LPLC always "enhances" the reconstructed video quality beyond that provided by the base layer.

We show the existence of the deficiency in LPLC experimentally, with an example given in Table 2.1, 2.2, and 2.3, and Fig. 2.1. The details of the setup for the experiments will be addressed in Section 2.5.1. It is shown in Table 2.1, 2.2 and 2.3 that when $\alpha = 0$, the reconstruction using both layers always achieves superior performance to the one provided using the base layer alone in terms of the video quality, regardless of the data rate allocated to the enhancement layer. When α is

Table 2.1
An example of the deficiency in LPLC - I (LPLC implemented using SAMCoW on QCIF *news* at frame rate 10 fps; Base layer data rate R_B = 60 kbps; R_E denotes the enhancement layer data rate; Average PSNR using the base layer alone PSNR_B = 32.88 dB; PSNR_E denotes the average PSNR using both layers; $\text{PSNR}_{E-B} = \text{PSNR}_E - \text{PSNR}_B$)

R_E (kbps)		2	4	6	8	10
$\alpha = 0$	PSNR_E [dB]	32.89	32.90	32.91	32.92	32.94
	PSNR_{E-B} [dB]	0.01	0.02	0.03	0.04	0.06
$\alpha = 1$	PSNR_E [dB]	32.73	32.19	32.34	32.23	32.15
	PSNR_{E-B} [dB]	-0.15	-0.69	-0.54	-0.65	-0.73
R_E (kbps)		12	14	16	18	20
$\alpha = 0$	PSNR_E [dB]	32.96	33.00	33.04	33.08	33.13
	PSNR_{E-B} [dB]	0.08	0.12	0.16	0.20	0.25
$\alpha = 1$	PSNR_E [dB]	32.10	32.37	32.65	32.75	32.85
	PSNR_{E-B} [dB]	-0.78	-0.51	-0.23	-0.13	-0.03

Table 2.2
An example of the deficiency in LPLC - II (LPLC implemented using SAMCoW on QCIF *news* at frame rate 10 fps; Base layer data rate $R_B = 60$ kbps; R_E denotes the enhancement layer data rate; Average PSNR using the base layer alone $\text{PSNR}_B = 32.88$ dB; PSNR_E denotes the average PSNR using both layers; $\text{PSNR}_{E-B} = \text{PSNR}_E - \text{PSNR}_B$)

R_E (kbps)		22	24	26	28	30
$\alpha = 0$	PSNR_E [dB]	33.18	33.23	33.29	33.34	33.40
	PSNR_{E-B} [dB]	0.30	0.35	0.41	0.46	0.52
$\alpha = 1$	PSNR_E [dB]	33.02	33.10	33.21	33.32	33.41
	PSNR_{E-B} [dB]	0.14	0.22	0.33	0.44	0.53
R_E (kbps)		32	34	36	38	40
$\alpha = 0$	PSNR_E [dB]	33.45	33.50	33.55	33.60	33.66
	PSNR_{E-B} [dB]	0.57	0.62	0.67	0.72	0.78
$\alpha = 1$	PSNR_E [dB]	33.51	33.59	33.67	33.74	33.83
	PSNR_{E-B} [dB]	0.63	0.71	0.79	0.86	0.95

Table 2.3
An example of the deficiency in LPLC - III (LPLC implemented using SAMCoW on QCIF *news* at frame rate 10 fps; Base layer data rate $R_B = 60$ kbps; R_E denotes the enhancement layer data rate; Average PSNR using the base layer alone $\text{PSNR}_B = 32.88$ dB; PSNR_E denotes the average PSNR using both layers; $\text{PSNR}_{E-B} = \text{PSNR}_E - \text{PSNR}_B$)

R_E (kbps)		42	44	46	48	50
$\alpha = 0$	PSNR_E [dB]	33.71	33.76	33.81	33.86	33.92
	PSNR_{E-B} [dB]	0.83	0.88	0.93	0.98	1.04
$\alpha = 1$	PSNR_E [dB]	33.91	33.96	34.01	34.06	34.18
	PSNR_{E-B} [dB]	1.03	1.08	1.13	1.18	1.30
R_E (kbps)		52	54	56	58	60
$\alpha = 0$	PSNR_E [dB]	33.98	34.03	34.09	34.15	34.21
	PSNR_{E-B} [dB]	1.10	1.15	1.21	1.27	1.33
$\alpha = 1$	PSNR_E [dB]	34.25	34.46	34.50	34.62	34.73
	PSNR_{E-B} [dB]	1.37	1.58	1.62	1.74	1.85

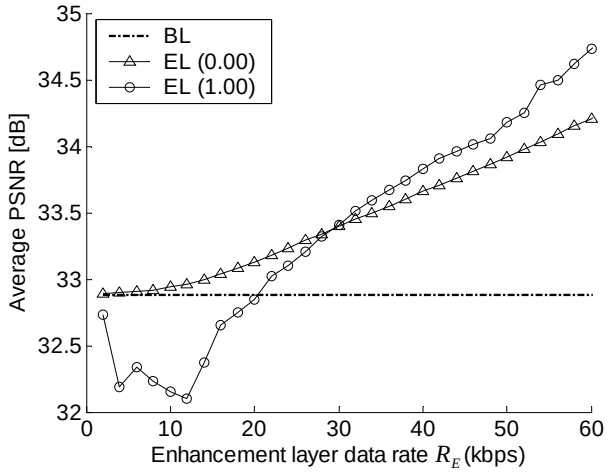


Fig. 2.1. An example of the deficiency in LPLC (LPLC implemented using SAMCoW on QCIF *news* at leaky factors $\alpha = 0$ and $\alpha = 1$; BL: reconstruction using the base layer alone; EL: reconstruction using both layers)

as large as 1, however, a low data rate allocation such as 12 kbps results in worse performance for the reconstruction obtained using both layers than that obtained using the base layer alone by as much as 0.78 dB in PSNR.

We may also interpret the deficiency in LPLC from another point of view. We know that the essential idea of a layered coding is to convey a coarse approximation of a given video signal to the decoder in the lower layer(s) (the base layer), and then encode the residual information between the original signal and the coarse approximation in the higher layer(s) (the enhancement layer) to refine the approximation. In the conventional layered coding structure, the enhancement layer carries information regarding $e_B - \hat{e}_B$ as given in (2.5), a residue between one PEF and its own approximation. Instead, LPLC carries information regarding $e_E - \hat{e}_B$ as given in (2.18), a *mismatch* between one PEF and the approximation of another PEF. Generally, it is believed that $\hat{e}_B$, which is an approximation of e_B, is also a good approximation of e_E. This is based on the fact that both e_B and e_E are the residues between the original signal and a motion compensated image of a reconstruction of the same reference frame using the same motion vectors. Hence, large redundancy should exist between $\hat{e}_B$ and e_E. Nevertheless, the amount of redundancy is closely related to how similar, or how different from each other, the two respective reconstructions exploited by the base layer and by the enhancement layer for motion compensation are. In LPLC, as given in (2.9), the larger the leaky factor, the more dominant $F_E^{(r)}$ is in $\Gamma_E^{(r)}$, and thus the more different the two reconstructions, $F_B^{(r)}$ and $\Gamma_E^{(r)}$, are from each other. If no enough redundancy exists between e_E and $\hat{e}_B$,

it is not guaranteed that encoding $e_E - \hat{e}_B$ will always be an effective way to convey finer information about the original signal in addition to $\hat{e}_B$.

In [20], a high-quality base layer scheme is proposed. Instead of $F_B^{(r)}$, the coder uses the following reconstruction by using a second leaky prediction to encode the base layer

$$\Gamma_B^{(r)}(n) \triangleq F_B^{(r)}(n) + \alpha_B \left(F_E^{(r,b)}(n) - F_B^{(r)}(n) \right), \tag{2.20}$$

where α_B denotes a second leaky factor, satisfying $0 \leqslant \alpha_B \leqslant \alpha \leqslant 1$, and $F_E^{(r,b)}(n)$ is a second reconstruction using both layers at the encoder side, which is obtained in a similar way as $F_E^{(r)}(n)$ in (2.14) except that $\Gamma_B^{(p)}(n)$, the motion compensated image of $\Gamma_B^{(r)}(n-1)$, and a different quantized version of $\psi_E(n)$ are used. FGS is utilized in [20] to encode the enhancement layer, and $F_E^{(r,b)}$ has inferior quality to $F_E^{(r)}$ since it takes less bitplanes when reconstructing the *mismatch* ψ_E. In [20] the above scheme is used for the sake of coding efficiency, since it reduces the dynamic range of the *mismatch* ψ_E.

We show that the above scheme also alleviates the deficiency in LPLC to some extent. With a better quality reconstruction used for the base layer, the difference between the two reconstructions using the two layers is mitigated. From (2.19), it is seen that decreasing $F_B^{(r)}(n-1) - F_E^{(r)}(n-1)$ diminishes the effect of the first term but emphasizes the second term, $F(n) - F_B^{(r)}(n)$, in the *mismatch* $\psi_E(n)$. Meanwhile, as discussed in our second point of view, decreasing $F_B^{(r)} - F_E^{(r)}$ results in larger redundancy between $\hat{e}_B$ and e_E. Therefore, from either point of view in our analysis,

decreasing $F_B^{(r)} - F_E^{(r)}$ makes it more likely that the reconstruction produced by both layers is superior in quality to that offered by the base layer alone.

2.3.2 Similarity between LPLC and an MDC Scheme

In this section, we show the similarity between LPLC and MDMC, a multiple description video coding approach using multiple description motion compensation. MDMC partitions the original video sequence into two descriptions, similar to the video redundancy coding (VRC) scheme presented in [55].[4] MDMC inserts the coded bitstream for the even frames in one description, Description I, and includes the odd frames in the other description, namely Description II. Unlike VRC where each description is independently coded except that a sync frame is periodically inserted, MDMC exploits a second-order predictor and uses three loops for motion compensation: one central loop and two side loops. Each description obtains its bitstream from the central loop as well as its own side loop.

For a given frame $F(n)$, MDMC generates two sets of motion vectors from the previous two frames $F(n-1)$ and $F(n-2)$, namely $MV1(n)$ and $MV2(n)$, and obtains the following two motion compensated images

$$F_1^{(p)}(n) = \mathrm{MC}_{MV1(n)}\left\{F_C^{(r)}(n-1)\right\} \tag{2.21}$$

$$F_2^{(p)}(n) = \mathrm{MC}_{MV2(n)}\left\{F_C^{(r)}(n-2)\right\}, \tag{2.22}$$

[4] VRC was included in the ITU-T standard H.263+ [56].

where $F_C^{(r)}(n)$ denotes the buffered reconstruction of the nth frame obtained from the central loop. Using $F_1^{(p)}$ and $F_2^{(p)}$, the central loop in MDMC obtains a PEF, denoted as e_C, using a second-order prediction

$$e_C(n) = F(n) - \alpha F_1^{(p)}(n) - (1-\alpha)F_2^{(p)}(n), \tag{2.23}$$

where $\alpha \in [0,1]$ denotes the parameter used in the prediction. MDMC then encodes the quantized version

$$\hat{e}_C(n) = \text{Quant}\{e_C(n)\}. \tag{2.24}$$

If n is even, MDMC places the encoded $\hat{e}_C$ in the bitstream of Description I together with the two sets of coded motion vectors, $MV1$ and $MV2$; otherwise MDMC places it along with the motion vectors in Description II. The reconstruction within the central loop, $F_C^{(r)}$, is then obtained as

$$F_C^{(r)}(n) = \alpha F_1^{(p)}(n) + (1-\alpha)F_2^{(p)}(n) + \hat{e}_C(n). \tag{2.25}$$

In order to handle the case where only one description is received at the decoder, MDMC includes data generated by a side motion compensation loop for each description. If n is even, for instance, MDMC obtains the PEF, denoted as e_S, by using the motion compensated image belonging to its own description, in this case Description I, as

$$e_S(n) = F(n) - F_2^{(p)}(n). \tag{2.26}$$

MDMC then obtains the following *mismatch*

$$\psi_S(n) = e_S(n) - \hat{e}_C(n), \tag{2.27}$$

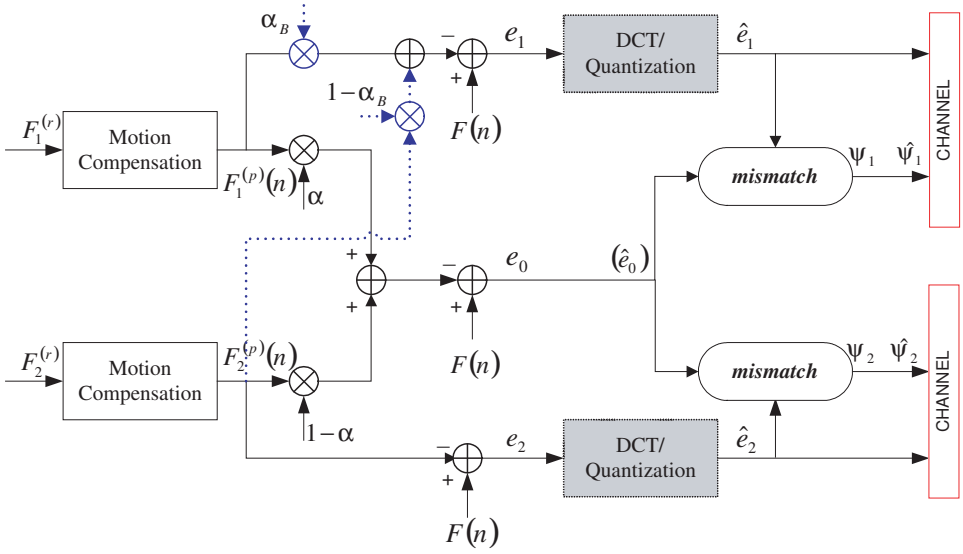


Fig. 2.2. A general framework for LPLC and MDMC

and adds its quantized version, $\hat{\psi}_S$, to the same description, namely Description I, where

$$\hat{\psi}_S(n) = \text{Quant}\{\psi_S(n)\}. \tag{2.28}$$

Thus the reconstruction, $F_S^{(r)}$, that uses the side loop is

$$F_S^{(r)}(n) = F_2^{(p)}(n) + \hat{e}_C(n) + \hat{\psi}_S(n). \tag{2.29}$$

It is to be noted that even though MDMC does not use $F_C^{(r)}(n-1)$, which belongs to the other description, to encode the nth frame in the side loop, yet it needs $F_C^{(r)}(n-2)$, which requires $F_C^{(r)}(n-3)$, a reconstruction from the other description (see equations (2.21), (2.22), and (2.25)). Hence, when there is only one description available at the decoder, MDMC has to estimate the other description even when just using the side loop, which always results in error drift.

Rewriting (2.23) as

$$e_C(n) = F(n) - F_2^{(p)}(n) - \alpha \left(F_1^{(p)}(n) - F_2^{(p)}(n) \right), \tag{2.30}$$

we cast the second order prediction utilized in MDMC in a leaky prediction framework. Using this alternative view point, it turns out that MDMC in fact uses leaky prediction in the central loop. In this case, the motion compensated image $F_1^{(p)}(n)$, belonging to the other description, is scaled by the leaky factor α in a similar way to which the motion predicted enhancement layer $F_E^{(p)}(n)$ in LPLC, given in (2.10) and (2.12), is scaled. When $\alpha = 0$, there is no contribution from $F_1^{(p)}(n)$, and $e_C(n)$ becomes $e_S(n)$, the PEF in the side loop.

Based on the above discussion, we propose a general framework, given in Fig. 2.2, which applies to both LPLC and MDMC. In this framework, if we let $F_1^{(r)} = F_E^{(r)}(n-1)$ and $F_2^{(r)} = F_B^{(r)}(n-1)$, the bottom solid path becomes the base layer motion compensation path used in LPLC, and the middle solid path becomes the enhancement layer motion compensation path used in LPLC. The top solid path (using the dotted line) is the motion compensation path that generates the second leaky prediction for the high-quality base layer scheme as indicated in (2.20). Also, if we let $F_1^{(r)} = F_C^{(r)}(n-1)$ and $F_2^{(r)} = F_C^{(r)}(n-2)$, and force $\alpha_B = 1$, our framework coincides exactly with MDMC: the middle path corresponds to the central loop while the top and bottom paths correspond to the two side loops.

Therefore, the framework given in Fig. 2.2 presents a significant similarity between LPLC and MDMC. The differences, however, between the two coding structures are: (1) From the encoder's point of view, the two reconstructions in LPLC

are two reconstructions derived from the same frame with different qualities. While in the case of MDMC, the two reconstructions are derived from different frames and thus include "non-overlapping" information. (2) From the decoder's point of view, LPLC is only concerned with whether the base layer or both layers are available, since the enhancement layer becomes useless without the base layer due to the nested scalability structure. As for MDMC, the decoder needs to consider three possibilities, namely the availability of Description I, or Description II, or both.

As described by the middle path in the framework of Fig. 2.2, the central loop of MDMC is analogous to the enhancement layer MC loop in LPLC. Both loops use a linear combination of two predictions for the current frame, or from another perspective, they both use a leaky factor to combine the two predictions. To the nth frame $F(n)$, $F_1^{(p)}(n)$ in MDMC, given by (2.21), is generally a better prediction compared to $F_2^{(p)}(n)$ by (2.22), since adjacent frames are usually considered to be more correlated than frames that are two frames away. The central loop in MDMC thus results in a reconstruction with superior quality compared to that obtained by the side loop, for it uses a leaky prediction that incorporates $F_1^{(p)}(n)$ in its motion compensation while $F_1^{(p)}(n)$ is excluded from the side loop. Analogously, the prediction $F_E^{(p)}(n)$ in LPLC by (2.11) is a better prediction of the nth frame, since it includes refined information beyond $F_B^{(p)}(n)$ by (2.4). Hence the enhancement layer MC loop in LPLC should generate a reconstruction with superior quality compared to that obtained using the base layer alone, for it uses a leaky prediction that incorporates $F_E^{(p)}(n)$ while $F_E^{(p)}(n)$ is not considered in the base layer MC loop. We

refer to this similarity between the central loop in MDMC and the enhancement layer in LPLC, considered from the leaky prediction point of view, as Similarity I. Similarity I conforms with the well-accepted supposition in the literature that the reconstructed quality using both layers in LPLC shall be superior to that using the base layer alone.

Nevertheless, another type of similarity exists between MDMC and LPLC. We have identified a *mismatch* as the difference between the PEF in one loop/layer and the quantized PEF in the other loop/layer. In MDMC, the *mismatch* ψ_S is obtained as (2.27), and it is the side loop that conveys the quantized version of the *mismatch*. $\hat{\psi}_S$ is coarsely quantized in order to maintain the introduced redundancy by the side loops at a low level. When both descriptions are available, the decoder always favors the image reconstructed in the central loop, $F_C^{(r)}$, and discards $F_S^{(r)}$ in the side loops. Similarly in LPLC, the *mismatch* ψ_E is obtained as (2.18), and it is the enhancement layer that carries the *mismatch*. We refer to the similarity between the side loops in MDMC and the enhancement layer in LPLC, considered from the *mismatch* point of view, as Similarity II. Due to Similarity II between MDMC and LPLC, the inferior reconstructed quality of $F_S^{(r)}$ by the side loop compared to $F_C^{(r)}$ in MDMC implies a possible inferior performance of $F_E^{(r)}$ by both the enhancement layer and the base layer compared to $F_B^{(r)}$ in LPLC, when the *mismatch* $\hat{\psi}_E$ carried by the enhancement layer in LPLC is coarsely quantized in a manner similar to $\hat{\psi}_S$. Thus, Similarity II between MDMC and LPLC confirms the deficiency that might exist in LPLC.

The seemingly disagreement between the above two types of similarities is consistent with our analysis that the superiority of the enhancement layer in LPLC is dependent on the leaky factor as well as the accuracy of the encoded enhancement layer itself.

2.4 ML Estimation Enhanced LPLC

Exploiting the similarity between LPLC and MDMC, we use maximum likelihood (ML) estimation that was originally developed for an MDMC-like multiple description video coding scheme [57] to enhance the rate distortion performance of LPLC. We term this approach ML-LPLC. For a given video sequence with frame size of $M \times N$ pixels, let $\rho_B^{(r)}(x,y)$, $\rho_E^{(r)}(x,y)$, and $\rho_{ML}^{(r)}(x,y)$ denote the pixel value at the location (x,y) in $F_B^{(r)}$, $F_E^{(r)}$, and $F_{ML}^{(r)}$ respectively, where $F_{ML}^{(r)}$ is the image reconstructed using ML estimation. Thus,

$$\begin{aligned} F_B^{(r)} &= \left\{\rho_B^{(r)}(x,y)\right\}_{(x,y)=(0,0)}^{(M-1,N-1)} \\ F_E^{(r)} &= \left\{\rho_E^{(r)}(x,y)\right\}_{(x,y)=(0,0)}^{(M-1,N-1)} \\ F_{ML}^{(r)} &= \left\{\rho_{ML}^{(r)}(x,y)\right\}_{(x,y)=(0,0)}^{(M-1,N-1)} . \end{aligned}$$

Assuming the quantization noise in each pixel to be an independent, identically distributed (i.i.d.) zero-mean Gaussian random variable, ML-LPLC obtains the reconstruction $F_{ML}^{(r)}$ when both layers are available at the decoder as follows

$$\begin{aligned}
\rho_{ML}^{(r)}(x,y) &= \left(\begin{bmatrix} 1 & 1 \end{bmatrix} \Sigma^{-1} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 & 1 \end{bmatrix} \Sigma^{-1} \begin{bmatrix} \rho_B^{(r)}(x,y) \\ \rho_E^{(r)}(x,y) \end{bmatrix} \\
&\triangleq \begin{bmatrix} \pi & 1-\pi \end{bmatrix} \begin{bmatrix} \rho_B^{(r)}(x,y) \\ \rho_E^{(r)}(x,y) \end{bmatrix} \\
&= \pi \rho_B^{(r)}(x,y) + (1-\pi)\rho_E^{(r)}(x,y), \qquad (2.31)
\end{aligned}$$

where Σ is the cross-correlation matrix between the two reconstruction errors, $\rho - \rho_B^{(r)}$ and $\rho - \rho_E^{(r)}$,

$$\begin{aligned}
\Sigma &= E\left(\begin{bmatrix} \rho - \rho_B^{(r)} \\ \rho - \rho_E^{(r)} \end{bmatrix} \begin{bmatrix} \rho - \rho_B^{(r)} & \rho - \rho_E^{(r)} \end{bmatrix} \right) \\
&= \begin{bmatrix} E\left[\left(\rho - \rho_B^{(r)} \right)^2 \right] & E\left[\left(\rho - \rho_B^{(r)} \right) \left(\rho - \rho_E^{(r)} \right) \right] \\ E\left[\left(\rho - \rho_B^{(r)} \right) \left(\rho - \rho_E^{(r)} \right) \right] & E\left[\left(\rho - \rho_E^{(r)} \right)^2 \right] \end{bmatrix} \\
&\triangleq \begin{bmatrix} a & b \\ b & d \end{bmatrix}, \qquad (2.32)
\end{aligned}$$

where $\rho(x,y)$ denotes the original pixel value at location (x,y), and $E\{\cdot\}$ denotes the expectation of a random variable or random vector.[5] We use empirical averages to approximate the expectations in (2.32). For instance,

$$E\left[(\rho - \hat{\rho}_B)^2 \right] \cong \frac{1}{M \times N} \sum_{(x,y)=(0,0)}^{(M-1,N-1)} \left(\rho(x,y) - \rho_B^{(r)}(x,y) \right)^2. \qquad (2.33)$$

[5]Interested readers may refer to [57] for the detailed derivation of (2.35). The ML approach we proposed here is very similar to that presented in [57].

Combining (2.31) and (2.32), we have

$$\pi = \frac{d-b}{a+d-2b}, \tag{2.34}$$

and we refer to π as the ML coefficient hereafter. ML-LPLC necessitates that we transmit the ML coefficient(s) associated with each frame as the side information to ensure that the ML estimate of each frame $F_{ML}^{(r)}$ can be obtained at the decoder. For color video sequences, we transmit three ML coefficients for each frame, one for the luminance component and two for the two chrominance components. Each ML coefficient is obtained as shown above. With the ML coefficient $\pi(n)$ calculated for the nth frame, ML-LPLC obtains the reconstruction $F_{ML}^{(r)}(n)$ as

$$F_{ML}^{(r)}(n) = \pi(n)F_{B}^{(r)}(n) + (1-\pi(n))F_{E}^{(r)}(n), \tag{2.35}$$

where $F_{B}^{(r)}(n)$ and $F_{E}^{(r)}(n)$ are the two reconstructions in LPLC.

It is observed that the ML coefficients include information from the original video signal, and are thus helpful to the decoder in obtaining better reconstruction from the two layers. Our experimental results in Section 2.5 demonstrate that the video quality of $F_{ML}^{(r)}$ achieved by ML-LPLC is always superior to or at least as good as the better of the two reconstructions using the two layers when both layers are available at the decoder, demonstrating that ML-LPLC is capable of overcoming the deficiency in LPLC.

It is also noted that MDMC can obtain three reconstructions for each frame at the decoder when both descriptions are available: the two reconstructions directly obtained in the central loop and the side loop the decoded frame belongs to, and a

third one estimated from the other side loop [53]. Exploiting the similarity between LPLC and MDMC, we can obtain another reconstruction in a similar way as in MDMC when $0 < \alpha \leq 1$ given by [51]

$$F_{est}^{(r)}(n) = \mathrm{MC}_{MV(n+1)}^{-1} \left\{ F_B^{(p)}(n+1) - \frac{\hat{\psi}_E(n+1)}{\alpha} \right\}, \tag{2.36}$$

where $\mathrm{MC}_{MV(n+1)}^{-1}\{\cdot\}$ denotes the backward motion compensation operation using the forward motion vector $MV(n+1)$. The estimated reconstruction $F_{est}^{(r)}$ is desirable for video transmission over error-prone networks, since the enhancement layer for the current frame that carries $\hat{\psi}_E(n)$ may be destroyed or lost while the following enhancement layer information $\hat{\psi}_E(n+1)$ might be available. However, two disadvantages are associated with $F_{est}^{(r)}(n)$: (1) The denominator $0 < \alpha < 1$ amplifies the quantization error introduced in $\hat{\psi}_E(n+1)$, and (2) the implementation of backward motion compensation based on forward motion vectors in fractional resolution is still an open problem.[6]

2.5 Experimental Results

As discussed in Section 2.3.1, the superiority of the enhancement layer in LPLC cannot always be guaranteed, but is closely related to the leaky factor and the allocated data rate to the enhancement layer. In this section, we examine the performance of LPLC and our proposed ML-LPLC as a function of the leaky factor and the enhancement layer data rates.

[6]The same problem also exists for MDMC. When the motion vectors are in fractional resolution, estimating one description from the other is not trivial. The MDMC described in [53] only discusses the case where integer motion vectors are used.

We implement LPLC and ML-LPLC using two video codecs, SAMCoW [5] and ITU-T H.26L version TML9.4 [54], with detailed descriptions given in the following two subsections. The video sequences for evaluation of both LPLC and ML-LPLC include *foreman* (400 frames), *news* (300 frames), *mother-daughter* (400 frames), all in QCIF 4:2:0 YUV format, and *foreman* (300 frames), *bus* (150 frames), *news* (300 frames), and *akiyo* (300 frames), all in CIF 4:2:0 YUV format. Note that these video sequences contain varying degrees of motions.

In our experiments, we intra-coded the first frame of each sequence and inter-coded all successive frames. The type of bidirectional frame, i.e., B frame, is not considered. We chose one reference frame for motion estimation and compensation. We used PSNR as the metric for evaluating the decoded video quality. We encoded all ML coefficients using 6-bits, which is a very acceptable redundancy associated with each frame (18-bits if we inlcude chrominance components). A sample of ML coefficients is given in Fig. 2.3. It is observed that the ML coefficients do not change much from frame to frame, implying that a more efficient way to encode ML coefficients is possible.

2.5.1 Experimental Results from Implementation Using SAMCoW

We first used a wavelet based fully rate scalable hybrid video codec, namely SAMCoW [5], to implement LPLC and ML-LPLC. We evaluated their operational rate distortion performance associated with various leaky factors for different video sequences.

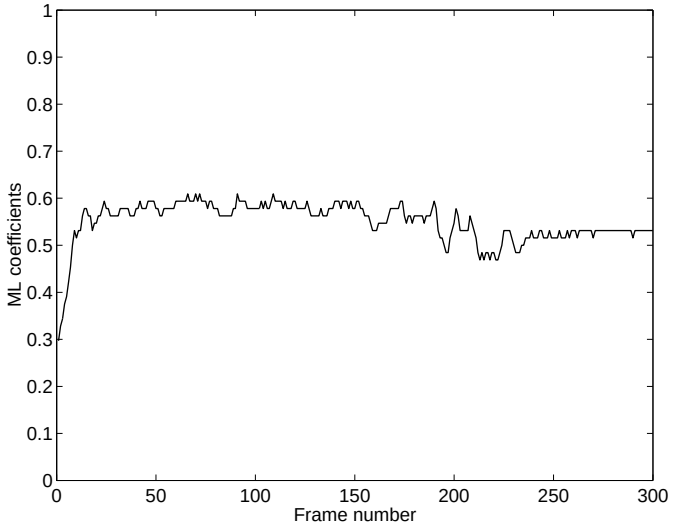
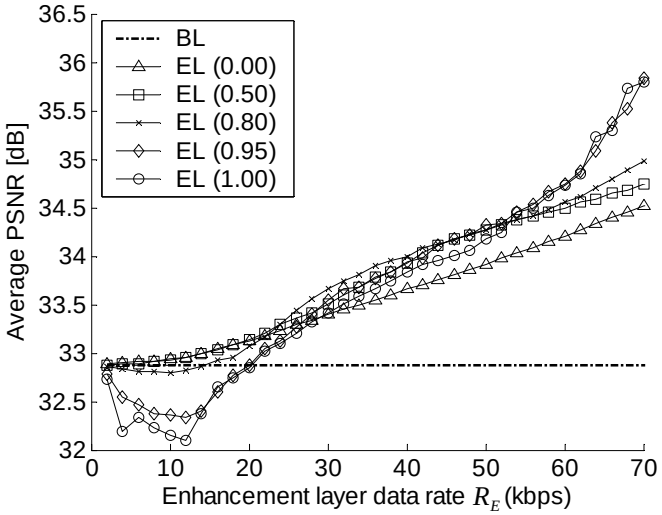


Fig. 2.3. An example of the ML coefficients $\pi(n)$ for the luminance component (ML-LPLC implemented using ITU-T H.26L TML9.4 on CIF *foreman*)

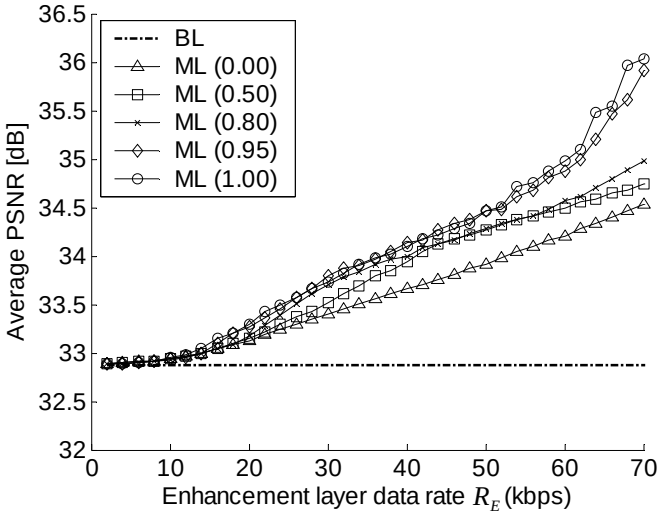
SAMCoW uses motion estimation/compensation to reduce the temporal redundancy and an approach similar to the embedded zero tree (EZW) algorithm to encode the intra frames and the PEFs. SAMCoW falls into the category of the conventional layered coding structure since only one MC loop is implemented and any embedded bitstream beyond the base layer is excluded from the MC loop. If R_B denotes the data rate allocated to the base layer, and R_T denotes the encoding data rate for the overall bitstream, SAMCoW allows drift-free decoding at any data rate between R_B and R_T if no error or truncation occurs to the bitstream.

In our LPLC implementation using SAMCoW, the data rate allocated to the base layer is R_B, and the data rate to the enhancement layer is R_E. For inter frames, the base layer contains the embedded bitstream of the motion vectors at a data rate of R_{MV} and the embedded bitstream of the PEFs at a data rate of $R_B - R_{MV}$. The reconstruction of the PEF using the base layer is $\hat{e}_B$, as in (2.2). The enhancement layer carries the embedded bitstream of the *mismatch* at a data rate of R_E and a reconstruction of the *mismatch* $\hat{\psi}_E$ is attained, as in (2.13). For the intra frame, both layers contain the embedded bitstream of the original frame at the respective data rate requirements, where the base layer includes the more significant bit planes and the enhancement layer carries the refinement bit planes.

For an arbitrary video sequence and a predefined leaky factor α, we fixed the base layer data rate R_B and varied the data rate for the enhancement layer R_E within a dynamic range. We encoded each video sequence at a frame rate 10 fps.

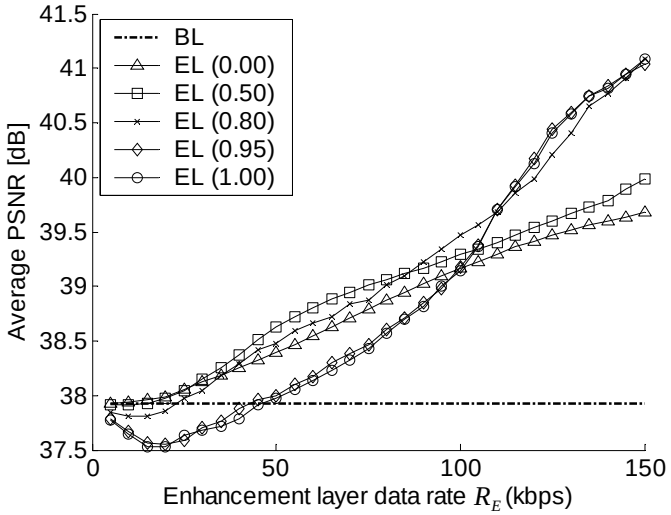


(a) *news* by LPLC (QCIF, $R_B = 60$ kbps)

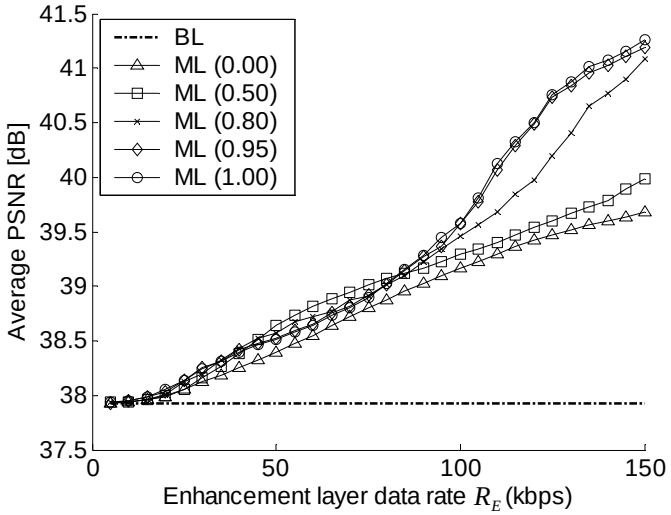


(b) *news* by ML-LPLC (QCIF, $R_B = 60$ kbps)

Fig. 2.4. Comparison of the performance of LPLC and ML-LPLC on *news* at various leaky factors (obtained from the implementation using SAMCoW; BL: reconstruction using the base layer alone; EL: reconstruction using both layers; ML: reconstruction using ML-LPLC)

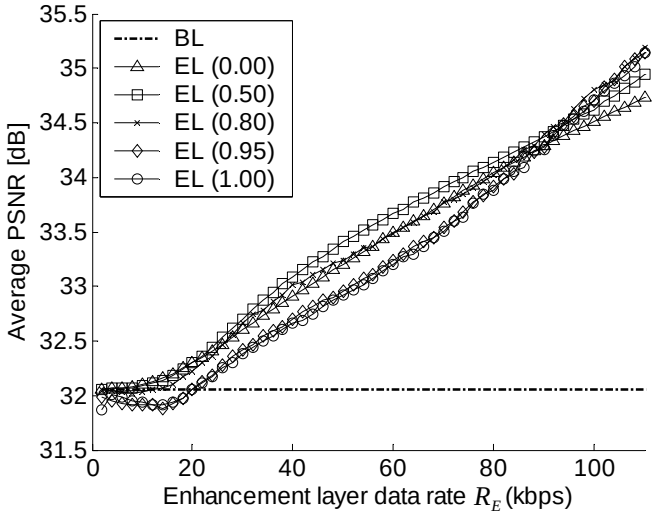


(a) *akiyo* by LPLC (CIF, $R_B = 80$ kbps)

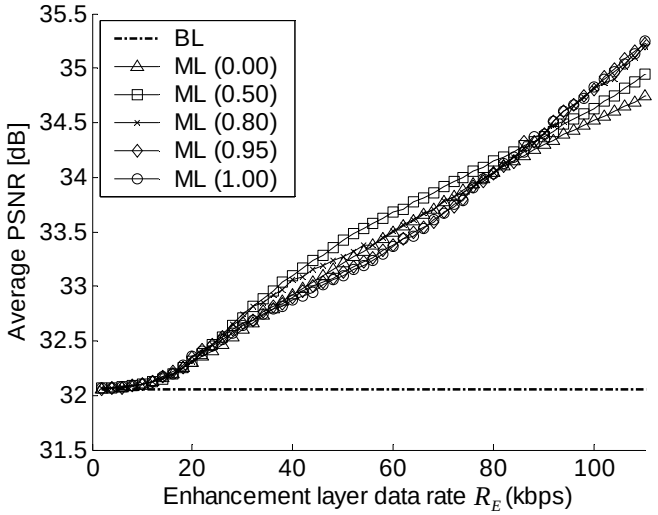


(b) *akiyo* by ML-LPLC (CIF, $R_B = 80$ kbps)

Fig. 2.5. Comparison of the performance of LPLC and ML-LPLC on *akiyo* at various leaky factors (obtained from the implementation using SAMCoW; BL: reconstruction using the base layer alone; EL: reconstruction using both layers; ML: reconstruction using ML-LPLC)

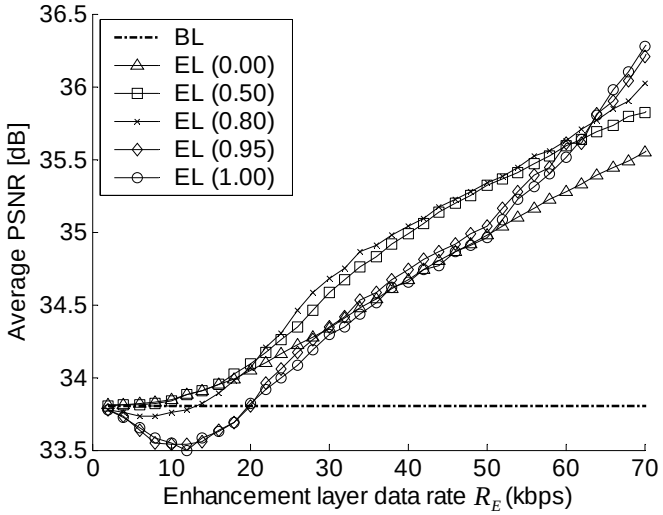


(a) *foreman* by LPLC (QCIF, $R_B = 70$ kbps)

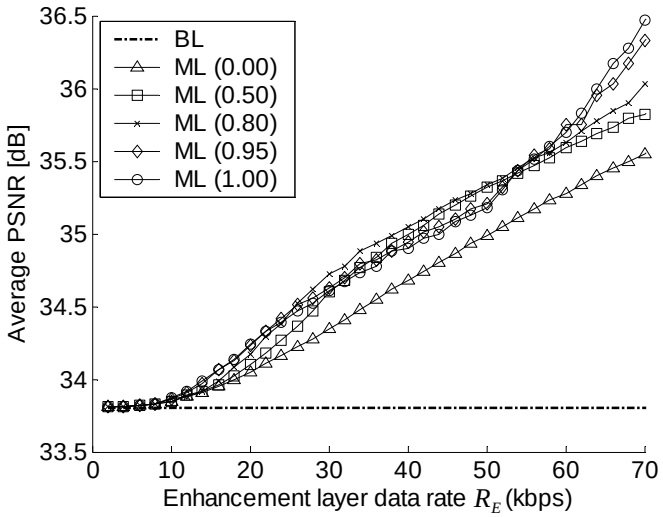


(b) *foreman* by ML-LPLC (QCIF, $R_B = 70$ kbps)

Fig. 2.6. Comparison of the performance of LPLC and ML-LPLC on *foreman* at various leaky factors (obtained from the implementation using SAMCoW; BL: reconstruction using the base layer alone; EL: reconstruction using both layers; ML: reconstruction using ML-LPLC)



(a) *mother-daughter* by LPLC (QCIF, $R_B = 40$ kbps)



(b) *mother-daughter* by ML-LPLC (QCIF, $R_B = 40$ kbps)

Fig. 2.7. Comparison of the performance of LPLC and ML-LPLC on *mother-daughter* at various leaky factors (obtained from the implementation using SAMCoW; BL: reconstruction using the base layer alone; EL: reconstruction using both layers; ML: reconstruction using ML-LPLC)

Examples of the operational rate distortion performance of LPLC using SAM-CoW are given in Fig. 2.4(a), 2.5(a), 2.6(a), and 2.7(a), where the leaky factor took on five values: 0, 0.5, 0.8, 0.95, and 1. It is observed that the decoded quality by both layers in LPLC, denoted as "EL" in the figure, is closely correlated with the leaky factors. We may heuristically partition the data rates allocated to the enhancement layer, R_E, into three segments, referred to hereafter as *high-rate*, *medium-rate*, and *low-rate*, each characterizing a specific rate distortion performance with respect to the leaky factors.

The *high-rate* segment is defined in a manner where the data rate allocated to the enhancement layer in LPLC, R_E, is sufficiently large such that the decoded quality by both layers monotonically increases with the increase of R_E and is always superior to that obtained using the base layer alone, regardless of the leaky factors. Moreover, at a certain data rate, larger leaky factors always result in superior decoded qualities, implying that the inclusion of a larger portion of the enhancement layer in the MC loop always gains an improvement in coding efficiency. For example, the enhancement layer data rate is considered as *high-rate* when $R_E \geq 54$ kbps for QCIF *news* as in Fig. 2.4(a) while when $R_E \geq 110$ kbps for CIF *akiyo* as in Fig. 2.5(a). When R_E belongs to the *high-rate* segment, the optimal leaky factor is $\alpha = 1$ in terms of the decoded video quality at a certain data rate, or equivalently, is optimal in terms of the coding efficiency, while $\alpha = 0$ results in the worst performance.

The *low-rate* segment is referred to the enhancement layer data rate range within which there exists at least one leaky factor resulting in the decoded quality using both

layers inferior to that provided by the base layer alone. Consistent with our analysis in Section 2.3.1, it is observed that larger leaky factors have higher probability than smaller ones resulting in the deficiency of LPLC when the enhancement layer data rate belongs to the *low-rate* segment. For example, for QCIF *news*, when $\alpha = 1$ and $R_E = 12$ kbps, the decoded quality using both layers results in a 0.8 dB loss in PSNR compared to that using the base layer alone, as shown in 2.4(a). In this case, the enhancement layer that consumes additional data rate beyond the base layer does not "enhance" the reconstructed video quality, instead, it degrades what has been achieved by the base layer. We refer to this phenomenon that characterizes the deficiency in LPLC as the *low-rate* phenomenon.

The *medium-rate* segment is located between the above two segments, where the enhancement layer does "enhance" the decoded video quality, but the optimal leaky factor in terms of the rate distortion performance varies across different data rates and different video sequences. For instance, for QCIF *news* and QCIF *mother-daughter*, $\alpha = 0.8$ obtains the best performance over a majority of their *medium-rate* segments, while for CIF *akiyo* and QCIF *foreman*, $\alpha = 0.5$ dominates all other leaky factor choices. This *medium-rate* phenomenon was explicitly addressed in a work presented in [24].

As observed in Fig. 2.4, 2.5, 2.6, and 2.7, at a certain data rate that belongs to either the *low-rate* or the *medium-rate* segments, there is no monotone trend in the decoded quality using both layers with respect to the leaky factor. This is because

the *mismatch* $\psi_E(n)$ for frame n given in (2.19) has $F_E^{(r)}(n-1)$ in its first term, which recursively relates with the leaky factor α as given in (2.10) and (2.14).

We obtained similar results for all other video sequences. Note that the definitions of "high" or "low" data rate are relative terms. The deficiency of LPLC, as we discussed in Section 2.3.1, manifests itself in the *low-rate* segment, especially for leaky factors as large as or close to 1. When $\alpha = 0$, i.e., when LPLC reduces to the conventional layered coding structure, the enhancement layer always fulfills its duty to "enhance" the performance. It is interesting to observe that when $\alpha = 0$, the decoded quality preserves an approximately constant increasing rate with respect to the enhancement layer data rate. For instance, for QCIF *news*, every increment of 10 kbps in data rate approximately results in a gain of 0.25 dB in average PSNR. This observation is consistent with the theoretical analysis of the rate distortion performance of the conventional layered scalable coding structure [58–60]. Interested readers could refer to the references for more details.

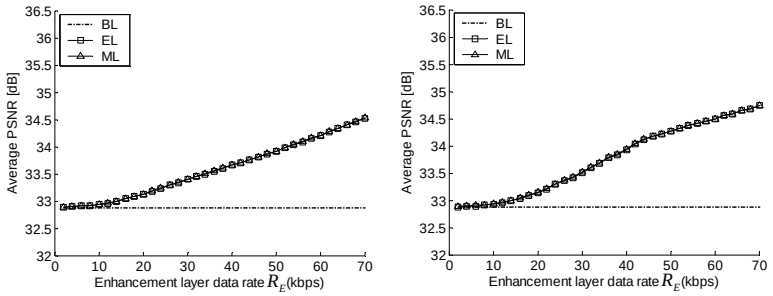
The operational rate distortion performance of our ML-LPLC approach is presented in Fig. 2.4(b), 2.5(b), 2.6(b), and 2.7(b), with parameters identical to those in LPLC except that ML-LPLC is used to reconstruct videos from both layers, denoted as "ML" in the figure. It is observed that regardless of the leaky factor, ML-LPLC moves the entire operational rate distortion curve above that obtained by the base layer and the *low-rate* segment disappears. The reconstruction quality obtained using ML-LPLC is always superior to that using the base layer alone, implying that ML-LPLC effectively addresses the deficiency in LPLC and allows the enhancement

layer always "enhance" the performance. Note that the *medium-rate* segment still exists in ML-LPLC. However, it is observed that this segment is either condensed, or within the segment, the differentiation across different leaky factors is diminished in terms of the operational rate distortion performance.

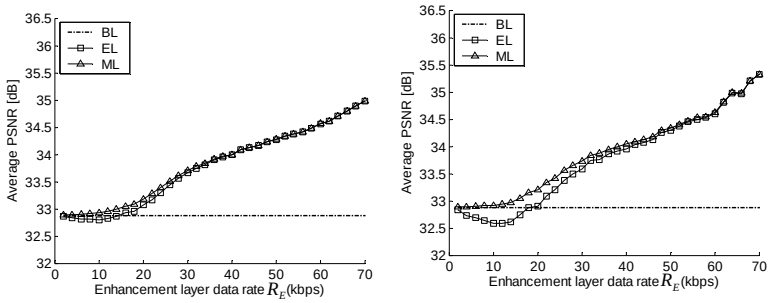
Furthermore, we compare the performance of ML-LPLC and LPLC at each predefined leaky factor, with an example given in Fig. 2.8, where six leaky factors are selected: 0, 0.5, 0.8, 0.9, 0.95, and 1. It is shown that ML-LPLC always obtains a reconstruction as good as or superior in decoded quality to the better of the reconstructions using both layers or using the base layer alone. When the leaky factor approaches 1, ML-LPLC can achieve a gain of up to 0.5 dB in average PSNR compared to LPLC using both layers, even when the operational rate distortion curve of LPLC obtained using both layers is well above that using the base layer alone.

2.5.2 Experimental Results from Implementation Using H.26L

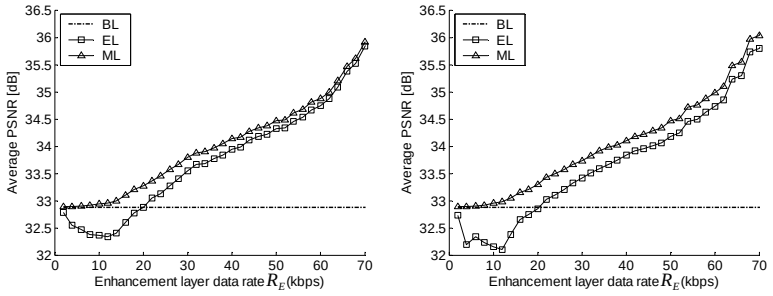
We also implemented LPLC and ML-LPLC by modifying the ITU-T H.26L version TML9.4 [54]. Compared to the previous video coding standards, such as MPEG-4 and H.263+, H.26L contains more features in its video coding layer (VCL) that further improve the coding efficiency at all data rates, and fulfill several tasks in the network abstraction layer (NAL) to improve the error resilience performance of the bitstream. H.26L uses seven block shapes for motion prediction, obtains motion vectors up to 1/4 or 1/8 pel resolution, and utilizes multiple reference frames to implement multihypothesis motion prediction for each frame. In all of our experiments,



(a) $\alpha = 0$ (b) $\alpha = 0.5$



(c) $\alpha = 0.8$ (d) $\alpha = 0.9$



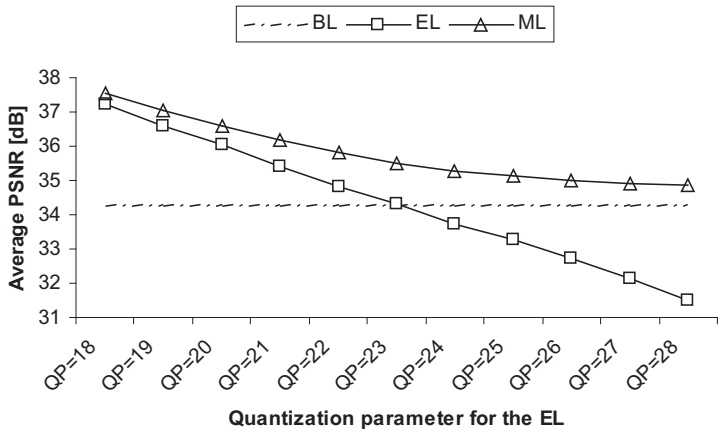
(e) $\alpha = 0.95$ (f) $\alpha = 1$

Fig. 2.8. Comparison of the performance of LPLC and ML-LPLC on QCIF *news* (obtained from the implementation using SAMCoW; $R_B = 60$ kbps; BL: reconstruction using the base layer alone; EL: reconstruction using both layers; ML: reconstruction using ML-LPLC)

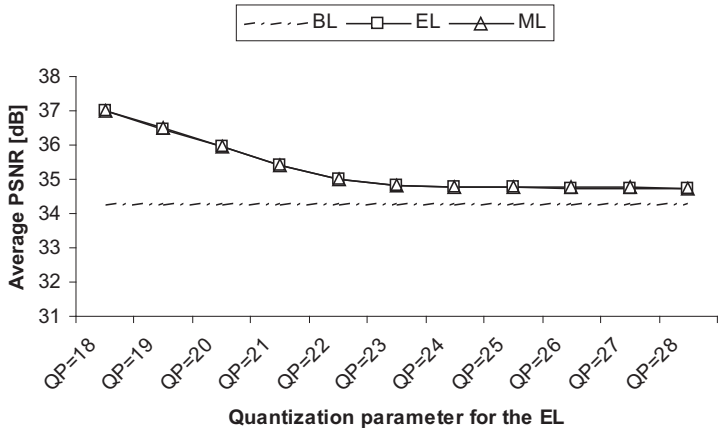
we turned on all seven block-shape options and used the full search range of 16 for motion estimation. We also encoded both the base layer and the enhancement layer using the universal variable length code (UVLC) mode with the same VLC table as the non-scalable coding structure. We adopted one slice for each frame and turned off the rate-distortion optimization option. Since no rate control was implemented in H.26L TML9.4, we encoded all sequences at a frame rate 30 fps and adjusted the quantization parameters of both layers to obtain various decoded video qualities.

As shown in Fig. 2.9, 2.10, 2.11, and 2.12, when the leaky factor α is set to 1.0, a coarse quantization of the enhancement layer results in inferior quality of the reconstruction using both the enhancement layer and the base layer, denoted as "EL" in the figure, relative to the reconstructed base layer, denoted as "BL". Increasing the accuracy of the encoded *mismatch* $\hat{\psi}_E$ carried by the enhancement layer increases the performance of the reconstruction by both layers beyond that of the base layer alone in terms of the decoded video quality. For comparison, we provide the PSNR values in Fig. 2.9, 2.10, 2.11, and 2.12 using the same parameters except that the leaky factor α is set to 0.0. It is seen that in this case the reconstruction by both layers is always superior to that using the base layer, regardless of the quantization steps used to encode the enhancement layer.

In Fig. 2.13 and 2.14 we fix the quantization steps and vary the leaky factor α. It is seen that the performance of the reconstruction by both layers is closely related to the leaky factor α. When the *mismatch* $\hat{\psi}_E$ is coarsely quantized, the reconstruction by both layers becomes inferior in quality to that using the base layer alone when

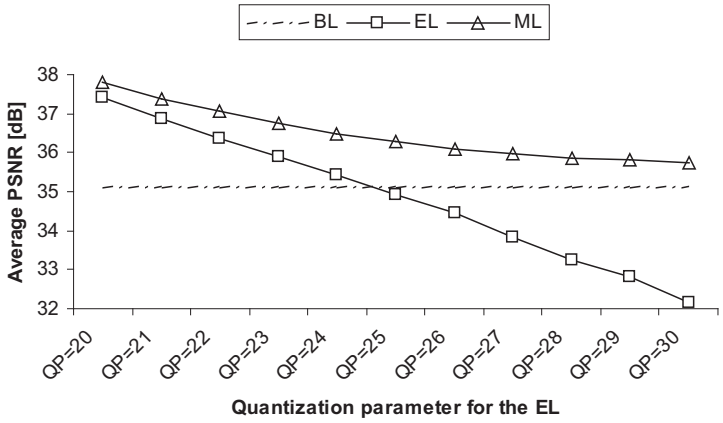


(a) CIF *news* (QP for BL = 24, $\alpha = 1.0$)

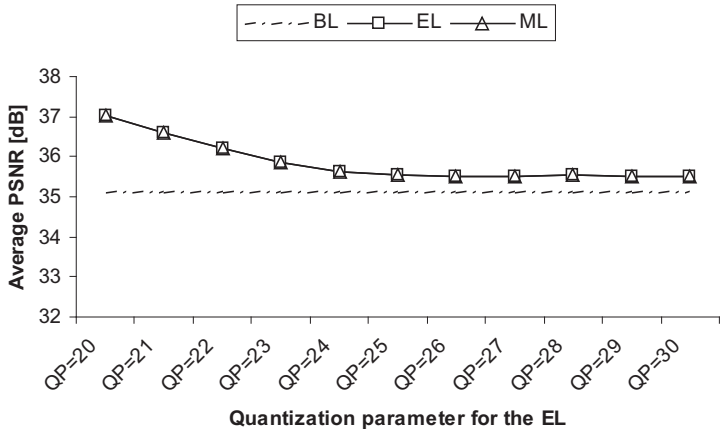


(b) CIF *news* (QP for BL = 24, $\alpha = 0.0$)

Fig. 2.9. Comparison of the performance of LPLC and ML-LPLC on *news* at different quantization parameters predefined for the enhancement layer (obtained from the implementation using H.26L; BL: reconstruction using the base layer alone; EL: reconstruction using both layers; ML: reconstruction using ML-LPLC)

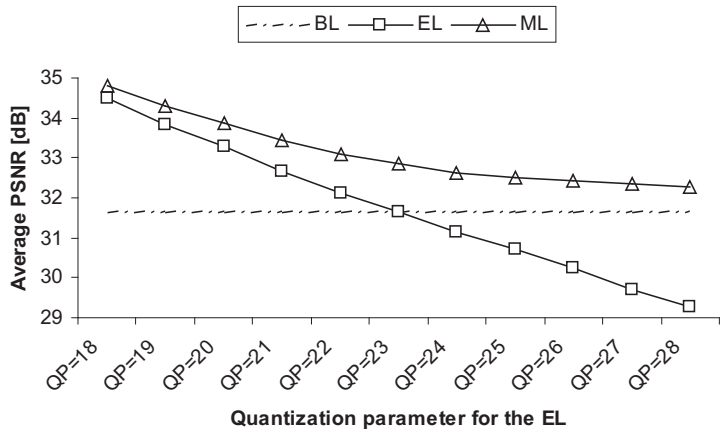


(a) CIF *akiyo* (QP for BL = 26, $\alpha = 1.0$)

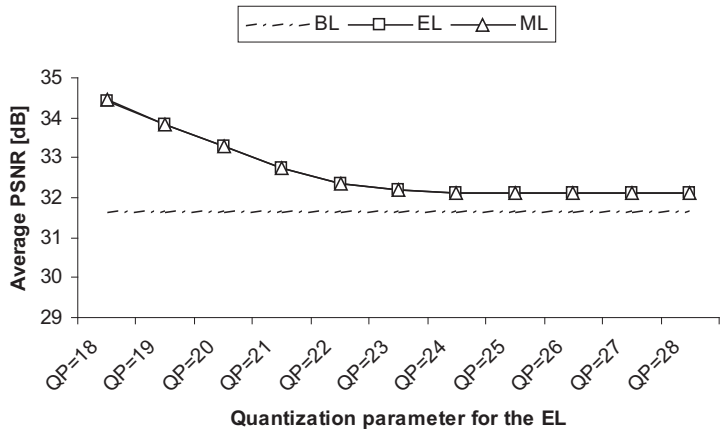


(b) CIF *akiyo* (QP for BL = 26, $\alpha = 0.0$)

Fig. 2.10. Comparison of the performance of LPLC and ML-LPLC on *akiyo* at different quantization parameters predefined for the enhancement layer (obtained from the implementation using H.26L; BL: reconstruction using the base layer alone; EL: reconstruction using both layers; ML: reconstruction using ML-LPLC)

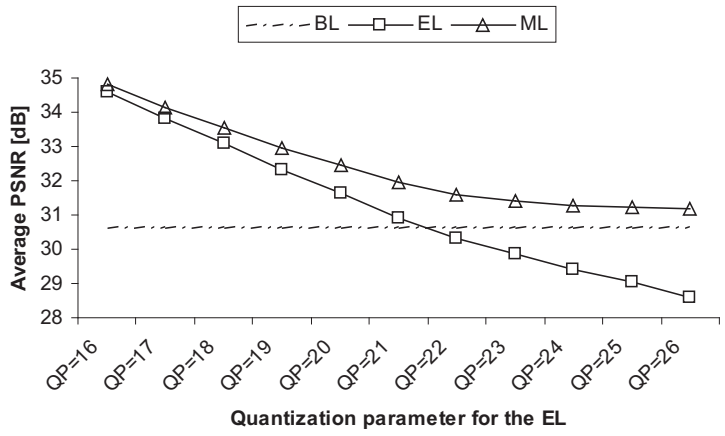


(a) CIF *foreman* (QP for BL = 24, $\alpha = 1.0$)

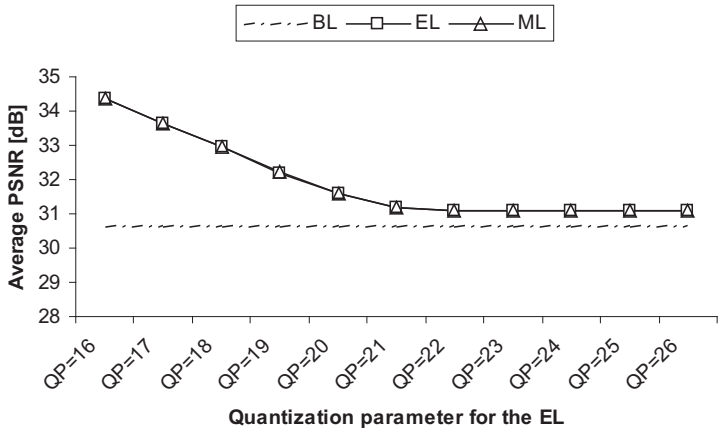


(b) CIF *foreman* (QP for BL = 24, $\alpha = 0.0$)

Fig. 2.11. Comparison of the performance of LPLC and ML-LPLC on *foreman* at different quantization parameters predefined for the enhancement layer (obtained from the implementation using H.26L; BL: reconstruction using the base layer alone; EL: reconstruction using both layers; ML: reconstruction using ML-LPLC)



(a) CIF *bus* (QP for BL = 22, $\alpha = 1.0$)



(b) CIF *bus* (QP for BL = 22, $\alpha = 0.0$)

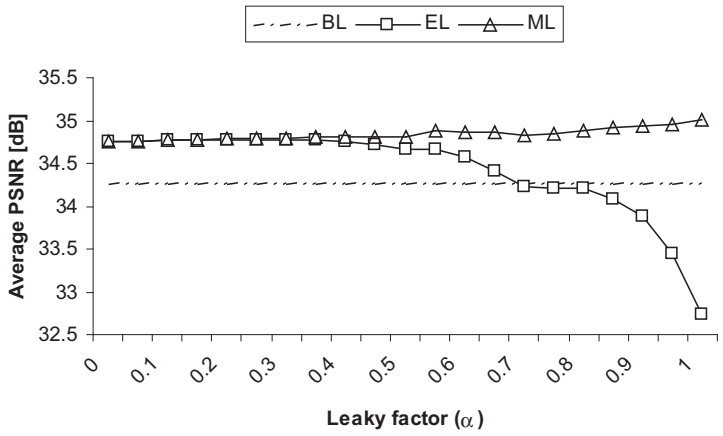
Fig. 2.12. Comparison of the performance of LPLC and ML-LPLC on *bus* at different quantization parameters predefined for the enhancement layer (obtained from the implementation using H.26L; BL: reconstruction using the base layer alone; EL: reconstruction using both layers; ML: reconstruction using ML-LPLC)

Table 2.4
Allocated data rates to the enhancement layer (EL) with respect to different quantization parameters (QP) for the EL when CIF *bus* encoded by LPLC (obtained from the implementation using H.26L; QP for BL = 22; base layer data rate R_B = 593.87 kbps; enhancement layer data rate denoted by R_E)

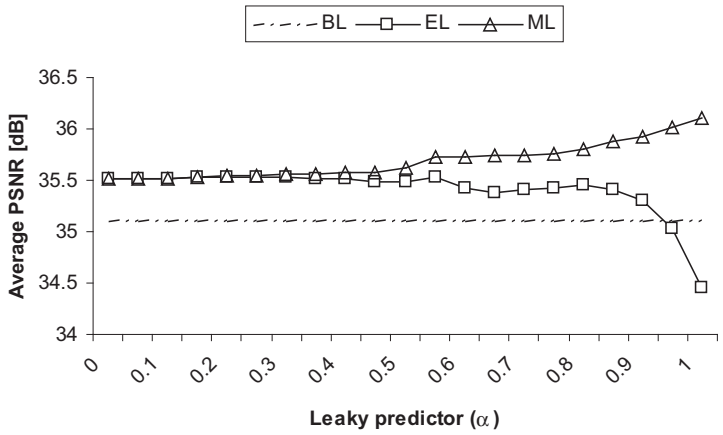
QP for EL	16	17	18	19	20	
R_E (kbps, $\alpha = 1$)	1170.05	958.27	787.42	615.41	470.14	
R_E (kbps, $\alpha = 0$)	1639.59	1269.63	941.38	586.07	298.07	
QP for EL	21	22	23	24	25	26
R_E (kbps, $\alpha = 1$)	323.10	209.18	147.67	112.45	93.62	79.00
R_E (kbps, $\alpha = 0$)	89.85	45.49	45.15	44.94	44.80	44.64

α approaches 1. Generally, the larger the leaky factor, the worse the video quality the enhancement layer provides. Again, it is observed that the decoded quality using both layers does not monotonically vary with an increase in the leaky factor, as observed from the implementation using SAMCoW in Section 2.5.1, which is because the first term of ψ_E in (2.19) is implicitly related with the leaky factor α.

In Table 2.4 and Table 2.5 we provide an example of the data rates obtained by encoding the enhancement layer using LPLC with different parameters, in accordance to the decoded qualities shown in Fig. 2.12(a) and 2.12(b) and Fig. 2.14(b), respectively. From Table 2.4 and Fig. 2.12(a) and 2.12(b), it is observed that for the same quantization parameters, different leaky factors result in different data rates. Generally, the data rate for the enhancement layer decreases with the increase of its

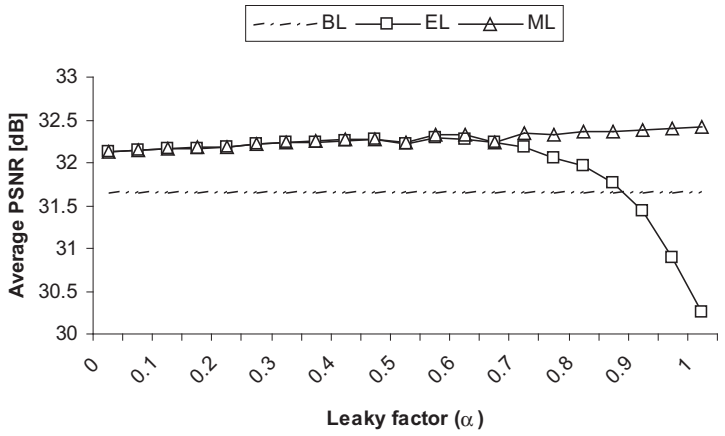


(a) CIF *news* (QP for BL = 24, QP for EL = 26)

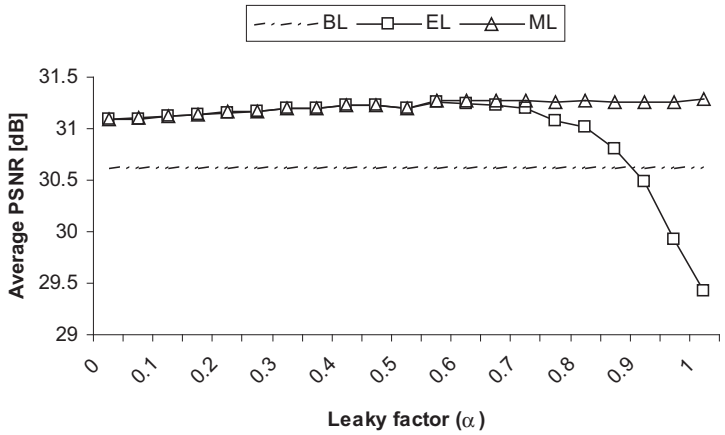


(b) CIF *akiyo* (QP for BL = 26, QP for EL = 26)

Fig. 2.13. Comparison of the performance of LPLC and ML-LPLC on *news* and *akiyo* at different predefined leaky factors (obtained from the implementation using H.26L; BL: reconstruction using the base layer alone; EL: reconstruction using both layers; ML: reconstruction using ML-LPLC)



(a) CIF *foreman* (QP for BL = 24, QP for EL = 26)



(b) CIF *bus* (QP for BL = 22, QP for EL = 24)

Fig. 2.14. Comparison of the performance of LPLC and ML-LPLC on *foreman* and *bus* at different predefined leaky factors (obtained from the implementation using H.26L; BL: reconstruction using the base layer alone; EL: reconstruction using both layers; ML: reconstruction using ML-LPLC)

Table 2.5
Allocated data rates to the enhancement layer (EL) with respect to different leaky factors when CIF *bus* encoded by LPLC (obtained from the implementation using H.26L; QP for BL = 22, QP for EL = 24; base layer data rate R_B = 593.87 kbps; enhancement layer data rate denoted by R_E)

leaky factor α	0.00	0.05	0.10	0.15	0.20	0.25	0.30
R_E (kbps)	44.94	44.95	44.96	44.96	44.97	45.00	45.01
leaky factor α	0.35	0.40	0.45	0.50	0.55	0.60	0.65
R_E (kbps)	45.04	45.10	45.15	45.24	45.30	45.37	45.50
leaky factor α	0.70	0.75	0.80	0.85	0.90	0.95	1.00
R_E (kbps)	45.73	46.61	48.26	51.58	58.55	77.42	112.45

quantization parameter at an arbitrary leaky factor. When the quantization parameter for the enhancement layer is relatively small, the choice of the leaky factor $\alpha = 1$ requires less data, compared to $\alpha = 0$, to provide a comparable decoded quality. In contrast, when the enhancement layer is coarsely quantized, the encoding with $\alpha = 1$ and a fixed quantization parameter results in higher data rate but inferior decoded quality compared to $\alpha = 0$. This implies that LPLC with larger leaky factors is more coding efficient when the enhancement layer carries more accurate information regarding the *mismatch*, while smaller leaky factors are more beneficial when the *mismatch* is coarsely quantized.

It is observed from Table 2.5 that the data rate for the enhancement layer does not change much with the quantization parameter when α is relatively small, such as $\alpha \leq 0.7$ in the example, but greatly increases when α approaches 1. This is partially because we fixed the VLC table for the entropy coding of the enhancement layer regardless of the chosen parameters. We may expect a lower data rate for the scenarios with larger leaky factors if the context-based adaptive binary arithmetic coding (CABAC) mode is exploited [39]. The increase in the data rate also conforms with our analysis of ψ_E in (2.19) that when the *mismatch* carried by the enhancement layer is coarsely quantized, it is more likely to generate an $\hat{\psi}_E$ with a larger magnitude as a result of a larger leaky factor. It has been shown from Fig. 2.14(b) and Table 2.5 that if a large quantization parameter is used for the enhancement layer in LPLC, a larger leaky factor results in a worse decoded quality by using both layers and requires

a higher data rate to encode the enhancement layer, which verifies the deficiency we have specified.

Regardless of the different parameters set for the encoder, all the experimental results from the implementation using H.26L demonstrate that our proposed approach ML-LPLC facilitated by the ML estimation is always superior to or as least as good as the better of the two reconstructions directly obtained from the two layers in terms of the decoded video quality. For instance, for the sequence *foreman*, when both layers choose the same quantization step to be 24 and the leaky factor $\alpha = 1.0$, the reconstruction obtained by ML-LPLC gains 1 dB above the reconstructed base layer and 1.5 dB above the one reconstructed using both layers. An example of different reconstructions for one frame is given in Fig. 2.15 and 2.16. With the decrease in α and the increase in the accuracy of $\hat{\psi}_E$, the reconstruction using both layers finally obtains superior quality beyond the one using the base layer alone, and the performance of ML-LPLC approaches that of the one by both layers but never is inferior to it.

2.6 Conclusions

In this chapter, we contribute the following work for LPLC:

- We have pointed out a deficiency inherent in the LPLC structure, namely that the reconstructed video quality from both the enhancement layer and the base layer cannot be guaranteed to be always superior to that of using the base layer

(a) Original image

(b) Reconstruction using the base layer (PSNR=32.40 dB)

Fig. 2.15. Different reconstructions for the 115th frame of CIF *foreman* using ML-LPLC - I (obtained from the implementation using H.26L; quantization steps for both layers set as 24; leaky factor $\alpha = 1.0$)

(a) Reconstruction using both layers (PSNR=31.58 dB)

(b) Reconstruction using ML-LPLC (PSNR=33.35 dB)

Fig. 2.16. Different reconstructions for the 115th frame of CIF *foreman* using ML-LPLC - II (obtained from the implementation using H.26L; quantization steps for both layers set as 24; leaky factor $\alpha = 1.0$)

alone, even when no drift occurs. In other words, the enhancement layer does not always "enhance" the performance. We highlighted this deficiency using a formulation that describes LPLC.

- We have proposed a general framework that applies to both LPLC and a multiple description coding scheme using motion compensation, known as MDMC. We have addressed two types of similarities, namely Similarity I and Similarity II, that exist between LPLC and MDMC. We refer to the similarity between the enhancement layer in LPLC and the central loop in MDMC, considered from the leaky prediction point of view, as Similarity I. Similarity I conforms with the well-accepted supposition in the literature that the reconstructed quality using both layers in LPLC shall be superior to that using the base layer alone. We refer to the similarity between the enhancement layer in LPLC and the side loops in MDMC, considered from the *mismatch* point of view, as Similarity II. The mismatch in LPLC is the difference between the enhancement layer predicted error frame (PEF) and the reconstructed version of the base layer PEF. Similarity II between MDMC and LPLC confirms the deficiency that might exist in LPLC. The seemingly disagreement between the above two types of similarities is consistent with our analysis that the superiority of the enhancement layer in LPLC is dependent on the leaky factor as well as the accuracy of the encoded enhancement layer itself.

- We have proposed an enhanced LPLC based on maximum-likelihood (ML) estimation, termed ML-LPLC, to address the previously specified deficiency in LPLC. ML-LPLC is capable of addressing the deficiency in LPLC in terms of the coding efficiency regardless of the characteristics of the encoded videos, the data rates allocated to either of the two layers, or the choices of the leaky factors. Our results from both implementations using SAMCoW and H.26L verified the effectiveness of ML-LPLC in alleviating the deficiency in LPLC.

When the enhancement layer suffers from drift errors, ML-LPLC cannot guarantee to always provide a reconstruction superior in quality to that using the base layer or using both layers, since the ML coefficients in (2.34) are derived from the error-free reconstructions using the two layers.

Nevertheless, ML-LPLC is beneficial especially when it is coupled with mode-adaptive approaches [23–25]. We have observed that the leaky factor is critical for the LPLC scheme. It has three functionalities: (1) It affects the coding efficiency; (2) It affects the error resilience performance; (3) It determines the superiority of the reconstruction by both layers. For the second functionality, it is straightforward to show that $\alpha = 0$ provides the best performance with respect to error resilience. For the third functionality, our experimental results have shown that the reconstruction using both layers is usually superior to that of using the base layer alone when $\alpha \leq 0.5$, and the deficiency of LPLC is usually manifested when $\alpha > 0.5$. Hence, for relatively small leaky factors, the enhancement layer in LPLC usually helps obtain a reconstruction superior in quality beyond that obtained by the base layer, and

the advantage of ML-LPLC is not evident. When the coding efficiency, i.e. the first functionality of the leaky factor is the predominant goal of a scalable video coding approach, the optimal leaky factor varies across different data rates allocated to the two layers as well as across different video sequences. It is seen from our experimental results that when the data rate allocated to the enhancement layer is sufficiently large, the decoded quality always increases with the increase of the leaky factor at an arbitrary data rate, implying that $\alpha = 1$ is always optimal in terms of the coding efficiency in this case.

Hence, which value should be chosen for the leaky factor in LPLC is closely related to the application. Adaptively adjusting the leaky factor thus becomes a good alternative. As we pointed in Section 2.1, approaches for LPLC that coupled a mode-adaptive scalable coding scheme with a drift-management system have already been presented in the literature [23–25]. From the LPLC perspective, to adaptively select a coding mode is equivalent to adaptively adjusting the leaky factor. ML-LPLC could be used in combination with these mode-adaptive approaches. Once a larger leaky factor is chosen by the adaptive mechanism, we could benefit from the advantage of ML-LPLC. ML-LPLC is capable of addressing the deficiency in LPLC in terms of the coding efficiency regardless of the characteristics of the encoded videos, the data rates allocated to either of the two layers, or the choices of the leaky factors. Our experimental results have shown that even when the data rate allocated to the enhancement layer is sufficiently large so that the decoded quality using both layers is far well above that obtained using the base layer, ML-LPLC is able to provide

further gains in quality compared to LPLC when the leaky factor α approaches 1. We will be exploring the mode-adaptive ML-LPLC approach facilitated with the drift-managing mechanism in our future work.

Also, ML-LPLC requires reasonable extra computation. As can be seen from equations (2.32), (2.33), and (2.34), the calculation of the ML-coefficients requires five additions and three multiplications for each pixel combined with three extra additions and one extra division for each frame. Once the ML-coefficients are obtained, it is seen from (2.31) that to obtain the ML reconstruction at the decoder only requires two scaling and one addition for each pixel. Moreover, the transmission of the side information for ML-LPLC, i.e., the transmission of the ML coefficients requires negligible additional data rate compared to the payload data rates. For example, an extra data rate of 180 bps is required by ML-LPLC, regardless of the frame size, if a video sequence with chrominance components is encoded at frame rate 10 fps and the ML coefficients are encoded using 6-bit each, as in our experiments.

3. RATE DISTORTION ANALYSIS OF LEAKY PREDICTION LAYERED VIDEO CODING

3.1 Introduction

As discussed in Section 1.1.2 of Chapter 1, leaky prediction layered video coding (LPLC) partially includes the enhancement layer in the motion compensated prediction (MCP) loop, by using a leaky factor between 0 and 1, to balance between coding efficiency and error resilience performance. In this chapter, we address the theoretic analysis of LPLC using two different approaches [61, 62].

First, we derive the rate distortion functions for LPLC using rate distortion theory [63]. An alternative block diagram of LPLC is first developed, which significantly simplifies the theoretic analysis. Closed form expressions are obtained for two scenarios of LPLC, as a function of the leaky factor, one where the enhancement layer stays intact and the other where the enhancement layer suffers from error drift due to channel noise corruption or data rate truncation. Secondly, we exploit quantization noise modeling to theoretically analyze the rate distortion performance of LPLC. We derive a second set of closed form rate distortion functions related to the leaky factor for the two scenarios, where drift error occurs in the enhancement layer and no drift occurs within the motion compensation loop. Theoretical results of both

analysis approaches are evaluated and compared with respect to different choices of the leaky factor, which conform with the operational results.

3.2 Rate Distortion Analysis of LPLC Using Rate Distortion Theory

The analysis of MCP based video coding, derived from rate distortion theory, is presented in [58, 64–66]. Essentially, the analysis characterizes the MCP operation by a three-dimensional (3D) stochastic filter combined with an optimum forward channel. The 3D stochastic filter fulfills the operations of time delay, motion compensation, and spatial filtering. For simplifying the analysis, an arbitrary video sequence is considered to contain exclusively translational motions, neglecting any other motions such as rotation, zoom-in or zoom-out, occlusion/unocclusion, or illuminance changes. The optimum forward channel, derived in [63], yields the parametric rate distortion function, in the mean-square-error (MSE) sense, for coding a two-dimensional (2D) Gaussian stationary signal. The optimum forward channel is used in the analysis of [58, 64–66] to characterize the coding of the 2D predicted error frames (PEFs), making the analysis independent of any particular algorithms used to encode the PEFs.

The rate distortion analysis for non-scalable MCP video coding, first developed in [64], obtained a closed form expression for the power spectral density (PSD) of the PEF, in relation to the PSD of the input video frame as well as the probability distribution of the motion vector estimation errors. Using the closed form function of the PEF, closed form rate distortion functions were further derived. The rate

distortion analysis of motion estimation with fractional pel accuracy was addressed in [67], and that of multihypothesis motion estimation was developed in [68–70].

The rate distortion analysis of conventional layered scalable video coding, which includes one MCP loop in the base layer and excludes the enhancement layer from the MCP loop, was developed in [58, 65, 66]. The MCP rate was explicitly defined as the data rate, in bits per symbol, that is incorporated within the MCP loop at the encoder. The analysis first derived the rate distortion functions for both the MSE optimal layered image codec and the cascaded optimal image codec. The rate distortion analysis of layered video coding was then developed, with two scenarios derived in closed form, where the bitstream is decoded *above* and *below* the MCP rate.

When a scalable coded bitstream is decoded above the MCP rate, the base layer at the decoder is consistent with that at the encoder, implying that no drift occurs. The theoretic analysis confirmed the well-accepted fact that layered scalable coding always demands more or at least as much data rate as required by the non-scalable coding approach to obtain the same distortion. When the bitstream is decoded below the MCP rate, a mismatch between the base layer at the decoder and the base layer at the encoder occurs, leading to prediction error drift. It was theoretically demonstrated that the distortion in this case steeply increased with the decrease of the decoding data rate. The theoretic analysis was shown to agree with the operational rate distortion results published in the literature.

In this section, we first briefly summarize the theoretic results presented in [64] and [58], and then develop the rate distortion analysis of LPLC using rate distortion theory by extending the work in [58]. We describe a block diagram that features the leaky prediction in layered video coding but is amenable to theoretic analysis. We derive the rate distortion functions for LPLC in closed form for one scenario where the enhancement layer is intact and the other where it has drift.

For each scenario we obtain closed form parametric rate distortion functions in relation to three parameters of the LPLC structure: the PSD of the input video frame, the probability distribution of the motion vector estimation errors, as well as the leaky factor. We demonstrate that the leaky factor is critical to the performance of coding efficiency, and validate that with the partial or full inclusion of the enhancement layer in the MCP loop, LPLC does improve the coding efficiency in contrast to the conventional layered scalable coding. We also show that the leaky factor is critical to error resilience performance when the enhancement layer in LPLC suffers from prediction drift.

For the notation of this chapter, we choose lower case letters to denote the signals and upper case to indicate the Fourier transform[1]. We use λ to denote the spatial variables x and y, where $\lambda = (x, y)$. A 2D signal may be denoted as $s(\lambda) = s(x, y)$, or simply written as $\{s\}$. The resulting Fourier transform is denoted as $S(\Lambda)$, or simply written as S, where $\Lambda = (\omega_x, \omega_y)$, and ω_x and ω_y denote the spatial frequency variables. If the time dimension is considered, the notation is expanded to $(\lambda, t) =$

[1]Strictly speaking, the Fourier transform of a random signal does not exist. We use this concept for the sake of simpler notation. Note that this does not affect the final theoretical results.

(x, y, t), where t denotes time. A 3D signal may be denoted as $s(\lambda, t) = s(x, y, t)$, or $\{s\}$. The resulting Fourier transform is denoted as $S(\Omega)$, or S, where $\Omega = (\Lambda, \omega_t) = (\omega_x, \omega_y, \omega_t)$, and ω_t denotes the temporal frequency variable. We use symbol Φ to indicate the PSD of a signal or the cross spectral density of two signals, with the subscripts specifying the signal(s).

3.2.1 Rate Distortion Functions for Two-Dimensional Image Coding

As background knowledge, we summarize the fundamental results of the theoretic analysis for 2D image coding derived from rate distortion theory.

Theorem 3.2.1 *The parametric representation of the MSE rate distortion functions for a 2D stationary Gaussian source $\{s\}$ is given by [63]*

$$D^\theta = \frac{1}{4\pi^2} \iint_\Lambda \min\{\theta, \Phi_{ss}(\Lambda)\} d\Lambda, \tag{3.1}$$

$$R^\theta = \frac{1}{8\pi^2} \iint_\Lambda \max\left\{0, \log_2\left(\frac{\Phi_{ss}(\Lambda)}{\theta}\right)\right\} d\Lambda, \tag{3.2}$$

where each integral extends over $[-\pi, \pi]$ for discrete time processes, and $[-\infty, \infty]$ for continuous time processes. $\Phi_{ss}(\Lambda)$ denotes the PSD of the signal $\{s\}$ at the spatial frequencies $\Lambda = (\omega_x, \omega_y)$, and θ denotes the parameter ranging over $[0, ess\sup \Phi(\Lambda)]$, with ess $\sup \Phi_{ss}(\Lambda)$ indicating the essential supremum of $\Phi_{ss}(\Lambda)$.

The following proposition simplifies the analysis of the rate distortion performance of image and video signals.

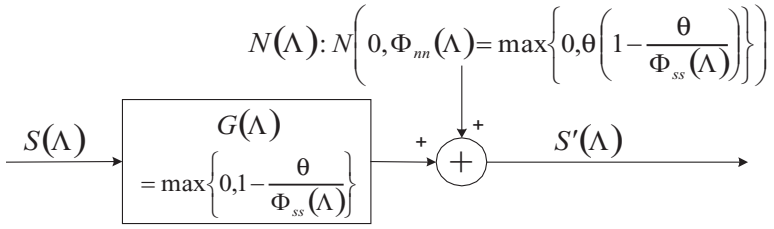


Fig. 3.1. Optimum forward channel that yields the Gaussian MSE rate distortion function

Proposition 3.2.1 *The optimum forward channel yielding the MSE parametric rate distortion functions for a Gaussian process* $\{s\}$*, as given by (3.1) and (3.2), is composed of a non-ideal bandlimited filter over* $\{\Lambda : \Phi_{ss}(\Lambda) > \theta\}$*, plus the addition of an independent non-white bandlimited Gaussian noise over* $\{\Lambda : \Phi_{ss}(\Lambda) > \theta\}$*, as shown in Fig. 3.1 [63].*

In the optimum forward channel shown in Fig. 3.1, the transform function, $G(\Lambda)$, is represented by

$$G(\Lambda) = \max\left\{0, 1 - \frac{\theta}{\Phi_{ss}(\Lambda)}\right\}, \tag{3.3}$$

and the additive, independent Gaussian noise has a zero mean and a PSD as follows

$$\Phi_{nn}(\Lambda) = \max\left\{0, \theta\left(1 - \frac{\theta}{\Phi_{ss}(\Lambda)}\right)\right\}. \tag{3.4}$$

It is easy to prove Proposition 3.2.1 with the help of the following lemma:

Lemma 1 *Assume* $\{s\}$ *and* $\{s'\}$ *are jointly 2D stationary Gaussian processes, with* $\Phi_{ss}(\Lambda)$*,* $\Phi_{s's'}(\Lambda)$*, and* $\Phi_{ss'}(\Lambda)$ *denoting the PSD of* $\{s\}$*, the PSD of* $\{s'\}$*, and the cross spectral density between* $\{s\}$ *and* $\{s'\}$ *respectively. The mutual information rate between* $\{s\}$ *and* $\{s'\}$ *is then given by [71]*

$$I(S; S') = -\frac{1}{8\pi^2}\iint_{\Lambda} \log\left[1 - \frac{|\Phi_{ss'}(\Lambda)|^2}{\Phi_{ss}(\Lambda)\Phi_{s's'}(\Lambda)}\right] d\Lambda. \tag{3.5}$$

From the optimum forward channel in Fig. 3.1, we obtain the PSD of the channel output signal $\{s'\}$ as

$$\Phi_{s's'}(\Lambda) = \max\left\{0, \Phi_{ss}(\Lambda) - \theta\right\}, \tag{3.6}$$

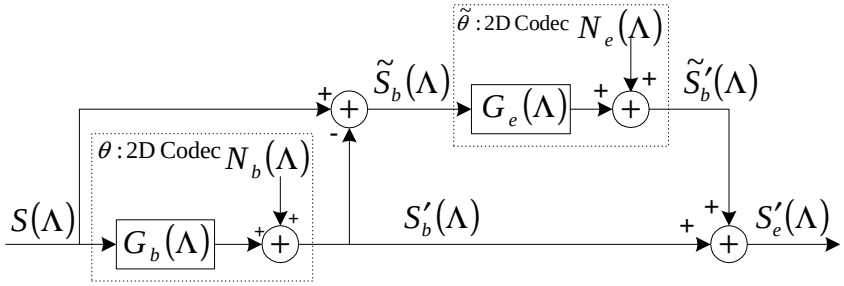


Fig. 3.2. Block diagram of an MSE optimum layered image codec

and the PSD of the difference signal between the input and the output of the channel $\{\tilde{s} \triangleq s - s'\}$ as

$$\Phi_{\tilde{s}\tilde{s}}(\Lambda) = \min\{\theta, \Phi_{ss}(\Lambda)\}. \tag{3.7}$$

Note that the MSE distortion in (3.1) is the inverse-Fourier transform of the PSD $\Phi_{\tilde{s}\tilde{s}}(\Lambda)$ in (3.7).

Using the optimum forward channel shown in Fig. 3.1, two scenarios of 2D image coding are further analyzed in [58]:

Scenario I for 2D image coding - layered image coding: An MSE optimum layered image codec is shown in Fig. 3.2, where two optimum forward channels are included, one representing the encoding of the base layer image and the other representing the encoding of the enhancement layer residue. The residue is the difference between the base layer reconstruction and the original 2D image. The rate distortion func-

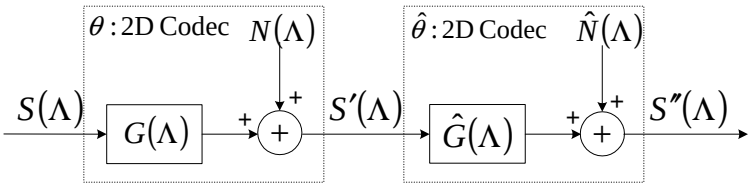


Fig. 3.3. Block diagram of an MSE optimum cascaded image codec

tions by both layers in the layered image coding structure, where the base layer is parameterized by θ and the enhancement layer by $\tilde{\theta}$, are

$$D_e^{I,\theta,\tilde{\theta}} = \frac{1}{4\pi^2}\iint_{\Lambda} \min\{\min\{\theta,\tilde{\theta}\},\Phi_{ss}(\Lambda)\}d\Lambda, \tag{3.8}$$

$$R_e^{I,\theta,\tilde{\theta}} = \frac{1}{8\pi^2}\iint_{\Lambda} \max\left\{0,\log_2\left(\frac{\Phi_{ss}(\Lambda)}{\min\{\theta,\tilde{\theta}\}}\right)\right\}d\Lambda. \tag{3.9}$$

It is straightforward to show (3.8) by noting that $s - s'_e = \tilde{s}_b - \tilde{s}'_b$, and to show (3.9) by summing the mutual information between $\{s\}$ and $\{s'_b\}$ plus the mutual information between $\{\tilde{s}_b\}$ and $\{\tilde{s}'_b\}$.

Scenario II for 2D image coding - cascaded image coding: The rate distortion functions of the cascaded non-scalable image codec, as shown in Fig. 3.3, where the first codec is parameterized by θ and the second codec by $\hat{\theta}$, are

$$D^{II,\theta,\tilde{\theta}} = \frac{1}{4\pi^2}\iint_{\Lambda} \min\{\tilde{\theta},\Phi_{ss}(\Lambda)\}d\Lambda, \tag{3.10}$$

$$R^{II,\theta,\tilde{\theta}} = \frac{1}{8\pi^2}\iint_{\Lambda} \max\left\{0,\log_2\left(\frac{\Phi_{ss}(\Lambda)}{\tilde{\theta}}\right)\right\}d\Lambda, \tag{3.11}$$

where $\tilde{\theta} = \theta + \hat{\theta}$. Note that (3.10) can be derived by noticing that $\{\tilde{s}_b \triangleq s - s'_b\}$ and $\{\tilde{s}'_b \triangleq s'_b - s''_b\}$ are uncorrelated [58]. To show (3.11), we introduce the following proposition for cascaded Gaussian MSE optimum forward channels:

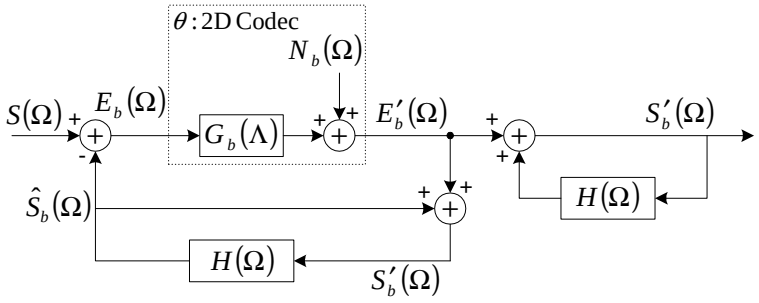


Fig. 3.4. Block diagram of a non-scalable MCP video codec

Proposition 3.2.2 *The cascaded Gaussian MSE optimum forward channels are still optimal in the rate distortion sense. Furthermore, the parameter of the equivalent optimum forward channel is the sum of the parameters featuring each of the cascaded channels.*

Proposition 3.2.2 can be proved in a similar way to the proof of Proposition 3.2.1 using Lemma 1. The details of the proof of Proposition 3.2.2 are given in Appendix A. The proof also includes the derivation of (3.10) and (3.11), and presents a different approach to obtaining the distortion function in (3.10) compared to the work in [58].

3.2.2 Rate Distortion Functions for Non-Scalable Video Coding

The theoretic analysis of non-scalable, MCP based video coding using rate distortion theory, first addressed in [64], uses the optimum forward channel in Fig. 3.1 to characterize the coding of the 2D PEF, $\{e\}$, thus independent of the specific algorithms used to encode the PEFs, as shown in Fig. 3.4. For the MCP loop, the

analysis uses a 3D stochastic filter, $H(\Omega)$, to characterize the combined operations of time delay, motion compensation, and spatial filtering as follows

$$H(\Omega) = H(\Lambda, \omega_t) = F(\Lambda) \exp\{-j\omega_x \hat{d}_x - j\omega_y \hat{d}_y - j\omega_t \Delta t\}, \tag{3.12}$$

where $F(\Lambda)$ denotes the spatial filter, Δt represents the temporal sampling interval, and $(\hat{d}_x, \hat{d}_y)$ denote the estimated motion vectors.

Combining $H(\Omega)$ with the optimum forward channel, the parametric rate distortion functions of the non-scalable video codec are [64]

$$D^\theta = \frac{1}{4\pi^2} \iint_\Lambda \min\{\theta, \Phi_{ee}^\theta(\Lambda)\} d\Lambda, \tag{3.13}$$

$$R^\theta = \frac{1}{8\pi^2} \iint_\Lambda \max\left\{0, \log_2\left(\frac{\Phi_{ee}^\theta(\Lambda)}{\theta}\right)\right\} d\Lambda, \tag{3.14}$$

where $\Phi_{ee}^\theta(\Lambda)$ denotes the PSD of the PEF $\{e\}$, which is dependent of the parameter θ and approximated by

$$\Phi_{ee}^\theta(\Lambda) \approx \Phi_{ee}^{appr,\theta}(\Lambda) = \begin{cases} \Phi_{ss}(\Lambda) & \Lambda : \Phi_{ss}(\Lambda) \le \theta \\ \max\{\theta, \Phi_{ee}^{I,\theta}(\Lambda)\} & \Lambda : \Phi_{ss}(\Lambda) > \theta \end{cases}, \tag{3.15}$$

and

$$\begin{aligned} \Phi_{ee}^{I,\theta}(\Lambda) &= \Phi_{ss}(\Lambda)[1 - 2\mathrm{Re}\{F(\Lambda)P^*(\Lambda)\} + |F(\Lambda)|^2] + \theta|F(\Lambda)|^2 \\ &= \Phi_{ss}(\Lambda)[1 - 2\mathrm{Re}\{F^*(\Lambda)P(\Lambda)\} + |F(\Lambda)|^2] + \theta|F(\Lambda)|^2, \end{aligned} \tag{3.16}$$

where $\mathrm{Re}\{\cdot\}$ represents the real part of a complex function. $P(\Lambda)$ denotes the characteristic function of the motion vector estimation error, $(\Delta d_x, \Delta d_y)$, where

$$\begin{pmatrix} \Delta d_x \\ \Delta d_y \end{pmatrix} = \begin{pmatrix} d_x \\ d_y \end{pmatrix} - \begin{pmatrix} \hat{d}_x \\ \hat{d}_y \end{pmatrix}, \tag{3.17}$$

and (d_x, d_y) represent the ideal motion vectors. It is discussed in [64] and [58] that the spatial filter $F(\Lambda)$ may be chosen as 0 for intra-frame coding, or 1 for inter-frame coding without the spatial filtering of the motion compensated picture. The optimal $F(\Lambda)$, denoted as F_{opt} that minimizes $\Phi_{ee}^{I,\theta}(\Lambda)$ in (3.16), is

$$F_{\text{opt}}(\Lambda) = P(\Lambda)\frac{\Phi_{ss}(\Lambda)}{\Phi_{ss}(\Lambda)+\theta}, \qquad \text{for } \Lambda : \Phi_{ss}(\Lambda) > \theta, \tag{3.18}$$

which results

$$\Phi_{ee,\min}^{I,\theta}(\Lambda) = \Phi_{ss}(\Lambda)\left(1 - \frac{|P(\Lambda)|^2\Phi_{ss}(\Lambda)}{\Phi_{ss}(\Lambda)+\theta}\right). \tag{3.19}$$

It was pointed out in [64] that $F_{\text{opt}}(\Lambda)$ in (3.18) can be interpreted as a combination of two Wiener filters, one characterizing the additive noise in the coding of the PEF and one taking into account the motion vector estimation errors. Moreover, when $\Phi_{ss}(\Lambda) \gg \theta$, $F_{\text{opt}}(\Lambda)$ can be approximated by

$$F_{\text{opt}}(\Lambda) \approx P(\Lambda), \tag{3.20}$$

implying that the optimal spatial filter is only related to the motion vector estimation errors.

3.2.3 Rate Distortion Functions for Conventional Layered Video Coding

The rate distortion analysis of conventional layered video coding using rate distortion theory is presented in [58], and was developed by extending the work on the rate distortion analysis of the non-scalable MCP video coding as we discussed in Subsection 3.2.2. In this section we summarize the major results presented in [58].

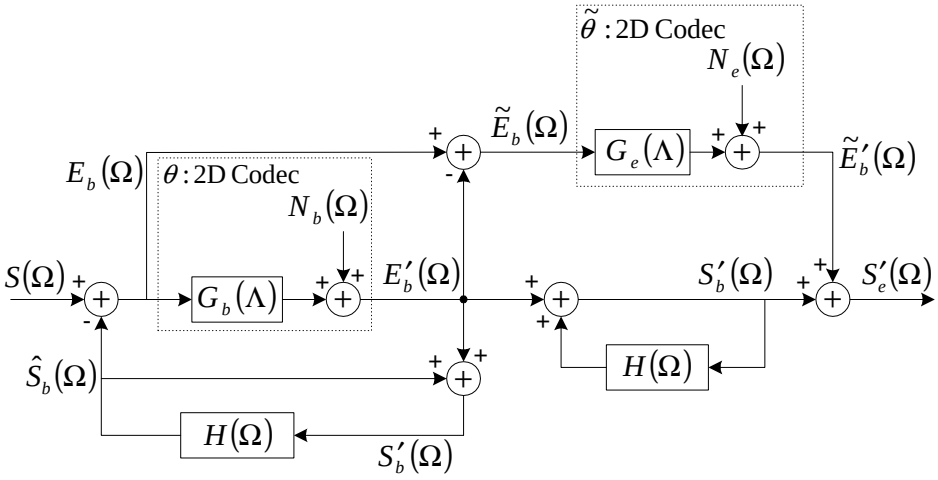


Fig. 3.5. Block diagram of a conventional layered video codec when the base layer is decoded above the MCP rate

Scenario I for conventional layered video coding: As shown in Fig. 3.5, two optimum forward channels are included, yielding the rate distortion optimized 2D signal compression of the base layer and the enhancement layer respectively. As addressed in [64], if all the 3D signals in Fig. 3.5 are assumed as time discrete with a temporal sampling interval Δt, the 3D PSD of each signal is periodic in terms of the temporal frequency ω_t, with a period of $2\pi/\Delta t$. Thus, the 2D PSD of each 3D signal can be obtained by integrating its 3D PSD over one temporal period as follows

$$\begin{aligned} \Phi_{ee}(\Lambda) = \Phi_{ee}(\omega_x, \omega_y) &= \frac{\Delta t}{2\pi} \int_{\omega_t=0}^{\frac{2\pi}{\Delta t}} \Phi_{ee}(\omega_x, \omega_y, \omega_t) d\omega_t \\ &= \frac{\Delta t}{2\pi} \int_{\omega_t=0}^{\frac{2\pi}{\Delta t}} \Phi_{ee}(\Omega) d\omega_t, \end{aligned} \tag{3.21}$$

where $\{e\}$ represents an arbitrary 3D signal in Fig. 3.5. Referring to (3.21) and the rate distortion analysis for *Scenario I for 2D image coding*, we have the rate distortion functions for both layers of the conventional layered video coding, when the parameters satisfy $\tilde{\theta} \leq \theta$, as follows

$$D_e^{I,\theta,\tilde{\theta}} = \frac{1}{4\pi^2} \iint_{\Lambda} \min\{\tilde{\theta}, \Phi^{\theta}_{e_b e_b}(\Lambda)\} d\Lambda, \quad \text{for } \tilde{\theta} \leq \theta, \tag{3.22}$$

$$R_e^{I,\theta,\tilde{\theta}} = \frac{1}{8\pi^2} \iint_{\Lambda} \max\left\{0, \log_2\left(\frac{\Phi^{\theta}_{e_b e_b}(\Lambda)}{\tilde{\theta}}\right)\right\} d\Lambda, \quad \text{for } \tilde{\theta} \leq \theta, \tag{3.23}$$

where θ denotes the parameter for the optimum forward channel in the base layer and $\tilde{\theta}$ for the enhancement layer. The PSD of the PEF in the base layer, $\Phi^{\theta}_{e_b e_b}(\Lambda)$, is identical to $\Phi^{\theta}_{ee}(\Lambda)$ in (3.15) in non-scalable video coding, and hence approximated as follows

$$\Phi^{\theta}_{e_b e_b}(\Lambda) \approx \Phi^{appr,\theta}_{e_b e_b}(\Lambda) = \begin{cases} \Phi_{ss}(\Lambda) & \Lambda : \Phi_{ss}(\Lambda) \leq \theta \\ \max\{\theta, \Phi^{I,\theta}_{e_b e_b}(\Lambda)\} & \Lambda : \Phi_{ss}(\Lambda) > \theta \end{cases}, \tag{3.24}$$

where

$$\Phi^{I,\theta}_{e_b e_b}(\Lambda) = \Phi_{ss}(\Lambda)[1 - 2\mathrm{Re}\{F(\Lambda)P^*(\Lambda)\} + |F(\Lambda)|^2] + \theta|F(\Lambda)|^2. \tag{3.25}$$

Using (3.24), we see that $\Phi^{\theta}_{e_b e_b}(\Lambda) \geq \theta$ when $\Phi_{ss}(\Lambda) > \theta$. Hence, the distortion function in (3.22) can be simplified to

$$D_e^{I,\theta,\tilde{\theta}} = \frac{1}{4\pi^2}\iint_{\Lambda} \min\{\tilde{\theta}, \Phi_{ss}(\Lambda)\}d\Lambda, \text{ for } \tilde{\theta} \leq \theta. \tag{3.26}$$

Note that the PSD of the input signal to the enhancement layer forward channel, $\{\tilde{e}_b\}$, satisfies

$$\Phi_{\tilde{e}_b\tilde{e}_b}(\Lambda) = \min\{\theta, \Phi^{\theta}_{e_b e_b}(\Lambda)\} = \min\{\theta, \Phi_{ss}(\Lambda)\} \leq \theta. \tag{3.27}$$

Thus, when $\tilde{\theta} > \theta$, the transform function in the enhancement layer in Fig. 3.5, $G_e(\Lambda)$, becomes

$$G_e(\Lambda) = \max\left\{0, 1 - \frac{\tilde{\theta}}{\Phi_{\tilde{e}_b\tilde{e}_b}(\Lambda)}\right\} = 0. \tag{3.28}$$

Consequently, the data rate consumed by the enhancement layer is zero and no distortion is further caused beyond that caused by the base layer. The rate distortion function of the MCP layered codec in Fig. 3.5 is then fixed at one point that is specified by the base layer, when θ is fixed and $\tilde{\theta}$ varies between θ and infinity. This is given by

$$D_b^{\theta} = \frac{1}{4\pi^2}\iint_{\Lambda} \min\{\theta, \Phi_{ss}(\Lambda)\}d\Lambda, \tag{3.29}$$

$$R_b^{\theta} = \frac{1}{8\pi^2}\iint_{\Lambda} \max\left\{0, \log_2\left(\frac{\Phi^{\theta}_{e_b e_b}(\Lambda)}{\theta}\right)\right\} d\Lambda. \tag{3.30}$$

Hence, we have the universal form of the rate distortion functions for *Scenario I for conventional layered video coding* as follows

$$D_e^{I,\theta,\tilde{\theta}} = \frac{1}{4\pi^2}\iint_{\Lambda} \min\{\min\{\tilde{\theta},\theta\}, \Phi^{\theta}_{e_b e_b}(\Lambda)\} d\Lambda, \tag{3.31}$$

$$R_e^{I,\theta,\tilde{\theta}} = \frac{1}{8\pi^2}\iint_{\Lambda} \max\left\{0, \log_2\left(\frac{\Phi^{\theta}_{e_b e_b}(\Lambda)}{\min\{\tilde{\theta},\theta\}}\right)\right\} d\Lambda. \tag{3.32}$$

Scenario II for conventional layered video coding: The block diagram of layered video coding when the base layer is decoded below the MCP rate is shown in Fig. 3.6. The enhancement layer disappears and prediction drift error occurs in the MCP loop due to the mismatch between the base layer at the decoder and that at the encoder. The video codec in Fig. 3.6 assumes that the data rate is truncated in a rate distortion optimization manner, hence using a second optimum forward channel to characterize the loss mechanism for prediction drift. Fundamentally, the data rate truncation is modelled as a re-encoding procedure of the encoded base layer using an "imaginary" codec operating at the rate distortion bound. The use of the second optimum forward channel provides a way to theoretically analyze the loss mechanism for prediction drift that takes into account both distortion and rate effects [58].

As proved in Appendix A, the two optimum forward channels in cascade, connecting $\{e_b\}$ and $\{e_b''\}$ in Fig. 3.6, are equivalent to one optimum forward channel with parameter $\theta + \hat{\theta}$. Here θ is the parameter for the first optimum forward channel and $\hat{\theta}$ of the second channel. Based on (3.21), we obtain the rate distortion functions

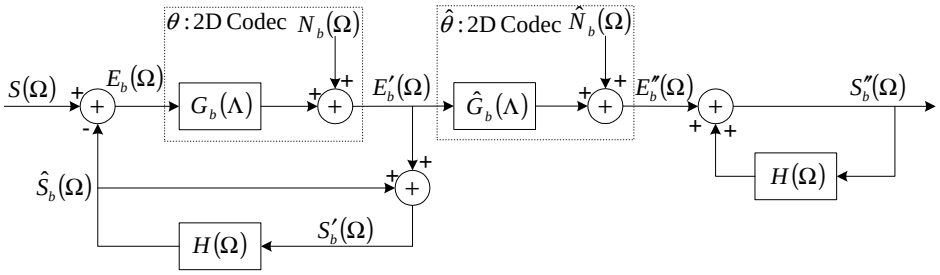


Fig. 3.6. Block diagram of a conventional layered video codec when the base layer is decoded below the MCP rate

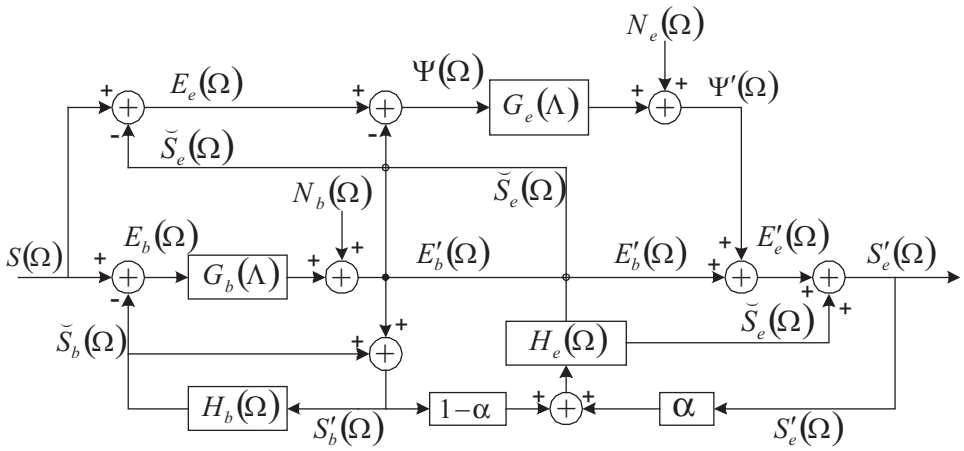


Fig. 3.7. Block diagram of a leaky prediction layered video codec (LPLC)

for the conventional layered video codec, where the base layer is decoded below the MCP rate, as follows

$$\begin{aligned} D^{II,\theta,\tilde{\theta}} &= \frac{1}{4\pi^2}\iint_{\Lambda}\min\{\theta,\Phi_{ss}(\Lambda)\} \\ &\quad + \frac{1}{1-|F(\Lambda)|^2}\min\left\{\tilde{\theta}-\theta,\max\left\{0,\Phi^{\theta}_{e_be_b}(\Lambda)-\theta\right\}\right\}d\Lambda, \qquad (3.33) \\ R^{II,\theta,\tilde{\theta}} &= \frac{1}{8\pi^2}\iint_{\Lambda}\max\left\{0,\log_2\left(\frac{\Phi^{\theta}_{e_be_b}(\Lambda)}{\tilde{\theta}}\right)\right\}d\Lambda, \qquad (3.34) \end{aligned}$$

where $\tilde{\theta} = \theta + \hat{\theta} \geq \theta$.

3.2.4 Rate Distortion Functions for LPLC Using Rate Distortion Theory

Unlike the conventional layered coding structure described in Fig. 3.5, LPLC introduces a second MCP loop in the enhancement layer that uses the same motion vectors as the base layer, and buffers $\alpha(s'_e(\lambda,t) - s'_b(\lambda,t)) + s'_b(\lambda,t)$ as the reference for encoding the video signal sampled at time instance $t + \Delta t$. $\{s'_e\}$ denotes the

reconstructed video signal using both layers and $\{s'_b\}$ denotes the reconstruction using the base layer alone. α is the leaky factor, taking on a value between 0 and 1. Equivalently, a linear combination of the two reconstructed signals $\{s'_e\}$ and $\{s'_b\}$, namely $\alpha s'_e(\lambda, t) + (1 - \alpha)s'_b(\lambda, t)$, is used as the reference in the enhancement layer MCP loop. The mismatch signal, $\{\psi\}$, is the difference between the PEF of the MCP step in the enhancement layer, $\{e_e\}$, and the encoded PEF of the MCP step in the base layer, $\{e'_b\}$. $\{\psi\}$ is encoded and carried by the enhancement layer in LPLC. The framework of LPLC is described in Fig. 3.7.

In Fig. 3.7, one optimum forward channel of parameter θ is used to characterize the coding of the base layer PEF $\{e_b\}$, and a second optimum forward channel of parameter $\tilde{\theta}$ is used to characterize the coding of the mismatch $\{\psi\}$. Since two MCP loops are included in LPLC, two 3D filters are incorporated in the LPLC codec in Fig. 3.7, $H_b(\Omega)$ and $H_e(\Omega)$. $H_b(\Omega)$ is the 3D filter combining time delay, motion compensation, and spatial filtering for the base layer MCP step, while $H_e(\Omega)$ is included for the enhancement layer MCP step.

Both layers in LPLC use the same motion vectors. For the ease of analysis, we may assume that the two MCP steps also incorporate the same spatial filtering operation. For instance, the optimal spatial filter $F_{\text{opt}}(\Lambda)$ in (3.20), which is approximated by $P(\Lambda)$, may be included by both 3D filters. Hence, we assume $H_b(\Omega)$ and $H_e(\Omega)$ are identical, which will be referred to as $H(\Omega)$ hereafter, and obtain an alternative diagram for LPLC as shown in Fig. 3.8. This alternative diagram is more amenable to the theoretic analysis of LPLC, since the two MCP loops are decoupled

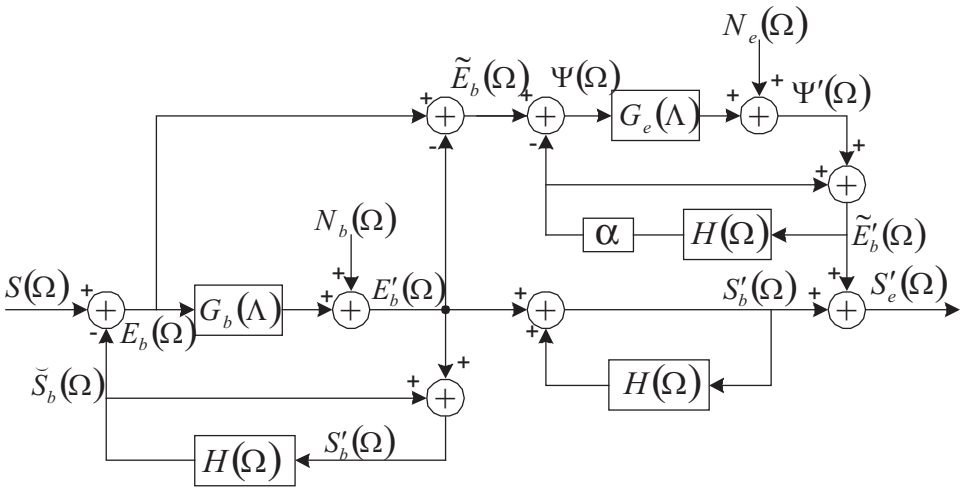


Fig. 3.8. Alternative block diagram of the LPLC codec

from each other and the leaky factor α is only present in the enhancement layer MCP step. The details of the development for the alternative block diagram are given in Appendix B.

Next we assume no error drift in the base layer and develop the rate distortion analysis for two scenarios of LPLC: with and without drift in the enhancement layer.

Scenario I for LPLC: Following a similar manner as the analysis developed for the PSD of the base layer PEF, $\Phi^{\theta}_{e_b e_b}(\Lambda)$, as in (3.24) and (3.25), we obtain an approximation of the PSD of the mismatch signal $\{\psi\}$ as

$$\Phi^{\tilde{\theta}}_{\psi\psi}(\Lambda) \approx \Phi^{appr,\tilde{\theta}}_{\psi\psi}(\Lambda) = \begin{cases} \Phi_{\tilde{e}_b\tilde{e}_b}(\Lambda) & \Lambda : \Phi_{\tilde{e}_b\tilde{e}_b}(\Lambda) \leq \tilde{\theta} \\ \max\{\tilde{\theta}, \Phi^{I,\tilde{\theta}}_{\psi\psi}(\Lambda)\} & \Lambda : \Phi_{\tilde{e}_b\tilde{e}_b}(\Lambda) > \tilde{\theta} \end{cases}, \tag{3.35}$$

where

$$\Phi^{I,\tilde{\theta}}_{\psi\psi}(\Lambda) = \Phi_{\tilde{e}_b\tilde{e}_b}(\Lambda)[1 - 2\alpha\mathrm{Re}\{F(\Lambda)P^*(\Lambda)\} + \alpha^2|F(\Lambda)|^2] + \tilde{\theta}\alpha^2|F(\Lambda)|^2, \tag{3.36}$$

and $\Phi_{\tilde{e}_b\tilde{e}_b}(\Lambda)$ is obtained by (3.27). If $F(\Lambda)$ is chosen as the optimal spatial filter for the base layer, F_{opt}, which is approximated by $P(\Lambda)$ as in (3.20), we have

$$\Phi_{\psi\psi}^{I,\tilde{\theta}}(\Lambda) = \Phi_{\tilde{e}_b\tilde{e}_b}(\Lambda)[1-\alpha(2-\alpha)|P(\Lambda)|^2] + \tilde{\theta}\alpha^2|P(\Lambda)|^2, \text{ for } F(\Lambda) = P(\Lambda). \quad (3.37)$$

Note that only when $\alpha = 1$, $F(\Lambda)\approx P(\Lambda)$ is also approximately optimal for the minimization of $\Phi_{\psi\psi}^{I,\tilde{\theta}}(\Lambda)$ for $\Lambda : \Phi_{\tilde{e}_b\tilde{e}_b}(\Lambda) > \tilde{\theta}$.

According to Fig. 3.8, we have

$$\begin{aligned} S(\Omega) - S'_e(\Omega) &= S(\Omega) - S'_b(\Omega) - (S'_e(\Omega) - S'_b(\Omega)) \\ &= \tilde{E}_b(\Omega) - \tilde{E}'_b(\Omega) = \Psi(\Omega) - \Psi'(\Omega) \triangleq \tilde{\Psi}(\Omega). \end{aligned} \quad (3.38)$$

Analogous to $\Phi_{\tilde{e}_b\tilde{e}_b}(\Lambda)$ in (3.27), we have

$$\Phi_{\tilde{\psi}\tilde{\psi}}(\Lambda) = \min\{\tilde{\theta}, \Phi_{\psi\psi}^{\tilde{\theta}}(\Lambda)\} = \min\{\tilde{\theta}, \Phi_{\tilde{e}_b\tilde{e}_b}(\Lambda)\}. \quad (3.39)$$

Similar to the discussion for *Scenario I for conventional layered video coding* in Subsection 3.2.2, we first assume $\tilde{\theta} \leq \theta$. Using (3.27), we simplify (3.39) as

$$\Phi_{\tilde{\psi}\tilde{\psi}}(\Lambda) = \min\{\tilde{\theta}, \Phi_{ss}(\Lambda)\}, \text{ for } \tilde{\theta} \leq \theta. \quad (3.40)$$

Hence, combining (3.38) and (3.40), we obtain the MSE distortion between the input video signal and the decoded signal by both layers in LPLC as

$$\begin{aligned} D_e^{I,\theta,\tilde{\theta}} = E\{(s-s'_e)^2\} &= \frac{1}{4\pi^2}\iint_\Lambda \Phi_{\tilde{\psi}\tilde{\psi}}(\Lambda)d\Lambda \\ &= \frac{1}{4\pi^2}\iint_\Lambda \min\{\tilde{\theta}, \Phi_{ss}(\Lambda)\}d\Lambda, \\ &\quad \text{for } \tilde{\theta} \leq \theta. \end{aligned} \quad (3.41)$$

Note that the distortion function in (3.41) for LPLC has the same form as that in (3.26) for the conventional layered video coding structure.

The data rate, in units of bits per symbol, consumed by the LPLC codec in Fig. 3.8 is the sum of the mutual information between $\{e_b\}$ and $\{e'_b\}$ plus the mutual information between $\{\psi\}$ and $\{\psi'\}$. Hence,

$$\begin{aligned} R_e^{I,\theta,\tilde{\theta}} &= \frac{1}{8\pi^2}\iint_{\Lambda} \max\left\{0, \log_2\left(\frac{\Phi^{\theta}_{e_b e_b}(\Lambda)}{\theta}\right)\right\} \\ &\quad + \max\left\{0, \log_2\left(\frac{\Phi^{\tilde{\theta}}_{\psi\psi}(\Lambda)}{\tilde{\theta}}\right)\right\} d\Lambda, \\ &\quad \text{for } \tilde{\theta} \le \theta, \end{aligned} \tag{3.42}$$

where $\Phi^{\tilde{\theta}}_{\psi\psi}(\Lambda)$ can be approximately obtained by (3.35) and $\Phi^{\theta}_{e_b e_b}(\Lambda)$ is approximated by (3.24). Note that when $\alpha = 0$, LPLC reduces to the conventional layered coding structure. At this time, $\Phi^{\tilde{\theta}}_{\psi\psi}(\Lambda)$ equals to $\Phi_{\tilde{e}_b\tilde{e}_b}(\Lambda)$ according to (3.35) and (3.36), and the rate function given by (3.42) reduces to the form in (3.23) for conventional layered coding.

When $\tilde{\theta} > \theta$, we have $\Phi^{\tilde{\theta}}_{\psi\psi}(\Lambda) = \Phi_{\tilde{e}_b\tilde{e}_b}(\Lambda) \le \theta < \tilde{\theta}$. It is easy to show that the rate distortion functions for both layers of LPLC, $D_e^{I,\theta,\tilde{\theta}}$ and $R_e^{I,\theta,\tilde{\theta}}$, as given by (3.41) and (3.42), reduce to the rate distortion functions for the base layer that are parameterized by θ, and have exactly the same form as (3.29) and (3.30) for the base layer in conventional layered coding.

Scenario II for LPLC: LPLC was designed to balance coding efficiency and error resilience performance. The use of the leaky factor α, when less than 1, is targeted to mitigate the effect of error propagation that is caused by the truncation or destruction

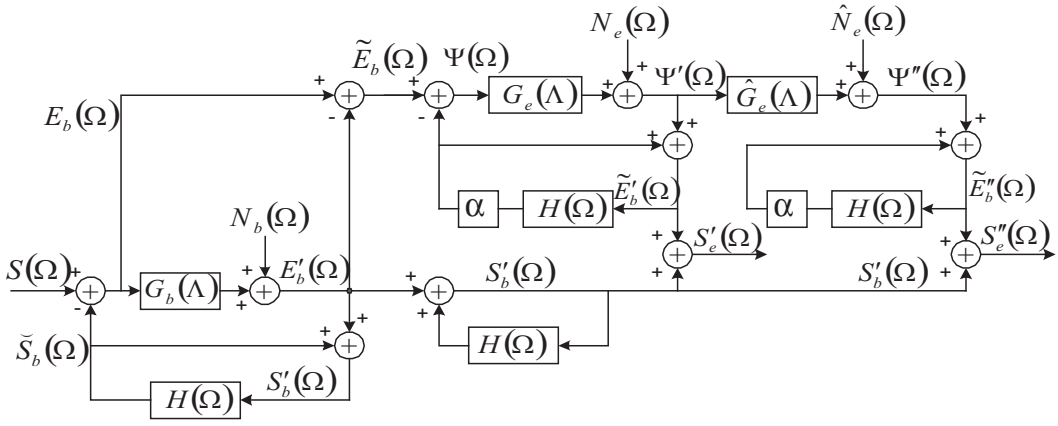


Fig. 3.9. Block diagram of an LPLC codec when the enhancement layer is decoded below the MCP rate

of the enhancement layer in the MCP loop. Next we derive the rate distortion functions for LPLC when drift occurs in the enhancement layer and evaluate the error resilience performance of LPLC with respect to α.

Based on the idea used in *Scenario II for conventional layered video coding* where drift occurs in the base layer MCP loop, as discussed in Subsection 3.2.2, we introduce a third optimum forward channel that takes the encoded mismatch signal $\{\psi'\}$ as its input. We use this channel to model the loss mechanism for prediction drift in the enhancement layer MCP loop, and obtain the framework for LPLC where the mismatch signal is decoded below the MCP rate, as shown in Fig. 3.9. It is not precise to assume that the prediction drift caused by channel noise occur in a rate distortion optimization way. The use of an optimum forward channel, however, provides a way for the theoretic analysis to characterize the effect of prediction error drift.

For the convenience of analysis, we change the parameter that features the optimum forward channel connecting $\{\psi\}$ and its encoded version $\{\psi'\}$ to $\breve{\theta}$, and use $\hat{\theta}$ to denote the parameter for the second optimum forward channel in the enhancement layer. Let $\tilde{\theta} \triangleq \breve{\theta} + \hat{\theta}$. We have

$$\begin{aligned} S(\Omega) - S_e''(\Omega) &= S(\Omega) - S_b'(\Omega) - (S_e''(\Omega) - S_b'(\Omega)) \\ &= \tilde{E}_b(\Omega) - \tilde{E}_b''(\Omega) \\ &= (\tilde{E}_b(\Omega) - \tilde{E}_b'(\Omega)) + (\tilde{E}_b'(\Omega) - \tilde{E}_b''(\Omega)) \\ &= (\Psi(\Omega) - \Psi'(\Omega)) + \frac{1}{1-\alpha H(\Omega)}(\Psi'(\Omega) - \Psi''(\Omega)), \\ &\triangleq \tilde{\Psi}(\Omega) + \frac{1}{1-\alpha H(\Omega)}\hat{\Psi}(\Omega). \end{aligned} \tag{3.43}$$

Since the two terms in the right hand side of (3.43) are uncorrelated with each other[2], we have the MSE distortion function for the LPLC codec in Fig. 3.9, where the enhancement layer is decoded below the MCP rate, as follows

$$D_e^{II,\theta,\tilde{\theta}} = \frac{1}{4\pi^2}\iint_{\Lambda} \Phi_{\tilde{\psi}\tilde{\psi}}(\Lambda) + E\left\{\left|\frac{1}{1-\alpha H(\Omega)}\right|^2\right\}\Phi_{\hat{\psi}\hat{\psi}}(\Lambda)d\Lambda, \tag{3.44}$$

where

$$E\left\{\left|\frac{1}{1-\alpha H(\Omega)}\right|^2\right\} = \frac{1}{1-\alpha^2|F(\Lambda)|^2}, \tag{3.45}$$

which is derived in a similar way as in [58],

$$\Phi_{\tilde{\psi}\tilde{\psi}}(\Lambda) = \min\left\{\breve{\theta}, \Phi_{\psi\psi}^{\breve{\theta}}(\Lambda)\right\}, \tag{3.46}$$

and

$$\Phi_{\hat{\psi}\hat{\psi}}(\Lambda) = \min\left\{\hat{\theta}, \Phi_{\psi'\psi'}(\Lambda)\right\} = \min\left\{\tilde{\theta} - \breve{\theta}, \max\left\{0, \Phi_{\psi\psi}^{\breve{\theta}}(\Lambda) - \breve{\theta}\right\}\right\}. \tag{3.47}$$

[2]This can be proved in a similar way as in [58].

When $F(\Lambda)$ is evaluated as $F_{\text{opt}}(\Lambda) \approx P(\Lambda)$, the MSE distortion $D_e^{II,\theta,\tilde{\theta}}$ in (3.44) becomes

$$\begin{aligned} D_e^{II,\theta,\tilde{\theta}} &= \frac{1}{4\pi^2} \iint_{\Lambda} \min\left\{\breve{\theta}, \Phi_{\psi\psi}^{\breve{\theta}}(\Lambda)\right\} \\ &\quad + \frac{1}{1-\alpha^2|P(\Lambda)|^2} \min\left\{\tilde{\theta} - \breve{\theta}, \max\left\{0, \Phi_{\psi\psi}^{\breve{\theta}}(\Lambda) - \breve{\theta}\right\}\right\} d\Lambda, \\ &\quad \text{for } F(\Lambda) = P(\Lambda). \end{aligned} \tag{3.48}$$

The data rate consumed by the LPLC codec in Fig. 3.9 is the sum of the mutual information between $\{e_b\}$ and $\{e_b'\}$ plus the mutual information between $\{\psi\}$ and $\{\psi''\}$, therefore,

$$\begin{aligned} R_e^{II,\theta,\tilde{\theta}} &= \frac{1}{8\pi^2} \iint_{\Lambda} \max\left\{0, \log_2\left(\frac{\Phi_{e_b e_b}^{\theta}(\Lambda)}{\theta}\right)\right\} \\ &\quad + \max\left\{0, \log_2\left(\frac{\Phi_{\psi\psi}^{\breve{\theta}}(\Lambda)}{\tilde{\theta}}\right)\right\} d\Lambda. \end{aligned} \tag{3.49}$$

Note that in Fig. 3.9, if θ is fixed for the base layer optimum forward channel, and $\breve{\theta} > \theta$, then no information is carried by the enhancement layer. The rate distortion functions in (3.48) and (3.49) reduce to a fixed point that is parameterized by θ, which is the same as that specified by (3.29) and (3.30). The PSD of the output signal from the first channel in the enhancement layer, $\Phi_{\psi'\psi'}(\Lambda)$, is zero, and no drift occurs in the enhancement layer. If the encoded bitstream still suffers from data rate truncation, it will be the base layer that suffers from drift, which yields *Scenario II* in Subsection 3.2.2 where the base layer is decoded below the MCP rate. Therefore, we have $\breve{\theta} \leq \theta$ for modeling the scenario where the base layer is intact while the enhancement layer is likely to suffer from drift.

If $\breve{\theta}$ is further fixed, namely $\breve{\theta}_0$ that satisfies $\breve{\theta}_0 \leq \theta$, the parameter of the second optimum forward channel in the enhancement layer, $\hat{\theta}$, which models the drift that affects the enhancement layer, ranges between 0 and $\theta - \breve{\theta}_0$. When $\hat{\theta} > \theta - \breve{\theta}_0$, i.e., $\tilde{\theta} > \theta$, the rate distortion functions in (3.48) and (3.49) should also reduce to the fixed point specified by (3.29) and (3.30). This is because $\tilde{\theta} = \breve{\theta}_0 + \hat{\theta}$ is the parameter of the equivalent optimum forward channel substituting the two optimum forward channels parameterized by $\breve{\theta}_0$ and $\hat{\theta}$. When $\tilde{\theta} > \theta$, no information is conveyed by the enhancement layer and the rate distortion functions by both layers in *Scenario II for LPLC* should become identical to that by the base layer. This is satisfied, however, in our closed form expressions in (3.48) and (3.49) only when $\alpha = 0$ or $\alpha = 1$. The details of further discussion on *Scenario II for LPLC* are given in Appendix C. Due to the approximation we used in (3.35) for the evaluation of $\Phi^{\breve{\theta}}_{\psi\psi}(\Lambda)$, when $0 < \alpha < 1$, the distortion (3.48) is usually smaller than that in (3.29).

3.3 Rate Distortion Analysis of LPLC Using Quantization Noise Modeling

In this section, we present a different approach to theoretically analyze the rate distortion performance of LPLC by using a quantization noise model that was originally proposed in [72]. This quantization noise model was used in [59, 60] to address the rate distortion analysis of conventional layered video coding. The theoretic analysis we address here for LPLC is an extension of the work in [59, 60].

Leaky prediction can also be used in the non-scalable coding structure, where the reference frame is scaled by a leaky factor α, having a value between 0 and 1, before it is taken by motion compensation. Both intra-frame coding and inter-frame coding can be considered as a special instantiation, where $\alpha = 0$ for intra-frame coding and $\alpha = 1$ for inter-frame coding. In [73, 74], rate distortion analysis of non-scalable video coding using leaky prediction was discussed by modeling the video signal as a first-order Markov model in the temporal direction. Similar to the work in [59, 60], the analysis developed in [73, 74] also uses heuristic models to theoretically analyze the encoding of 2D PEFs generated by the MCP step.

3.3.1 Quantization Noise Modeling for 2D Image Coding

Given a 2D image $\{s\}$, we model the quantization noise $\{q\}$ as an additive, uncorrelated signal whose variance is

$$Var\{q\} = \sigma_q^2 = \sigma_s^2 2^{-\beta R_s}, \tag{3.50}$$

where σ_s^2 denotes the variance of the original signal, β denotes the parameter related to the 2D image coding efficiency, usually taking a value between 0.8 and 1.5, and R_s is the data rate in unit of bits/pixel used to encode $\{s\}$ [72].

As addressed in [72] and [59,60], using the heuristic model in (3.50), the encoding of a 2D image can be modelled as the adding to the image with an uncorrelated noise,

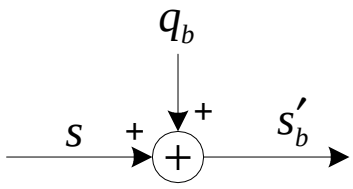


Fig. 3.10. Quantization noise modeling for 2D image coding

as shown in Fig. 3.10. The heuristic model in (3.50) provides a way to directly relate the MSE distortion of the decoded image, D, to the encoding data rate, R, as follows

$$D(R) = E\left[(s - s')^2\right] = Var\{q\} = \sigma_q^2 = \sigma_s^2 2^{-\beta R_s}, \tag{3.51}$$

where $E[X]$ denotes the expectation of a random variable X.

3.3.2 Rate Distortion Functions for LPLC Using Quantization Noise Modeling

To explore the theoretic analysis of LPLC using quantization noise modeling, we still use the alternative block diagram for LPLC shown in Fig. 3.8, where the two 3D filters included in the two MCP loops of LPLC are assumed to be identical. Different than the approach in Section 3.2 which uses the optimum forward channel, we use the quantization noise model in (3.50) to characterize the encoding of 2D images.

The optimum forward channel in Fig. 3.1 is derived from rate distortion theory, thus providing the rate distortion bound at which a 2D stationary, Gaussian random signal is encoded. The quantization noise model in (3.50) is a heuristic model, which was obtained from the operational results. We use the quantization noise model since it provides a different perspective to theoretically explore the rate distortion performance of LPLC. The use of the quantization noise model may result in a closed formulation of the MSE distortion explicitly related to the data rate, which is different from the parametric rate distortion functions specified by the optimum forward channel in Fig. 3.1.

We specify three types of data rates: the data rate used by the base layer, R_b, the minimum data rate used by both layers, $R_{e,\min}$, and the maximum data rate used by both layers, $R_{e,\max}$, which satisfy $R_b < R_{e,\min} < R_{e,\max}$. As we mentioned, an MCP rate is defined as the data rate that is incorporated in the MCP step. Thus, R_b is the MCP rate in the base layer, and $(R_{e,\min} - R_b)$ is the MCP rate in the enhancement layer. We consider two scenarios, with and without drift in the enhancement layer, and assume no drift in the base layer for both scenarios. If let $R_{e,\mathrm{dec}}$ denote the decoded data rate, the above two scenarios correspond to the circumstances where $R_b \le R^I_{e,\mathrm{dec}} < R_{e,\min}$ and $R_{e,\min} \le R^{II}_{e,\mathrm{dec}} \le R_{e,\max}$ respectively.

A. The Rate Distortion Function for the Base Layer

Using the quantization noise model in (3.50) and the block diagram in Fig. 3.8, we obtain the block diagram of LPLC when the enhancement layer is decoded above the MCP rate in Fig. 3.11.

From Fig. 3.11, we have

$$e'_b = e_b + q_b, \tag{3.52}$$

where $\{q_b\}$ denotes the quantization noise introduced in the encoding of the base layer PEF $\{e_b\}$ at the MCP data rate R_b. The variance of $\{q_b\}$ is

$$\sigma^2_{q_b} = \sigma^2_{e_b} 2^{-\beta R_b}, \tag{3.53}$$

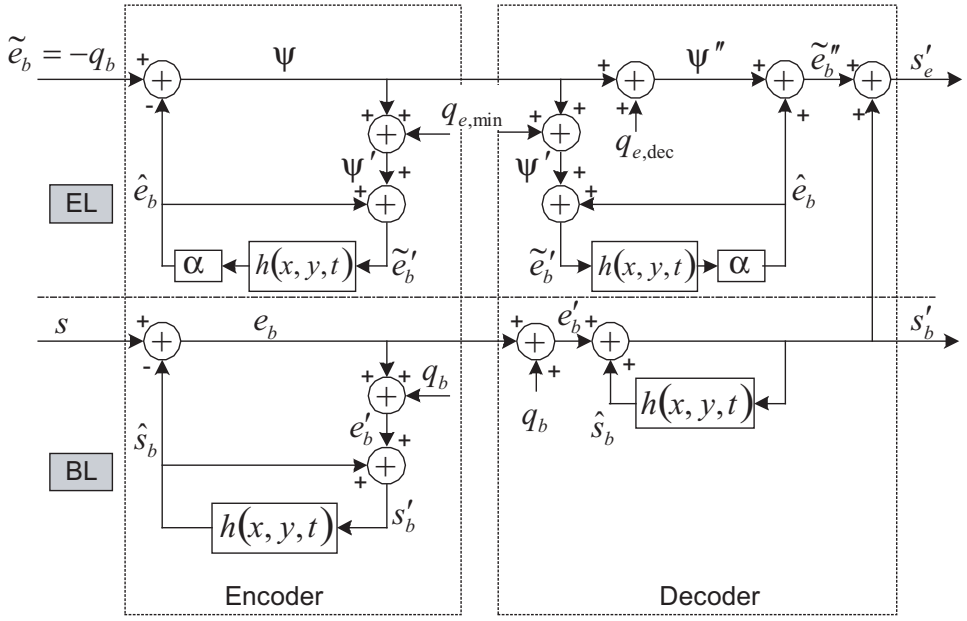


Fig. 3.11. Block diagram of an LPLC codec when the enhancement layer is decoded above the MCP rate (using quantization noise modeling)

where $\sigma^2_{e_b}$ denotes the variance of $\{e_b\}$. Let $\Phi_{e_b e_b}(\Lambda)$ denote the 2D PSD of $\{e_b\}$. Similar to the derivation for the equation (3.25) [64], we derive

$$\begin{aligned}\Phi_{e_b e_b}(\Lambda) &= \Phi_{ss}(\Lambda)\left[1 - 2\mathrm{Re}\{F(\Lambda)P^*(\Lambda)\} + |F(\Lambda)|^2\right] \\ &\quad + \Phi_{q_b q_b}(\Lambda)|F(\Lambda)|^2, \end{aligned} \tag{3.54}$$

where $\Phi_{ss}(\Lambda)$ and $\Phi_{q_b q_b}(\Lambda)$ denote the 2D PSD of $\{s\}$ and $\{q_b\}$ respectively. As opposed to $\Phi^{I,\theta}_{e_b e_b}(\Lambda)$ in (3.25) that was derived from the use of the optimum forward channel, $\Phi_{e_b e_b}(\Lambda)$ in (3.54) is a universal from for the PSD of the base layer PEF $\{e_b\}$.

If all the quantization noise is assumed to be white, we have

$$\Phi_{q_b q_b}(\Lambda) = \sigma^2_{q_b}. \tag{3.55}$$

Since

$$\sigma^2_{e_b} = \frac{1}{4\pi^2}\iint_{\Lambda} \Phi_{e_b e_b}(\Lambda) d\Lambda, \tag{3.56}$$

combining (3.54), (3.55), and (3.56), we have

$$\sigma^2_{e_b} = \frac{\theta_s}{1 - 2^{-\beta R_b}\theta_f}, \tag{3.57}$$

and

$$\sigma^2_{q_b} = \frac{\theta_s}{1 - 2^{-\beta R_b}\theta_f} 2^{-\beta R_b} = \frac{\theta_s}{2^{\beta R_b} - \theta_f}, \tag{3.58}$$

where

$$\theta_s \triangleq \frac{1}{4\pi^2}\iint_{\Lambda} \Phi_{ss}(\Lambda)\left[1 - 2\mathrm{Re}\{F(\Lambda)P^*(\Lambda)\} + |F(\Lambda)|^2\right] d\Lambda, \tag{3.59}$$

$$\theta_f \triangleq \frac{1}{4\pi^2}\iint_{\Lambda} |F(\Lambda)|^2 d\Lambda. \tag{3.60}$$

From Fig. 3.11, the reconstruction error of the base layer is

$$r_b = s'_b - s = e'_b - e_b = q_b. \tag{3.61}$$

Hence the MSE distortion of the base layer, as a function of the base layer data rate, is

$$D_b(R_b) = Var\{r_b\} = \sigma^2_{q_b} = \frac{\theta_s}{2^{\beta R_b} - \theta_f} \triangleq \sigma^2_b. \tag{3.62}$$

The signal-to-noise ratio (SNR) of the base layer in dB is

$$\mathrm{SNR}_b(R_b) = 10\log_{10}\left(\frac{\sigma^2_s}{\sigma^2_b}\right) = 10\log_{10}\left(\frac{2^{\beta R_b} - \theta_f}{\theta_s}\sigma^2_s\right). \tag{3.63}$$

B. The Rate Distortion Function for the Enhancement Layer

Scenario I for LPLC: The enhancement layer of LPLC is decoded above the MCP rate, namely $R_{e,\min} \leq R^I_{e,\mathrm{dec}} \leq R_{e,\max}$. In this scenario, no error drift occurs in the enhancement layer.

As shown in Fig. 3.11, the input to the enhancement layer MCP loop in LPLC is $\{\tilde{e}_b\}$, the residue between the original video signal and the reconstruction by the base layer. Using (3.52), we have

$$\tilde{e}_b = s - s'_b = e_b - e'_b = -q_b. \tag{3.64}$$

The mismatch signal is

$$\psi = \tilde{e}_b - \hat{e}_b, \tag{3.65}$$

which is encoded to $\{\psi'\}$, where

$$\psi' = \psi + q_{e,\min}, \tag{3.66}$$

and $\{q_{e,\min}\}$ denotes the quantization noise introduced in the encoding of the mismatch at the enhancement layer MCP data rate $(R_{e,\min} - R_b)$. The variance of $\{q_{e,\min}\}$ is

$$\sigma^2_{q_{e,\min}} = \sigma^2_{\psi} 2^{-\beta(R_{e,\min}-R_b)}, \tag{3.67}$$

where σ^2_{ψ} denotes the variance of $\{\psi\}$. Similar to (3.54), we derive the 2D PSD of the mismatch signal using (3.64) as follows

$$\begin{aligned} \Phi_{\psi\psi}(\Lambda) &= \Phi_{q_b q_b}(\Lambda)\left[1 - 2\alpha \mathrm{Re}\{F(\Lambda)P^*(\Lambda)\} + \alpha^2 |F(\Lambda)|^2\right] \\ &\quad + \alpha^2 \Phi_{q_{e,\min} q_{e,\min}}(\Lambda)|F(\Lambda)|^2, \end{aligned} \tag{3.68}$$

where $\Phi_{q_{e,\min} q_{e,\min}}(\Lambda)$ is the PSD of $\{q_{e,\min}\}$ and

$$\Phi_{q_{e,\min} q_{e,\min}}(\Lambda) = \sigma^2_{q_{e,\min}}. \tag{3.69}$$

Since

$$\sigma^2_{\psi} = \frac{1}{4\pi^2} \iint_{\Lambda} \Phi_{\psi\psi}(\Lambda) d\Lambda, \tag{3.70}$$

combining (3.67) through (3.70), we have

$$\sigma^2_{\psi} = \frac{1 - 2\alpha\theta_{fp} + \alpha^2\theta_f}{1 - 2^{-\beta(R_{e,\min}-R_b)}\alpha^2\theta_f} \sigma^2_{q_b}, \tag{3.71}$$

and

$$\begin{aligned} \sigma^2_{q_{e,\min}} &= \frac{\sigma^2_{q_b}(1 - 2\alpha\theta_{fp} + \alpha^2\theta_f)}{1 - 2^{-\beta(R_{e,\min}-R_b)}\alpha^2\theta_f} 2^{-\beta(R_{e,\min}-R_b)} \\ &= \frac{1 - 2\alpha\theta_{fp} + \alpha^2\theta_f}{2^{\beta(R_{e,\min}-R_b)} - \alpha^2\theta_f} \sigma^2_{q_b}, \end{aligned} \tag{3.72}$$

where $\sigma^2_{q_b}$ is obtained by (3.58), θ_f by (3.60), and

$$\theta_{fp} \triangleq \frac{1}{4\pi^2} \iint_{\Lambda} \mathrm{Re}\{F(\Lambda)P^*(\Lambda)\} d\Lambda. \tag{3.73}$$

At the encoder in Fig. 3.11, $\{\tilde{e}_b\}$ is reconstructed as $\{\tilde{e}'_b\}$. At the decoder, since no drift occurs in the enhancement layer, an identical MCP step is included and hence the same MCP signal $\{\hat{e}_b\}$ attained. To reconstruct $\{\tilde{e}_b\}$, however, a different quantization procedure might be used, where the quantization noise is $\{q_{e,\mathrm{dec}}\}$. The noise signal $\{q_{e,\mathrm{dec}}\}$ models the distortion by the coding of the mismatch at the decoding data rate $(R^I_{e,\mathrm{dec}} - R_b)$, and has a variance as

$$\sigma^{2(I)}_{q_{e,\mathrm{dec}}} = \sigma^2_\psi 2^{-\beta(R^I_{e,\mathrm{dec}} - R_b)}, \tag{3.74}$$

where σ^2_ψ is given by (3.71). Note that

$$\sigma^{2(I)}_{q_{e,\mathrm{dec}}} \leq \sigma^2_{q_{e,\min}}, \tag{3.75}$$

since $R^I_{e,\mathrm{dec}} \geq R_{e,\min}$. This results in a second quantized version of the mismatch signal,

$$\psi'' = \psi + q_{e,\mathrm{dec}}, \tag{3.76}$$

and $\{\tilde{e}_b\}$ is reconstructed as

$$\tilde{e}''_b = \psi'' + \hat{e}_b. \tag{3.77}$$

The decoded video signal from both layers is then obtained as

$$s'_e = s'_b + \tilde{e}''_b. \tag{3.78}$$

The reconstruction error of the enhancement layer is

$$r^I_e = s'_e - s = (s'_e - s'_b) + (s'_b - s) = \tilde{e}''_b + r_b. \tag{3.79}$$

Since

$$\tilde{e}''_b - \tilde{e}_b = \psi'' - \psi = q_{e,\mathrm{dec}}, \tag{3.80}$$

we have

$$r_e^I = \tilde{e}_b + q_{e,\mathrm{dec}} + r_b = -q_b + q_{e,\mathrm{dec}} + q_b = q_{e,\mathrm{dec}}. \tag{3.81}$$

Hence the MSE distortion of the enhancement layer, as a function of the three types of data rates we specified, is

$$\begin{aligned} D_e^I(R_b, R_{e,\min}, R_{e,\mathrm{dec}}^I) &= Var\left\{r_e^I\right\} = \sigma_{q_{e,\mathrm{dec}}}^{2(I)} \\ &= \frac{\sigma_{q_b}^2(1 - 2\alpha\theta_{fp} + \alpha^2\theta_f)}{1 - 2^{-\beta(R_{e,\min}-R_b)}\alpha^2\theta_f} 2^{-\beta(R_{e,\mathrm{dec}}^I - R_b)} \\ &= \frac{1 - 2\alpha\theta_{fp} + \alpha^2\theta_f}{2^{\beta(R_{e,\min}-R_b)} - \alpha^2\theta_f} 2^{-\beta(R_{e,\mathrm{dec}}^I - R_{e,\min})} D_b(R_b) \\ &\triangleq \sigma_e^{2(I)}. \end{aligned} \tag{3.82}$$

The SNR of the enhancement layer in dB then is

$$\mathrm{SNR}_e^I(R_b, R_{e,\min}, R_{e,\mathrm{dec}}^I) = 10\log_{10}\left(\frac{\sigma_s^2}{\sigma_e^{2(I)}}\right). \tag{3.83}$$

Scenario II for LPLC: The enhancement layer of LPLC is decoded below the MCP rate, namely $R_b \leq R_{e,\mathrm{dec}}^{II} < R_{e,\min}$. In this scenario, prediction error drift occurs in the enhancement layer.

As shown in Fig. 3.12, since drift occurs to the enhancement layer, the signal applied to the MCP loop at the decoder is no longer the same as that at the encoder. The reconstruction error of the enhancement layer is

$$r_e^{II} = s_e' - s = \tilde{e}_b'' + q_b. \tag{3.84}$$

We have

$$\tilde{e}_b'' = \psi'' * h_d^\alpha = -q_b + q_{e,\min} + \Delta q_{e,\mathrm{dec}} * h_d^\alpha, \tag{3.85}$$

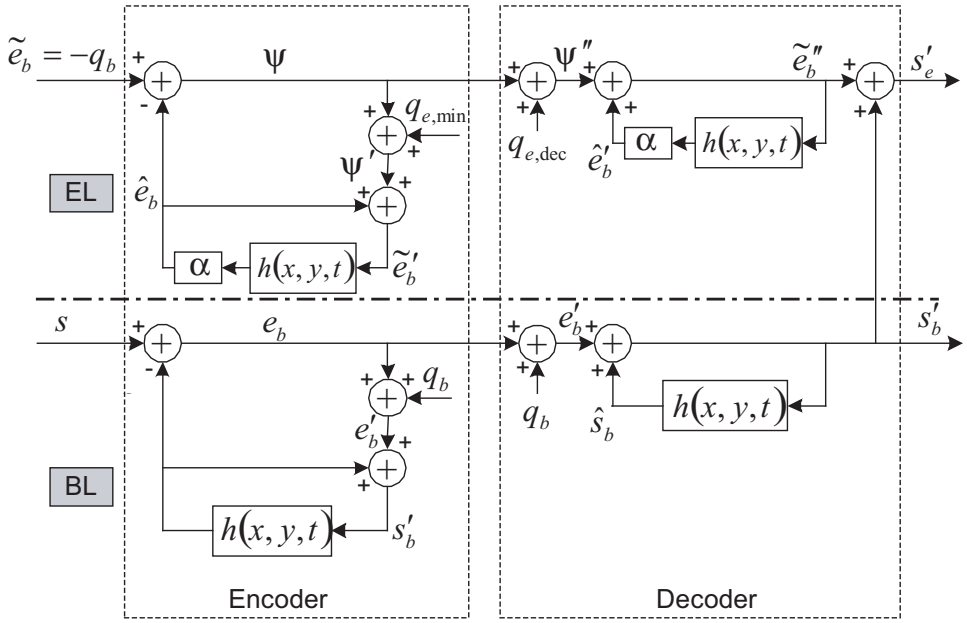
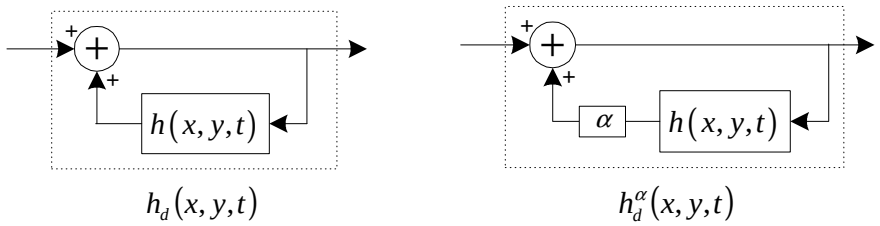


Fig. 3.12. Block diagram of an LPLC codec when the enhancement layer is decoded below the MCP rate (using quantization noise modeling)



(a) base layer (b) enhancement layer

Fig. 3.13. Equivalent filters for the decoder MCP steps in LPLC

where

$$\Delta q_{e,\mathrm{dec}} \triangleq q_{e,\mathrm{dec}} - q_{e,\min}, \tag{3.86}$$

and $\{*\}$ denotes the convolution operation. Here the quantization noise signal $\{q_{e,\mathrm{dec}}\}$ models the distortion by the coding of the mismatch at the decoding data rate $(R^{II}_{e,\mathrm{dec}} - R_b)$, and has a variance as

$$\sigma^{2(II)}_{q_{e,\mathrm{dec}}} = \sigma^2_{\psi} 2^{-\beta(R^{II}_{e,\mathrm{dec}} - R_b)}. \tag{3.87}$$

$\{h^{\alpha}_d\}$ is the impulsive response signal of the equivalent filter representing the enhancement layer MCP loop at the decoder, as shown in Fig 3.13(b). The corresponding transform function is represented by

$$H^{\alpha}_d(\Omega) \triangleq \frac{1}{1 - \alpha H(\Omega)}. \tag{3.88}$$

Thus we have

$$r^{II}_e = q_{e,\min} + \Delta q_{e,\mathrm{dec}} * h^{\alpha}_d. \tag{3.89}$$

Under the assumptions that uniform embedded quantization operations are used in the enhancement layer, and drift occurs as a result of the truncation to the bit-stream of the enhancement layer, the white signals $\{q_{e,\min}\}$ and $\{\Delta q_{e,\mathrm{dec}}\}$ are approximately uncorrelated with each other, and the variance of $\{\Delta q_{e,\mathrm{dec}}\}$ is approximated by

$$\sigma^2_{\Delta q_{e,\mathrm{dec}}} \approx \sigma^{2(II)}_{q_{e,\mathrm{dec}}} - \sigma^2_{q_{e,\min}}. \tag{3.90}$$

The details of above are given in Appendix D.

Using the results in [58, 65] and [61], we have the variance of $\left\{r_e^{II}\right\}$ as

$$\begin{aligned} Var\left\{r_e^{II}\right\} &= \sigma^2_{q_{e,\min}} + \sigma^2_{\Delta q_{e,\mathrm{dec}}} \left(\frac{\Delta t}{8\pi^3} \iiint_{\Omega} E\left[\left| H_d^{\alpha}(\Omega) \right|^2 \right] d\Omega \right) \qquad (3.91) \\ &= \sigma^2_{q_{e,\min}} + \sigma^2_{\Delta q_{e,\mathrm{dec}}} \left(\frac{1}{4\pi^2} \iint_{\Lambda} \frac{\Delta t}{2\pi} \int_{\omega_t=0}^{2\pi/\Delta t} E\left[\left| \frac{1}{1-\alpha H(\Omega)} \right|^2 \right] d\omega_t d\Lambda \right) \\ &= \sigma^2_{q_{e,\min}} + \sigma^2_{\Delta q_{e,\mathrm{dec}}} \left(\frac{1}{4\pi^2} \iint_{\Lambda} \frac{1}{1-\alpha^2 |F(\Lambda)|^2} d\Lambda \right) \\ &= \sigma^2_{q_{e,\min}} + \sigma^2_{\Delta q_{e,\mathrm{dec}}} \theta_d^{\alpha}, \qquad (3.92) \end{aligned}$$

where

$$\theta_d^{\alpha} \triangleq \frac{1}{4\pi^2} \iint_{\Lambda} \frac{1}{1-\alpha^2 |F(\Lambda)|^2} d\Lambda. \tag{3.93}$$

Hence, the MSE distortion of the enhancement layer, as a function of the three types of data rates we specified, is

$$\begin{aligned}
& D_e^{II}(R_b, R_{e,\min}, R_{e,\mathrm{dec}}^{II}) = Var\left\{r_e^{II}\right\} = \sigma_{q_{e,\min}}^2 + \sigma_{\Delta q_{e,\mathrm{dec}}}^2 \theta_d^\alpha \\
= \;& \sigma_{q_{e,\min}}^2 + \left(\sigma_{q_{e,\mathrm{dec}}}^{2(II)} - \sigma_{q_{e,\min}}^2\right)\theta_d^\alpha \\
= \;& \sigma_\psi^2 2^{-\beta(R_{e,\min}-R_b)} + \left(\sigma_\psi^2 2^{-\beta(R_{e,\mathrm{dec}}^{II}-R_b)} - \sigma_\psi^2 2^{-\beta(R_{e,\min}-R_b)}\right)\theta_d^\alpha \\
= \;& \sigma_\psi^2 2^{-\beta(R_{e,\min}-R_b)}\left(1 + \left(2^{\beta(R_{e,min}-R_{e,\mathrm{dec}}^{II})} - 1\right)\theta_d^\alpha\right) \\
= \;& \frac{\sigma_{q_b}^2\left(1 - 2\alpha\theta_{fp} + \alpha^2\theta_f\right)}{2^{\beta(R_{e,\min}-R_b)} - \alpha^2\theta_f}\left(1 + \left(2^{\beta(R_{e,\min}-R_{e,\mathrm{dec}}^{II})} - 1\right)\theta_d^\alpha\right) \\
= \;& \frac{1 - 2\alpha\theta_{fp} + \alpha^2\theta_f}{2^{\beta(R_{e,\min}-R_b)} - \alpha^2\theta_f}\left(1 + \left(2^{\beta(R_{e,\min}-R_{e,\mathrm{dec}}^{II})} - 1\right)\theta_d^\alpha\right) D_b(R_b) \\
\triangleq \;& \sigma_e^{2(II)}.
\end{aligned} \tag{3.94}$$

The SNR of the enhancement layer in dB then is

$$\mathrm{SNR}_e^{II}(R_b, R_{e,\min}, R_{e,\mathrm{dec}}^{II}) = 10\log_{10}\left(\frac{\sigma_s^2}{\sigma_e^{2(II)}}\right). \tag{3.95}$$

3.4 Evaluation of LPLC Rate Distortion Functions

3.4.1 Rate Distortion Performance of LPLC from Theoretic Results

Similar to [64] and [58], we model the PSD of the input video signal as

$$\Phi_{ss}(\Lambda) = \Phi_{ss}(\omega_x, \omega_y) = \begin{cases} \frac{2\pi}{\omega_0^2}\left(1 + \frac{\omega_x^2+\omega_y^2}{\omega_0^2}\right)^{-3/2} & |\omega_x| \le \pi f_{sx} \text{ and } |\omega_y| \le \pi f_{sy} \\ 0 & \text{otherwise} \end{cases}, \tag{3.96}$$

where f_{sx} and f_{sy} denote the sampling frequencies when the input video signal $\{s\}$ is spatially sampled at the Nyquist rate. We choose ω_0 as

$$\omega_0 = \frac{\pi f_{sx}}{42.19} = \frac{\pi f_{sy}}{46.15}, \tag{3.97}$$

which corresponds to a horizontal and vertical correlation of 0.928 and 0.934, respectively, and well matches up the model in (3.96) with real video signals of this format. Also, we constrain f_{sx} and f_{sy} to satisfy

$$f_{sx} f_{sy} = 1 \text{ pixels/(unit length)}^2, \tag{3.98}$$

and measure the data rate in bits/pixel.

We model the characteristic function of the motion vector estimation error as

$$P(\Lambda) = \exp\left[-\frac{\sigma_{\Delta d}^2}{2}\Lambda \cdot \Lambda\right] = \exp\left[-\frac{\sigma_{\Delta d}^2}{2}(\omega_x^2 + \omega_y^2)\right], \tag{3.99}$$

where $\sigma_{\Delta d}^2$ denotes the variance of the motion vector estimation error in (3.17). Moreover, we choose the spatial filter $F(\Lambda)$ in both MCP steps in LPLC as $P(\Lambda)$ when evaluating the theoretic results.

A. Evaluation of Rate Distortion Functions of LPLC Using Rate Distortion Theory

As described in Fig. 3.14, we evaluate the rate distortion performance of LPLC with respect to the leaky factor, α, according to the closed form expressions we derived using rate distortion theory for the two scenarios of LPLC in Section 3.2.4. Rather than the MSE distortion, we use SNR to evaluate the decoded video quality.

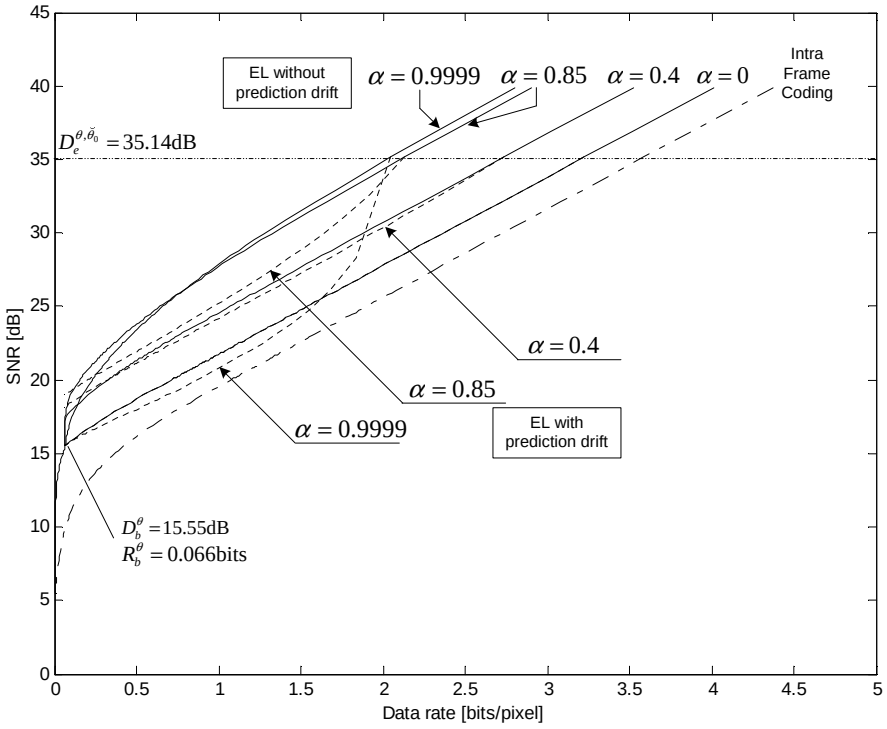


Fig. 3.14. Rate distortion functions of LPLC using rate distortion theory for various leaky factors (α) ($\sigma^2_{\Delta d} = 0.04$ for $P(\Lambda)$)

Results of *Scenario I for LPLC* are shown in solid lines in Fig. 3.14, where the enhancement layer (denoted as EL in the figure) is decoded above the MCP rate and hence does not suffer from drift. When θ is fixed at the point that yields the rate distortion function at $D_b^\theta = 15.55$ dB in (3.29) and $R_b^\theta = 0.066$ bits/pixel in (3.30), we vary the parameter $\tilde{\theta}$ between an extraordinarily small value and θ to obtain the rate distortion curves for different values of α according to (3.41) and (3.42).

It is shown that when the leaky factor α takes on a specific value between 0 and 1, the decoded video quality increases with the increase of the data rate. When the data rate is sufficiently large, LPLC achieves better performance in the rate distortion sense, or in coding efficiency (at a fixed distortion), with increasing leaky factor. To obtain the same amount of distortion, a larger leaky factor requires less data rate than the smaller ones. This implies that $\alpha = 1$ is always optimal in terms of the error-free rate distortion performance. For example, to obtain the distortion $D_e^{I,\theta,\tilde{\theta}} = 35.14$ dB as shown in the figure, LPLC has a gain of approximately 0.12 bits/pixel in rate whenever α increases by 0.1.

It is interesting to note that when the enhancement layer MCP rate is small, it might be possible that a larger leaky factor yields a less efficient codec, especially when the leaky factor is close to 1. In our theoretic results, this arises because the PSD in (3.37) can be rewritten as

$$\begin{aligned} \Phi_{\psi\psi}^{I,\tilde{\theta}}(\Lambda) &= (\Phi_{\tilde{e}_b\tilde{e}_b}(\Lambda) - \tilde{\theta}) \left[1 - \alpha(2-\alpha)|P(\Lambda)|^2\right] \\ &\quad + \tilde{\theta} \left[1 - 2\alpha(1-\alpha)|P(\Lambda)|^2\right]. \end{aligned} \tag{3.100}$$

For a fixed Λ, the first term of $\Phi_{\psi\psi}^{I,\tilde{\theta}}(\Lambda)$ achieves its minimum with respect to α when $\alpha = 1$ while the second term achieves its minimum when $\alpha = \frac{1}{2}$. When both θ and $\tilde{\theta}$ are fixed, the distortion in (3.41) is fixed, but the second term in the integrand of the data rate in (3.42) is a function of $\Phi_{\psi\psi}^{I,\tilde{\theta}}(\Lambda)$, which further relates to α as in (3.100). Hence, $\alpha = 1$ does not always minimize the data rate at a specific distortion.

As expected, when $\tilde{\theta}$ increases to θ, all the rate distortion curves converge to the point specified by θ regardless of the leaky factor.

Results of *Scenario II for LPLC* are shown by dotted lines in Fig. 3.14, where the enhancement layer suffers from data rate truncation. We choose the same θ as above when we evaluate *Scenario I for LPLC*. We also fix $\breve{\theta} = \breve{\theta}_0$. We choose the value for $\breve{\theta}_0$ so that when no drift occurs in the enhancement layer, i.e., $\tilde{\theta} = \breve{\theta} = \breve{\theta}_0$, the distortion in SNR is at 35.14 dB. We then vary $\tilde{\theta}$ between $\breve{\theta}_0$ and θ in (3.48) and (3.49).

It is observed that larger leaky factors yield a larger drop in the rate distortion performance when drift occurs in the enhancement layer. In our closed form expressions, the term $\frac{1}{1-\alpha^2|P(\Lambda)|^2}$ in (3.48) stands for the effect of error propagation when drift occurs. The larger α, the larger decrease in fidelity as a result of the amplification of the drift by this term, implying poor error resilience performance.

When $\tilde{\theta} \geq \theta$, no information is conveyed by the enhancement layer. Hence, when $\tilde{\theta}$ approaches θ from below, the rate distortion curves representing the drift scenario should also converge to the point specified by θ. As discussed in Section 3.2.4, due to the approximation of the PSD of the mismatch, this convergence only occurs when

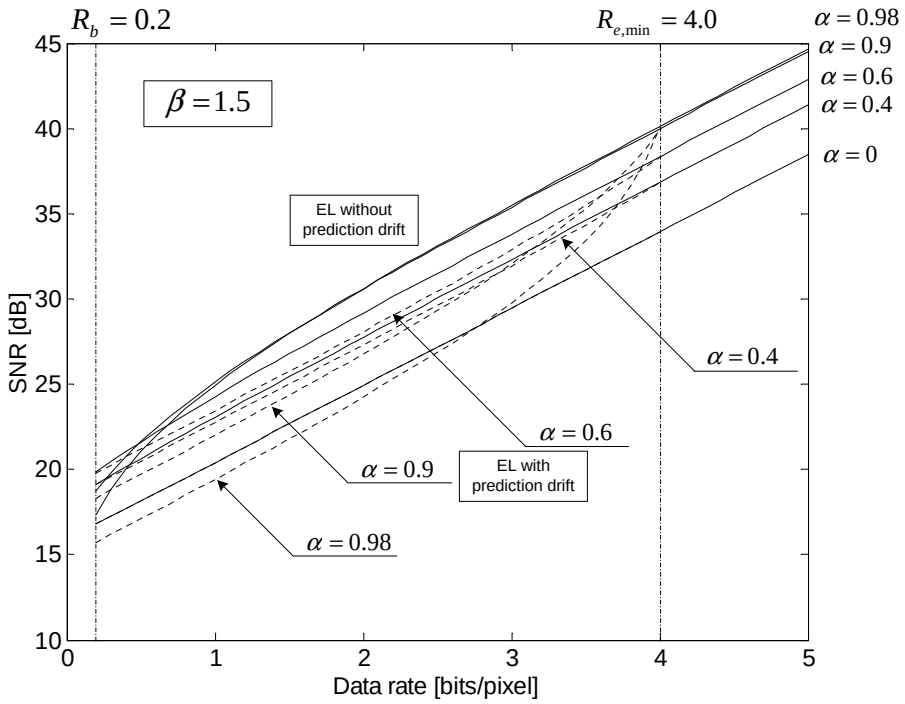


Fig. 3.15. Rate distortion functions of LPLC using quantization noise modeling for various leaky factors (α) ($\sigma^2_{\Delta d} = 0.04$ for $P(\Lambda)$)

$\alpha = 0$ or $\alpha = 1$. When α takes on a value in between, the rate distortion curve usually converges to a point above that specified by θ in the SNR-Rate plane, as shown by the dotted lines in Fig. 3.14.

B. Evaluation of Rate Distortion Functions of LPLC Using Quantization Noise Modeling

As shown in Fig. 3.15, we evaluate the rate distortion performance of LPLC with respect to the leaky factor α according to the closed form expressions we derived using quantization noise modeling for the two scenarios of LPLC in Section 3.3.2.

Results of *Scenario I for LPLC* are shown in solid lines in Fig. 3.15, where the enhancement layer (denoted as EL in the figure) does not suffer from drift in LPLC. From (3.82) and (3.83), we notice that when R_b and $R_{e,\min}$ as well as the leaky factor are fixed, SNR_e^I linearly increases with $R_{e,\mathrm{dec}}^I$. We then let $R_{e,\mathrm{dec}}^I = R_{e,\min}$ in (3.82), i.e., $\tilde{e}_b' = \tilde{e}_b''$, and vary $R_{e,\min}$ between R_b and $R_{e,\max}$, i.e. vary the enhancement layer MCP rate between 0 and $(R_{e,\max} - R_b)$, as described by the solid lines in Fig. 3.15.

At a specific $R_{e,\min}$, we derive the optimal leaky factor to minimize D_e^I in (3.82), i.e. to maximize SNR_e^I in (3.83), as follows

$$\alpha_{\mathrm{opt}} = \frac{\gamma + 1 - \sqrt{(\gamma+1)^2 - 4\gamma\theta_f}}{2\theta_f}, \tag{3.101}$$

where $\gamma = 2^{\beta(R_{e,\min} - R_b)}$. Note that α_{opt} is a function of the MCP rate in the enhancement layer, namely $(R_{e,\min} - R_b)$. When $R_{e,\min}$ is sufficiently large, LPLC achieves a better performance in the rate distortion sense with increasing leaky factor. A larger leaky factor results in a better decoded quality at the same data rate. For example, when $R_{e,\min} = 5$ bits/pixel, SNR_e^I obtains a gain of 3 dB by increasing α from 0 to 0.4, or from 0.4 to 0.9. Notice that when the enhancement layer MCP rate is small, α_{opt} will be far smaller than 1, implying that a larger leaky factor might yield a less efficient codec, especially when the leaky factor is close to 1.

The result in (3.101) implies that the optimal leaky factor to obtain the best rate distortion performance in LPLC is closely related to the enhancement layer MCP rate, which is consistent with that obtained from our first theoretic analysis approach using rate distortion theory. It is seen that by using heuristic quantization

noise modeling, we derive a closed from expression for the optimal leaky factor in terms of the error-free rate distortion performance, as in (3.101).

Results of *Scenario II for LPLC* are shown by dotted lines in Fig. 3.15, where the enhancement layer suffers from data rate truncation and is decoded below the MCP rate. We fix $R_{e,\min} = 4.0$ while vary $R^{II}_{e,\text{dec}}$ between R_b and $R_{e,\min}$ according to (3.94) and (3.95). It is observed that larger leaky factors yield a larger drop in the rate distortion performance when drift occurs in the enhancement layer, which is consistent with the theoretic results of the first approach using rate distortion theory. In our closed form expressions, the term θ_d^α in (3.94) stands for the effect of error propagation when drift occurs. From (3.93), when α approaches 1, we have $\theta_d^\alpha \gg 1$. Since $R^{II}_{e,\text{dec}} < R_{e,\min}$ in the error drift scenario for LPLC, the term $(2^{\beta(R_{e,\min} - R^{II}_{e,\text{dec}})} - 1)\theta_d^\alpha$ in (3.94) greatly amplifies the distortion with larger leaky factors.

We also evaluated our closed-form expressions with different choices for the parameter β and the three data rates. We varied β between 0.8 and 1.5 as suggested in [72], and the base layer data rate R_b between 0.05 and 1.0. These rate distortion curves present similar performance to that shown in Fig. 3.15.

3.4.2 Comparison to Operational Results

In this subsection, we use a wavelet based fully rate scalable hybrid video codec, namely SAMCoW (Scalable Adaptive Motion Compensated Wavelet) [5], to implement LPLC and obtain the operational results. We simulate both scenarios, with

and without drift in the enhancement layer, and evaluate the operational rate distortion performance of LPLC associated with various leaky factors. We chose three video sequences that contain varying degrees of motions in our experiments: *foreman*, *coastguard*, and *mother-daughter* (as *mthrdghtr* in short in the figure), all in QCIF YUV 4:2:0 format. We used the peak signal-to-noise ratio (PSNR) to measure the distortion. We intra-coded the first frame of each sequence and inter-coded all successive frames at a frame rate of 10 fps.

In our LPLC implementation using SAMCoW, the data rate allocated to the base layer is R_B, the data rate to the enhancement layer is R_E, and the total data rate is $R_T = R_B + R_E$. Specifically, R_B is the base layer MCP rate, and R_E is the enhancement layer MCP rate. For inter frames, the base layer contains the embedded bitstream of the motion vectors at a data rate of R_{MV} and the embedded bitstream of the base layer PEFs at a data rate of $(R_B - R_{\mathrm{MV}})$. The enhancement layer carries the embedded bitstream of the mismatch at a data rate of R_E. For the intra frame, both layers contain the embedded bitstream of the original frame at the respective data rate requirements, where the base layer includes the more significant bit planes and the enhancement layer carries the refinement bit planes.

Examples of the operational rate distortion performance of LPLC using SAMCoW are given in Fig. 3.16, Fig. 3.17, and Fig. 3.18, where the leaky factor takes on three values: 0, 0.5, and 0.98.

For the scenario where no drift occurs in the enhancement layer of LPLC, as shown by the solid lines in the figures, we decode the mismatch carried by

the enhancement layer at the MCP rate R_E. We fixed the base layer MCP rate R_B at 50 kbps, and vary R_E between 0 and 200 kbps, i.e. vary the total data rate R_T between 50 kbps and 250 kbps. This corresponds to encoding the base layer at 0.20 bits/pixel, and encoding the entire video sequence between 0.20 bits/pixel and 0.99 bits/pixel. (If the data rate is 50 kbps, for example, since the frame rate is 10 fps and the frame size for QCIF videos is 176×144, we have $50\times1000/(10\times176\times144)\approx0.20$ bits/pixel.)

If no prediction drift occurs in the enhancement layer, it is observed that the decoded quality by both layers in LPLC, denoted as "EL" in the figure, increases with the increase of the total data rate R_T for a specific leaky factor. The optimal leaky factor that yields the best decoded quality for a certain data rate varies across different data rates. When R_T is sufficiently large, a larger leaky factor is always beneficial in obtaining a better decoded quality, implying that the optimal leaky factor is $\alpha = 1$. The gain in the decoded quality, as a result of the increase of the leaky factor, varies across different video characteristics. For example, when $R_T = 250$ kbps, the decoded quality in PSNR increases by roughly 1 dB for all three video sequences if the leaky factor α increases from 0 to 0.5. But increasing α from 0.5 to 0.98 results in a marginal gain in PSNR, less than 0.4 dB, for videos that contain a larger amount of motions such as *foreman* and *coastguard*, compared to a 1.3 dB increase in PSNR for *mother-daughter* that contains a much smaller amount of motions.

When the data rate is relatively small, the optimal leaky factor in maximizing the decoded video quality, for a specific data rate, varies across different data rates. Larger leaky factors do not always result in a gain in terms of the rate distortion performance, implying that the inclusion of a larger portion of the enhancement layer in the MCP loop is not always beneficial.

The operational rate distortion performance for the prediction drift scenario of LPLC are illustrated by the dotted lines in Fig. 3.16, Fig. 3.17, and Fig. 3.18. We fixed the enhancement layer MCP rate at 200 kbps, and decoded the mismatch below the MCP rate. It is seen that the rate distortion curve for the prediction drift scenario when $\alpha = 0$ completely overlaps with the curve for the same α but no drift. A relatively small drop occurs when the leaky factor is comparatively small, such as 0.5, but a steep drop occurs in the rate distortion performance when the leaky factor approaches 1, such as 0.98. For example, for all three video sequences, when $\alpha = 0.5$, a PSNR drop of roughly 1 dB occurs at the decoding data rate 175 kbps, but when α increases to 0.98, a drop as large as more than 3 dB is resulted at the same decoding data rate. It is expected to mitigate the prediction drift by inserting more intra-coded frames.

It is seen that the operational results we obtained in terms of the rate distortion performance of LPLC are consistent with the theoretic results presented in Fig. 3.14 and Fig. 3.15 for both scenarios, especially considering the role of the leaky factor. We would like to point out that the theoretic results of Fig. 3.14 and Fig. 3.15 cannot present a precise rate distortion bound for the operational results, due to our mod-

eling of the source video signal and the assumptions we used in deriving the closed form expressions. Nevertheless, the theoretic results still provide a correct trend, thus beneficial in guiding the design of a practical implementation. Specifically, the results demonstrated by Fig. 3.14 and Fig. 3.15 provide an insight in understanding the functionalities of the leaky factor in LPLC, which is critical in determining both the coding efficiency and error resilience performance. The theoretic results recommend that an adaptive approach in choosing the leaky factor, rather than fixing it, is a better solution to wisely use the leaky factor in improving the overall rate distortion performance of LPLC.

3.5 Conclusions

In this chapter, we contribute the following work for theoretic analysis of LPLC:

- To theoretically analyze the rate distortion performance of LPLC, we have developed an alternative block diagram of LPLC. Similar to the original LPLC framework, the alternative block diagram includes two motion compensated prediction (MCP) loops, where the base layer PEF is encoded in the base layer MCP step and the mismatch is encoded in the enhancement layer MCP step. Different from the original framework, the leaky factor is only present in the enhancement layer MCP step in the alternative block diagram, which significantly simplifies the theoretic analysis. We have addressed the theoretic analysis of LPLC using two different approaches, namely the one using rate

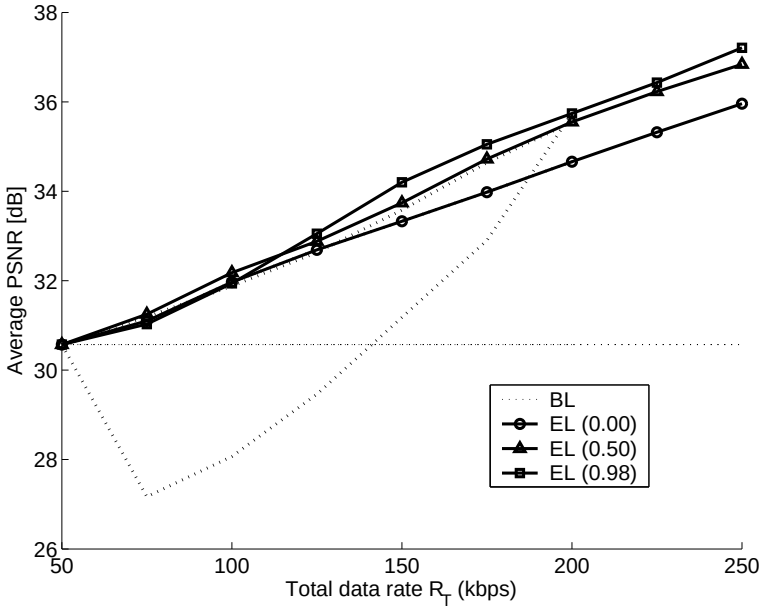


Fig. 3.16. Operational rate distortion performance of LPLC for QCIF *foreman* at various leaky factors; The base layer MCP rate is fixed at $R_B = 50$ kbps; Error drift occurs due to data rate truncation at the enhancement layer MCP rate $R_E = 150$ kbps; Solid lines represent the scenario without error drift; Dotted lines declining from the respective solid lines represent the scenario with drift. (obtained from the implementation of LPLC using SAMCoW; BL: reconstruction using the base layer alone; EL: reconstruction using both layers)

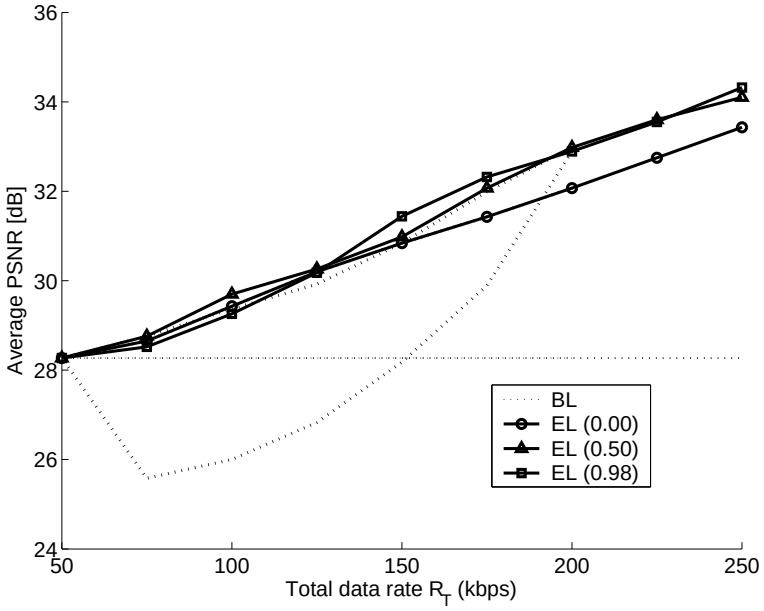


Fig. 3.17. Operational rate distortion performance of LPLC for QCIF *coastguard* at various leaky factors; The base layer MCP rate is fixed at $R_B = 50$ kbps; Error drift occurs due to data rate truncation at the enhancement layer MCP rate $R_E = 150$ kbps; Solid lines represent the scenario without error drift; Dotted lines declining from the respective solid lines represent the scenario with drift. (obtained from the implementation of LPLC using SAMCoW; BL: reconstruction using the base layer alone; EL: reconstruction using both layers)

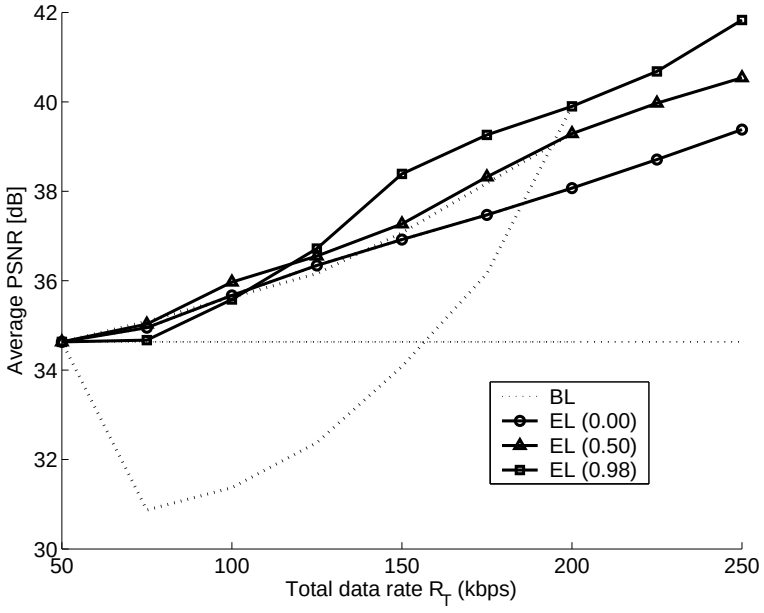


Fig. 3.18. Operational rate distortion performance of LPLC for QCIF *mthrdghtr* at various leaky factors; The base layer MCP rate is fixed at $R_B = 50$ kbps; Error drift occurs due to data rate truncation at the enhancement layer MCP rate $R_E = 150$ kbps; Solid lines represent the scenario without error drift; Dotted lines declining from the respective solid lines represent the scenario with drift. (obtained from the implementation of LPLC using SAMCoW; BL: reconstruction using the base layer alone; EL: reconstruction using both layers)

distortion theory and the one using a quantization noise model, based on the alternative block diagram.

- In the first approach for theoretically analyzing LPLC, we have used the optimum forward channel derived from rate distortion theory to model the encoding of a 2D image and obtain the parametric rate distortion functions for LPLC in closed form for one scenario where the enhancement layer is intact and the other where it has drift. For each scenario, the closed form rate distortion functions are in relation to three parameters: the power spectral density (PSD) of the input video frame, the probability distribution of the motion vector estimation errors, as well as the leaky factor.

- In the second approach for theoretically analyzing LPLC, we have addressed the rate distortion performance of LPLC by using quantization noise modeling. Since the optimum forward channel used in the first approach is derived from rate distortion theory, it provides the rate distortion bound at which a 2D stationary, Gaussian random signal is encoded. The quantization noise model in the second approach is a heuristic model, which was obtained from the operational results. The optimum forward channel specifies parametric rate distortion functions, whereas the use of the quantization noise model provides a closed formulation of the MSE distortion that is explicitly related to the data rate. We also obtain closed form rate distortion functions for the two scenarios of LPLC.

- We have evaluated both closed form expressions and demonstrated that the leaky factor is critical in the performance of coding efficiency. We have validated that with the partial or full inclusion of the enhancement layer in the MCP loop, LPLC does improve the coding efficiency as opposed to the conventional layered scalable coding. When the enhancement layer has no error drift, it is shown that at a specific leaky factor value between 0 and 1, the decoded video quality increases with the increase of the data rate. When the data rate is sufficiently large, LPLC achieves a better rate distortion performance with increasing leaky factor. It is interesting to note that when the enhancement layer data rate is small, it might be possible that a larger leaky factor yields a less efficient codec, especially when the leaky factor is close to 1. We have also shown that the leaky factor is critical in error resilience performance when the enhancement layer in LPLC suffers from error drift. When drift occurs in the enhancement layer, it is observed that larger leaky factors yield a larger drop in the rate distortion performance, especially when the leaky factor approaches 1. We have simulated both scenarios, with and without drift in the enhancement layer, using SAMCoW and evaluated the operational rate distortion performance of LPLC associated with various leaky factors for video sequences containing varying degrees of motions. It is shown that the theoretical results conform with the operational results.

4. MULTIPLE DESCRIPTION SCALABLE CODING FOR ERROR RESILIENT VIDEO TRANSMISSION OVER PACKET NETWORKS

4.1 Introduction

As we mentioned in Section 1.1.1 of Chapter 1, two types of scalabilities exist in current scalable video streaming schemes: (1) nested scalability, in which different representations of each frame are generated using layered scalable coding and have to be decoded in a fixed sequential order, and (2) parallel scalability, which is used in multiple description coding (MDC) where different descriptions are mutually refinable and independently decodable [19]. In this chapter, we use the general framework that applies to both scalabilities in Chapter 2 from another perspective [75]. The framework in Fig. 2.2 demonstrates the similarity between the leaky prediction layered video coding (LPLC) and an MDC scheme that uses motion compensation. Based on this framework, we introduce nested scalability into each description of the MDC stream and propose a fine granularity scalability (FGS) based MDC approach. We also develop a scalable video coding structure that is characterized by the dual-leaky prediction, termed Dual-LPLC, to balance the trade-off between coding efficiency and the error resilience performance of the coded bitstream.

A majority of current multiple description (MD) image coding approaches were motivated by single description coding (SDC) schemes. Multiple description scalar quantization (MDSQ) [76], as opposed to the single description quantization in the transform domain, is usually regarded as the first approach to designing practical MDC techniques and applying them to image coding. This work was extended in [77], where a wavelet transform was utilized instead of the conventional DCT. Multiple description transform coding (MDT) is another approach, where multiple description transforms such as pairwise correlating transform [78] or polyphase transform [79] were employed to generate two sets of transform coefficients, in contrast to the traditional single description orthogonal transforms such as DCT and the wavelet transform.

For MD video coding, a significant challenge is how to avoid or lessen the mismatch between the encoder and the decoder if only one description is available when motion compensation has been used. If MD image coding approaches such as MDSQ or MDT are used to code video, the encoder has to determine the description from which it selects the reference frame for motion compensation. If the encoder exploits the reference frame that is reconstructed by both descriptions, drift will occur at the decoding side if only one description is available at the decoder. This results in serious video quality degradation. There are roughly two MD video coding approaches. The first type extends MD image coding schemes to video by designing appropriate error-drift control mechanisms. In the work presented in [80, 81], three separate loops are used at the encoder to take into account the three possible scenarios that

may occur at the decoder. The scheme uses pairwise correlating transforms to realize MDC. The same three-loop motion compensation structure is used in [57] and matching pursuits are used to achieve MDC. In [82], MDT is incorporated within the conventional motion compensation loop using leaky prediction to encode the predicted error frame (PEF). In order to control the possible error-drift, a leaky prediction mechanism is proposed where part of the reconstruction of the reference frame, based on both descriptions, is utilized. The second group of MD video coding schemes directly exploits the characteristic of motion prediction. In [83], MDC is designed by encoding motion vector data into two descriptions. In [53], an MD video coding scheme using motion compensation is proposed, which is referred to as MDMC.

As we discussed in Chapter 2, MDMC predicts each frame from the two previous frames thus providing two sets of motion vectors for each frame in the central loop. It then uses the frame prior to the previous one as the reference in each of the two side loops. The mismatch between the central loop and the side loop is transmitted to the decoder. In general, if the mismatch between one description and two descriptions is available, more than one reconstruction for each frame is possible when two descriptions are present. Estimation techniques are thus investigated in [57] and [84] to further improve the video quality when both descriptions are received.

Aside from MDC approaches based on source coding schemes, MDC techniques based on channel coding such as FEC have been proposed in [85], and FEC-based MD video streaming over Internet is investigated in [86]. The error resilient performance

of MDC over erasure packet networks is investigated in [87] and [88] and the superiority of MDC over SDC is demonstrated. Moreover, studies have been conducted to facilitate MDC for error resilient video streaming over wireline and wireless networks with specific networking strategies. Content Delivery Networks (CDN) coupled with MDC has been investigated in [89, 90]. In particular, approaches of MDC with path diversity have been proposed in [91, 92]. An approach that addresses client cooperation in distributing content to alleviate the server's burden is presented in [93]. This is denoted as Cooperative Networking (CoopNet), and MDC is used to stream video over the CoopNet due to its good balance between robustness and controllable redundancy.

A problem of concern in MDC design is the redundancy introduced by the parallel scalable structure. A redundancy rate distortion (RRD) function was proposed to evaluate the performance of a balanced MDC coder [94]. For the two-description scenario, let $R^{(1)}$ and $R^{(2)}$ denote the data rates for each description, $D_M^{(1)}$, $D_M^{(2)}$, and D_M denote the distortion if only the first description is available, the distortion if only the second description is present, and the distortion if both descriptions are available at the decoder, respectively. Let R_S denote the data rate required by a SDC scheme to achieve the same distortion D_M. The RRD optimization problem for MDC is formulated as

$$\begin{aligned} \min\{D_M^i\} \qquad & \text{subject to } R_M^{(i)} \leq R_{\text{budget}}^{(i)} \\ & \text{given } D_M \text{ achieved at } R_S \text{ by an SDC scheme, } \; i = 1, 2, \end{aligned} \tag{4.1}$$

where $R^{(1)}_{\text{budget}}$ and $R^{(2)}_{\text{budget}}$ specify the data rate budgets for each description.

Comparison between layered scalable coding and MDC has been discussed in [95, 96]. It was demonstrated that the performance of these two scalable schemes depends on the transmission scenarios considered. Since layered scalable coding techniques are relatively more mature than MDC, a class of MDC schemes have evolved from the layered coding framework, which is referred to as multiple description layered coding (MDLC) [97–99]. MDLC duplicates the base layer information in both descriptions and splits the enhancement layer streams between the two descriptions. An approach that combines MDT and MDLC under one framework was proposed in [100].

4.2 MDC with Nested Scalability in Every Single Description - FS-MDC

In this section, we focus on MDC and use the similarity between LPLC and MDMC shown in Fig. 2.2 of Chapter 2 to migrate the idea of "nested scalability" in LPLC into MDMC. Other approaches that introduce the nested scalability into each description of the MDC structure include the work presented in [101], where FEC was used.

In the original MDMC scheme, the mismatch between the central loop and the side loop is transmitted to compensate for the inconsistency between motion compensation at the decoder and that performed by the encoder if only one description is available. Based on the similarity between LPLC and MDMC, we view MDMC from an alternative point of view. We consider the reconstructed frame in the central loop $F^{(r)}_C(n)$ as the "base layer" frame $F^{(r)}_B(n)$, and the reconstruction in the side loop

$F_S^{(r)}(n)$ as the "enhancement layer" $F_E^{(r)}(n)$, which correspond to the base layer and the enhancement layer in LPLC respectively. We call this coding framework fine scalable MDC, or FS-MDC. The coding structure of FS-MDC can be compared to the nested scalable coding structure if we introduce nested layered scalability independently to each description of the Video Redundancy Coding (VRC) bitstream [55]. The difference between FS-MDC and the nested scalable VRC is that the base layer in FS-MDC is achieved by two descriptions. The enhancement layer, on the other hand, is obtained from only one description, which is similar to the nested scalable VRC. The reconstructed base layer and enhancement layer of the nth frame in FS-MDC are obtained using

$$F_B^{(r)}(n) = \alpha\tilde{F}_B(n-1) + (1-\alpha)\tilde{F}_B(n-2) + \hat{e}_B(n), \tag{4.2}$$

$$F_E^{(r)}(n) = \tilde{F}_B(n-2) + \hat{e}_B(n) + \hat{\psi}(n), \tag{4.3}$$

where

$$\tilde{F}_B(n-1) = \mathrm{MC}_{MV1(n)}\left\{F_B^{(r)}(n-1)\right\}, \tag{4.4}$$

$$\tilde{F}_B(n-2) = \mathrm{MC}_{MV2(n)}\left\{F_B^{(r)}(n-2)\right\}. \tag{4.5}$$

$e_B(n)$ is the base layer PEF and $\psi(n)$ is the mismatch between the two PEFs from the two layers, where

$$e_B(n) = F(n) - \alpha\tilde{F}_B(n-1) - (1-\alpha)\tilde{F}_B(n-2), \tag{4.6}$$

$$\psi(n) = F(n) - \tilde{F}_B(n-2) - \hat{e}_B(n). \tag{4.7}$$

FS-MDC introduces the MD structure in the base layer and generates the enhancement layer by using nested scalable coding to encode the residual information

within one description. The essential framework of FS-MDC is identical to that of MDMC, however, FS-MDC views the functionality of the mismatch transmitted in the side loops as the enhancement information, instead of as compensation to the central loop in MDMC. This is motivated by the similarity between MDMC and LPLC, as LPLC treats the mismatch between two descriptions of one frame as the enhancement layer. Moreover, it is straightforward to incorporate fine granularity scalability (FGS) in FS-MDC by using an embedded coding scheme to encode the enhancement layer. This is where the "Fine Scalability" in FS-MDC is manifested. It is to be emphasized that since the base layer in FS-MDC is implemented using an MD structure, as given in (4.2), the base layer of the current frame can be reconstructed only when the base layers in both descriptions are available at the decoder. From a single description point of view, this increases the risk of drift in the base layer at the decoder for video streaming over error-prone channels. This disadvantage could be compensated for by prioritizing the transport of the base layer, as the nested scalability in FS-MDC is suitable for unequal error protection (UEP). From the overall descriptions point of view, the MD structure in the base layer has error resilient capability since any single description of the base layer is decodable. The enhancement layer, on the other hand, can be achieved if only one description for the enhancement layer is available, regardless of the availability of the enhancement layer in the other description.

During decoding, FS-MDC always favors the reconstruction by the enhancement layer, i.e., the reconstruction by (4.3). This is different from the original MDMC,

since MDMC always favors the reconstruction in the central loop, which corresponds to the base layer loop in FS-MDC. Similar to what we discussed with LPLC in Chapter 2, the superiority of the enhancement layer in FS-MDC is not always achievable but depends on the parameter set in the coding structure, since the enhancement layer carries the mismatch between the two PEFs of the two layers instead of the actual residual information between the original signal and the reconstruction of the base layer. In fact, the mismatch carried by the enhancement layer in FS-MDC is

$$\begin{aligned}\psi(n) &= \alpha\left(\mathrm{MC}_{MV1(n)}\left\{F_B^{(r)}(n-1)\right\} - \mathrm{MC}_{MV2(n)}\left\{F_B^{(r)}(n-2)\right\}\right) \\ &\quad + \left(F(n) - F_B^{(r)}(n)\right), \end{aligned} \tag{4.8}$$

which demonstrates that the superiority of the enhancement layer over the base layer in FS-MDC is largely determined by the second-order prediction factor α. When $\alpha = 0$, $\hat{\psi}(n)$ is the quantized version of the mismatch between the original signal and the reconstructed frame of the base layer. Thus, any approximation of $\psi(n)$ has more information about the original signal than knowledge of just the base layer. If $\alpha > 0$, $\hat{\psi}(n)$ includes another portion that is the difference between the two motion compensated frames multiplied by the leaky factor. In summary, three factors affect the performance of the enhancement layer in FS-MDC: the quantization step for the mismatch $\psi(n)$, the leaky factor α, and the difference between the two reconstructions by the base layer from the two reference frames.

In FS-MDC, one description could be estimated from the other description, by utilizing the MD structure used in the base layer, which is similar to the original

MDMC scheme. Given $F_E^{(r)}(n-1)$ and $\hat{\psi}(n+1)$, the optimal estimate for the motion compensated frame of the base layer $\tilde{F}_B(n)$ in the minimum mean square error (MSE) sense is

$$\begin{aligned}
& \tilde{F}_{\text{mse}}(n) = E\left[\tilde{F}_B(n)|F_E^{(r)}(n+1), \hat{\psi}(n+1)\right] \\
= \;& \tilde{F}_E(n-1) + \frac{\hat{\psi}(n+1)}{\alpha} + \text{MC}_{MV2(n+1)}\left\{E\left[\left(\psi(n-1) - \hat{\psi}(n-1)\right)\right.\right. \\
& \left.\left. - \left(e_B(n-1) - \hat{e}_B(n-1)\right)|F_E^{(r)}(n-1), \hat{\psi}(n+1)\right]\right\} \\
+ \;& \frac{E\left[\left(\psi(n+1) - \hat{\psi}(n+1)\right) - \left(e_B(n+1) - \hat{e}_B(n+1)\right)|F_E^{(r)}(n-1), \hat{\psi}(n+1)\right]}{\alpha}.
\end{aligned} \tag{4.9}$$

Therefore, aside from the two reconstructions from the base and enhancement layers (in equations (4.2) and (4.3)), we obtain a third reconstruction for each frame by exploiting the information from the enhancement layer of the next frame belonging to the other description via backward motion compensation as follows

$$F_{\text{est}}^{(r)}(n) = \text{MC}_{MV1(n+1)}^{-1}\left\{\tilde{F}_E(n-1) + \frac{\hat{\psi}(n+1)}{\alpha}\right\}, \;\; 0 < \alpha \leq 1. \tag{4.10}$$

This reconstruction is one advantage in FS-MDC, since the information of the other description is implicitly included in the available description. If the original video sequence is simply partitioned into two streams and the streams coded using motion compensation independently of each other as in VRC, each description would only be approximated, in the event that data is corrupted or lost, by error concealment schemes such as forward motion compensation.

Considering the possibility that the performance of the enhancement layer might be inferior to that of the base layer in FS-MDC, we use the maximum likelihood

(ML) estimation scheme originally proposed in [57] to facilitate the FS-MDC scheme. We assume that the quantization noise in each pixel is an independent, identically distributed (i.i.d.) zero-mean Gaussian random variable. For a frame with size $M \times N$, let $\rho_B^{(r)}(x,y)$, $\rho_E^{(r)}(x,y)$, and $\rho_{ML}^{(r)}(x,y)$ denote the pixel value at the location (x,y) in $F_B^{(r)}$, $F_E^{(r)}$, and $F_{ML}^{(r)}$ respectively, where $F_{ML}^{(r)}$ is the image reconstructed using ML estimation. Thus,

$$
\begin{aligned}
F_B^{(r)} &= \left\{\rho_B^{(r)}(x,y)\right\}_{(x,y)=(0,0)}^{(M-1,N-1)}, && (4.11)\\
F_E^{(r)} &= \left\{\rho_E^{(r)}(x,y)\right\}_{(x,y)=(0,0)}^{(M-1,N-1)}, && (4.12)\\
F_{ML}^{(r)} &= \left\{\rho_{ML}^{(r)}(x,y)\right\}_{(x,y)=(0,0)}^{(M-1,N-1)}. && (4.13)
\end{aligned}
$$

We obtain the ML estimate for each pixel as

$$
\begin{aligned}
\rho_{ML}^{(r)}(x,y) &= \left(\begin{bmatrix}1 & 1\end{bmatrix}\Sigma^{-1}\begin{bmatrix}1\\1\end{bmatrix}\right)^{-1}\begin{bmatrix}1 & 1\end{bmatrix}\Sigma^{-1}\begin{bmatrix}\rho_B^{(r)}(x,y)\\ \rho_E^{(r)}(x,y)\end{bmatrix}\\
&\triangleq \begin{bmatrix}\pi & 1-\pi\end{bmatrix}\begin{bmatrix}\rho_B^{(r)}(x,y)\\ \rho_E^{(r)}(x,y)\end{bmatrix}\\
&= \pi\rho_B^{(r)}(x,y) + (1-\pi)\rho_E^{(r)}(x,y), && (4.14)
\end{aligned}
$$

where Σ is the cross-correlation matrix between the two reconstruction errors, $\rho - \rho_B^{(r)}$ and $\rho - \rho_E^{(r)}$,

$$\begin{aligned}
\Sigma &= E\left(\begin{bmatrix} \rho - \rho_B^{(r)} \\ \rho - \rho_E^{(r)} \end{bmatrix} \begin{bmatrix} \rho - \rho_B^{(r)} & \rho - \rho_E^{(r)} \end{bmatrix}\right) \\
&= \begin{bmatrix} E\left[\left(\rho - \rho_B^{(r)}\right)^2\right] & E\left[\left(\rho - \rho_B^{(r)}\right)\left(\rho - \rho_E^{(r)}\right)\right] \\ E\left[\left(\rho - \rho_B^{(r)}\right)\left(\rho - \rho_E^{(r)}\right)\right] & E\left[\left(\rho - \rho_E^{(r)}\right)^2\right] \end{bmatrix} \\
&\triangleq \begin{bmatrix} a & b \\ b & d \end{bmatrix},
\end{aligned} \tag{4.15}$$

where $\rho(x, y)$ denotes the original pixel value at location (x, y), and $E\{\cdot\}$ denotes the expectation of a random variable or random vector. We use empirical averages to approximate the expectations in (4.15). For instance,

$$E\left[(\rho - \hat{\rho}_B)^2\right] \cong \frac{1}{M \times N} \sum_{(x,y)=(0,0)}^{(M-1,N-1)} \left(\rho(x, y) - \rho_B^{(r)}(x, y)\right)^2. \tag{4.16}$$

Combining (4.14) and (4.15), we have

$$\pi = \frac{d - b}{a + d - 2b}, \tag{4.17}$$

and we refer to π as the ML coefficient hereafter. This necessitates that we transmit the ML coefficient(s) associated with each frame as the side information to ensure that the ML estimate of each frame $F_{ML}^{(r)}$ can be obtained at the decoder. For color video sequences, we transmit three ML coefficients for each frame, one for the luminance component and two for the two chrominance components. Each ML coefficient is obtained as shown above. With the ML coefficient $\pi(n)$ calculated for

the nth frame, we obtain a fourth reconstruction $F_{ML}^{(r)}(n)$, the ML estimate of each frame, as

$$F_{ML}^{(r)}(n) = \pi(n)F_B^{(r)}(n) + (1 - \pi(n))F_E^{(r)}(n), \tag{4.18}$$

where $F_B^{(r)}(n)$ and $F_E^{(r)}(n)$ are the two reconstructions in FS-MDC.[1]

4.3 Dual-Leaky Prediction Layered Video Coding - Dual-LPLC

In the previous section, we mentioned that from the nested scalability point of view, the coding structure of FS-MDC falls into the category of the conventional FGS structure where the enhancement layer is completely excluded from the motion compensation loop. This maximizes the error resilient capability of the bitstream but results in poor coding efficiency. Using LPLC that introduces leaky prediction in the enhancement layer in the nested scalable coding structure [20–22], we utilize leaky prediction in the enhancement layer for each single description of FS-MDC. We propose a coding structure characterized as "dual-leaky" shown in Fig. 4.1, and refer to this coding structure as Dual-LPLC. The "dual-leaky" feature is defined in the sense that one leaky prediction is exploited in the parallel scalable coded base layer and a second leaky prediction adopted in the nested scalable coded enhancement layer. The leaky prediction in the base layer is manifested by the second-order prediction that originated from the MD structure, while the leaky prediction in the enhancement layer is implemented by incorporating a scaled

[1]Interested readers may refer to [57] for the detailed derivation of (4.18). The ML approach we proposed here is very similar to that presented in [57].

version of the enhancement layer in the motion compensation loop of each single description. Dual-LPLC combines nested scalability and parallel scalability under one framework and maintains a good balance between scalable video streaming and error resilient streaming due to its "dual-leaky" prediction feature. In Dual-LPLC, the base layer is obtained in exactly the same way as in FS-MDC given in (4.2), which is $F_B^{(r)}(n) = \alpha\tilde{F}_B(n-1) + (1-\alpha)\tilde{F}_B(n-2) + \hat{e}_B(n)$. In contrast to (4.3), the enhancement layer is obtained using

$$\begin{aligned} F_E^{(r)}(n) &= \tilde{F}_B(n-2) + \theta\left(\tilde{F}_E(n-2) - \tilde{F}_B(n-2)\right) + \hat{e}_B(n) + \hat{\psi}(n) \\ &= \theta\tilde{F}_E(n-2) + (1-\theta)\tilde{F}_B(n-2) + \hat{e}_B(n) + \hat{\psi}(n), \end{aligned} \tag{4.19}$$

where $0 \leq \theta \leq 1$ denotes the leaky factor introduced in the nested scalable coded enhancement layer of Dual-LPLC in Fig. 4.1.

4.4 Experimental Results

We use the *foreman* sequence in our experiments. All frames are in 4:2:0 YUV QCIF format. We implemented our proposed approach by modifying the H.26L reference software version TML9.4 [54]. We chose H.26L since compared to previous video coding standards, such as MPEG-4 and H.263+, H.26L contains more features that further improve the coding efficiency at all bit rates, and fulfill several tasks in the network abstraction layer (NAL) to improve the error resilience performance of the bitstream. In all of our experiments, we turned on all seven block-shape options and used the full search range of 16 for motion estimation. We encoded both the

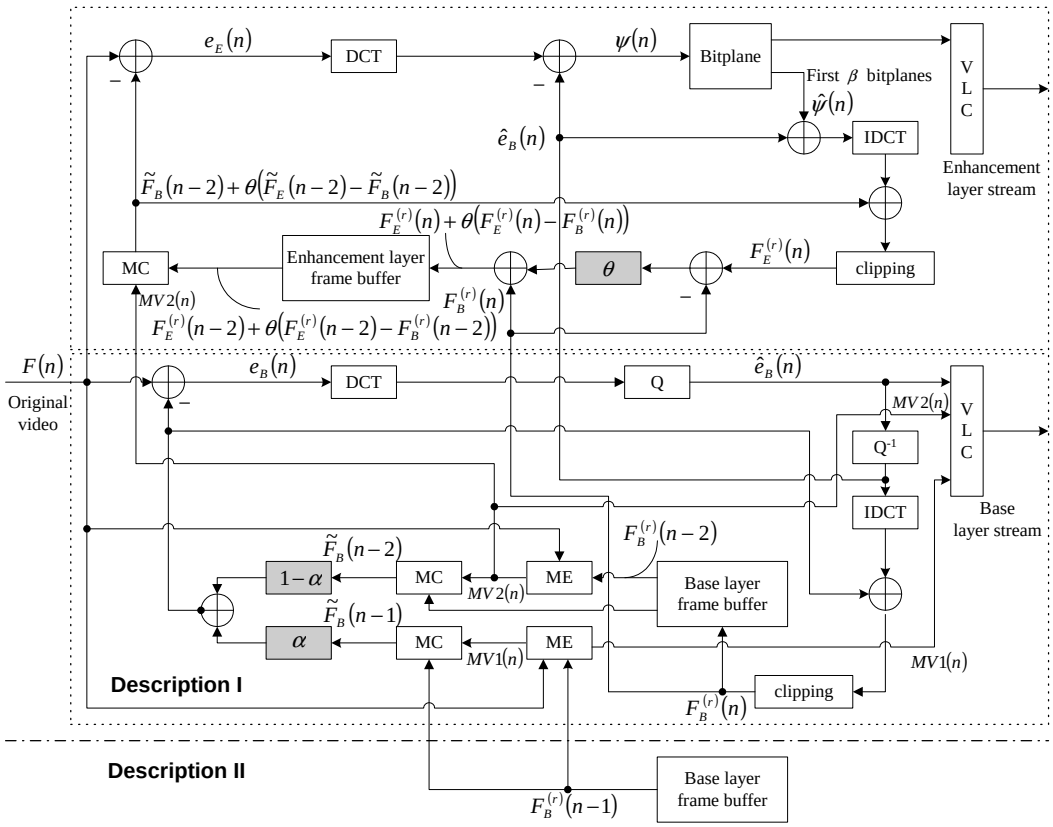


Fig. 4.1. A dual-leaky prediction error resilient layered scalable coding structure (Dual-LPLC)

base and enhancement layers using the UVLC mode with the same VLC table as the non-scalable coding structure. We adopted one slice for each frame and turned off the rate-distortion optimization option. Since no rate control is implemented in current H.26L reference software, we encoded all sequences at a frame rate of 30 fps and adjusted the quantization parameters to achieve various decoded video qualities.

4.4.1 Experiments - FS-MDC

We intra-coded the first two frames of each sequence and inter-coded all successive frames in testing FS-MDC. The INTRA frames, the PEFs for the INTER frames of the base layer, together with the two sets of motion vector data for the INTER frames compose the base layer stream of each description, and the PEFs for the INTER frames of the enhancement layer compose the whole enhancement layer. We encoded all ML coefficients using 6-bits. We observed that the ML coefficients do not change much from frame to frame, implying that a more efficient way to encode ML coefficients is possible. We first evaluate the performance of the enhancement layer relative to that of the base layer in FS-MDC. As shown in Fig. 4.2(a), the performance of both the base layer and the enhancement layer are related to the leaky factor in the second-order predictor. From Eqn. (4.2), the motion compensated frame for the base layer of each frame is a linear combination of the previous two reconstructions using the base layer in the two descriptions. Thus, the decoded quality of the base layer varies within a small range of PSNR even though the quantization step for the base layer is fixed. The enhancement layer carries the mismatch

between the two PEFs of the two layers, and thus also varies with the leaky factor. When the mismatch $\psi(n)$ is coarsely quantized and the leaky factor α is close to 1, the performance of the enhancement layer is inferior to that of the base layer. With the decrease of the leaky factor in the second-order predictor, the decoded video quality of the enhancement layer increases beyond that of the base layer and the enhancement layer finally achieves superior performance over the base layer when the leaky factor is small enough. This is because the term $\left(F(n) - F_B^{(r)}(n)\right)$ has a larger contribution in the mismatch $\psi(n)$ when the leaky factor decreases, and the information carried by the enhancement layer thus compensates for the quantized error caused by the base layer. Since we are interested in the superior performance of the enhancement layer in FS-MDC, we chose a smaller leaky factor value, $\alpha = 0.15$. As shown in Fig. 4.2(b), increasing the accuracy of $\psi(n)$ increases the performance of the enhancement layer beyond the video quality achieved by the base layer. Fig. 4.2 demonstrates that the scheme for FS-MDC facilitated by ML estimation achieves superior or the same decoded video quality compared to those by both layers regardless of the leaky factor value and the quantization step size of the enhancement layer.

Next we evaluate the performance of FS-MDC regarding the redundancy introduced by the parallel scalability adopted in FS-MDC. In other words, we need to evaluate FS-MDC regarding its RRD performance given in Eqn. (4.1), since the base layer in FS-MDC is actually implemented by an MD coding structure. In contrast with the RRD evaluation of conventional MDC schemes, we examine the RRD per-

formance of FS-MDC by referring to a single description scalable coding (SDSC) structure, since FS-MDC contains two layers. Compared to FS-MDC, the base layer in SDSC includes the INTRA frames, the PEFs for INTER frames of the base layer, together with one set of motion vector data for each INTER frame. The enhancement layer is composed of the PEFs for the INTER frames of the enhancement layer. We implement the reference SDSC by choosing the same quantization parameters as in FS-MDC. To have a fair comparison, we use two reference frames to encode each INTER frame in SDSC, as opposed to the two reference frames used in FS-MDC to obtain the two set of motion vectors for each INTER frame. We list the RRD performance of FS-MDC referring to SDSC as a function of the leaky factor and the quantization step for the enhancement layer in Table 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, and 4.8. Table 4.4, 4.5, and Table 4.8 present the percentage redundancy introduced by the INTER frames of FS-MDC over SDSC. This redundancy is mainly incurred by the second set of motion vectors required by the parallel scalable coding structure adopted for the base layer of FS-MDC, and is also due to the reduced coding efficiency of the PEFs in both layers. The reduction in coding efficiency arises since the reference frames are fixed and a reference frame that is two-frame away from the current coded frame is used. We also list the difference decoded video quality between FS-MDC and SDSC for both layers. Since we maintain the same quantization steps for both schemes, the differences in PSNR vary within a small range.

Table 4.1, 4.2, 4.4, and 4.5 demonstrate that the RRD performance of FS-MDC varies closely with the leaky factor α in the second-order predictor. The redundancy

in FS-MDC is relatively large for a large leaky factor value due to the large mismatch between the two PEFs that are included in the enhancement layer. The RRD performance of FS-MDC is poor for small leaky factor values due to the poor coding efficiency of the base layer that comes about from using a remote frame. Table 4.4 and 4.5 show that the redundancy can be decreased by 20% by choosing an appropriate leaky factor value such as 0.55 as opposed to a smaller leaky factor value of 0.0. Moreover, Table 4.1 and 4.2 also show that the bits consumed by the second set of motion vectors make up more than 1/3 of the total bits of the base layer of each frame. This is partly due to the fact that we encode the second motion vectors for each frame in the same manner as the non-scalable coding structure in H.26L. A scheme that jointly encodes the two sets of motion vectors for each frame is expected to achieve better coding efficiency. On the other hand, the enhancement layer of FS-MDC consumes almost five times as many bits as those by the enhancement layer in SDSC. Besides the reduced coding efficiency of the enhancement layer in FS-MDC, another reason for such a large redundancy is that we use the UVLC mode and always use the same VLC table in our schemes. It seems that the UVLC designed in H.26L is not suitable for large dynamic ranges. We expect a better coding efficiency if we adopt the context-based adaptive binary arithmetic coding (CABAC) mode in H.26L. Table 4.6 and 4.8 show the redundancy introduced by FS-MDC, relative to SDSC, in INTER frames ranges from 80% to 33% with the decrease of the quantization step size. As previously mentioned, an increase in symbol value also results in poorer performance of SDSC due to the UVLC entropy coding scheme it used,

which partly contributes to the better RRD performance of FS-MDC in the case of smaller quantization steps chosen for the enhancement layer.

4.4.2 Experiments - Dual-LPLC

In this subsection, we present the performance of the leaky prediction used in the nested scalability of the Dual-LPLC scheme. First we list the RRD performance of Dual-LPLC relative to SDSC in Table 4.9, Table 4.10, and Table 4.11, as a function of the nested scalable leaky factor θ. When the leaky factor $\theta = 0$, Dual-LPLC reduces to the coding structure of FS-MDC. It can be observed that Dual-LPLC greatly improves the RRD performance over FS-MDC. For example, in Table 4.11, when the leaky factor θ increases from 0.0 to 0.60, Dual-LPLC achieves an almost 20% reduction in redundancy than FS-MDC at the same decoded video quality. It should be pointed out in contrast to the conventional LPLC schemes, Dual-LPLC has worse RRD performance for large leaky factor values than small leaky factors. This is because the information carried by the enhancement layer of Dual-LPLC is the mismatch between the two PEFs and large leaky factor increases the difference between the reconstructions of the two layers, which results in poorer coding performance of the enhancement layer. Next we examine the error recovery capability of Dual-LPLC for video streaming over an error-prone environment. We grouped a fixed number of frames into a GOP and intra-coded the first two frames and inter-coded the successive frames in each GOP. Similar to the scheme used in [20], we assumed error-free transmission of the base layer and assumed that either the en-

hancement layer of the first INTER frame in each GOP (which belongs to Description I of Dual-LPLC coded bitstream) was lost due to channel errors or the enhancement layer of the second INTER frame in each GOP (which belongs to Description II of Dual-LPLC) was lost. From Fig. 4.3, it can be observed that the error recovery capability of Dual-LPLC is related to the leaky factor θ of the nested scalability in Dual-LPLC. Smaller values of θ result in faster error recovery, which is consistent with the results obtained in [20] for single description LPLC. Also, because of the parallel scalability exploited in Dual-LPLC, one description completely stays intact when errors corrupt the enhancement layer of the other description.

4.5 Conclusions

In this chapter, we contribute the following work for introducing nested scalability in the parallel scalable coding structure:

- We have used the framework that applies to both LPLC and MDMC to introduce the nested scalability into each description of the MDC stream. We have proposed a fine granularity scalability (FGS) based MDC approach, termed fine-scalable-MDC, FS-MDC. The essential framework of FS-MDC is identical to that of MDMC, however, FS-MDC views the functionality of the mismatch transmitted in the side loops as the enhancement information, instead of as compensation to the central loop in MDMC. From the overall descriptions

point of view, the MD structure in the base layer has error resilient capability since any single description of the base layer is decodable.

- We have proposed a coding structure characterized as "dual-leaky", and referred to it as Dual-LPLC. The "dual-leaky" feature is defined in the sense that one leaky prediction is exploited in the parallel scalable coded base layer and a second leaky prediction is included in the nested scalable coded enhancement layer. The leaky prediction in the base layer is manifested by the second-order prediction that originated from the MD structure, while the leaky prediction in the enhancement layer is implemented by incorporating a scaled version of the enhancement layer in the motion compensation loop of each single description. Dual-LPLC combines nested scalability and parallel scalability under one framework and maintains a good balance between scalable video coding and error resilient video coding due to its "dual-leaky" prediction feature.

We have observed that the leaky factor is critical and has four functionalities: (1) It affects the coding efficiency; (2) It is related with the error resilience performance of the coded bitstream; (3) It determines the superiority of the enhancement layer; and (4) It determines the performance of the approximation of one description from the other description. Our results have shown that the proposed schemes rendered the coded bitstream more resilient to errors without adding too much redundancy in the bitstream.

Our future work will focus on how to optimize the leaky factor to achieve the best performance under various scenarios. We will also further investigate how to perform backward motion compensation from forward motion vectors especially when motion vectors have fractional resolution. Our approach will exploit the base layer information to help reconstruct the enhancement layer by backward motion compensation operations.

Table 4.1
Evaluation of FS-MDC regarding the RRD performance with respect to the leaky factor ($0.5 \leq \alpha \leq 1.0$) in the second-order predictor (INTRA frames: QP=15; INTER frames: Base Layer QP=24, Enhancement Layer QP=31)

Leaky factor (α)	**FS-MDC**			
	Overall data rate (kb/s)	Base layer (bits/frame)		Enh. layer (b/f)
		PEFs for INTER frames	MV2 for INTER frames	
1.00	106.38	1601	931	895
0.95	105.28	1582	923	888
0.90	103.83	1550	909	885
0.85	102.84	1534	902	881
0.80	101.77	1510	893	877
0.75	102.16	1540	883	873
0.70	100.51	1501	875	868
0.65	100.20	1499	869	864
0.60	99.97	1491	870	861
0.55	99.68	1493	862	859
0.50	103.75	1618	872	854

Table 4.2
Evaluation of FS-MDC regarding the RRD performance with respect to the leaky factor ($0.0 \leq \alpha < 0.5$) in the second-order predictor (INTRA frames: QP=15; INTER frames: Base Layer QP=24, Enhancement Layer QP=31)

Leaky factor (α)	**FS-MDC**			
	Overall data rate (kb/s)	Base layer (bits/frame)		Enh. layer (b/f)
		PEFs for INTER frames	MV2 for INTER frames	
0.45	99.51	1495	858	852
0.40	100.15	1520	860	849
0.35	100.42	1532	859	849
0.30	101.60	1562	865	849
0.25	103.56	1624	871	844
0.20	103.12	1606	872	847
0.15	104.47	1641	874	846
0.10	106.86	1701	891	849
0.05	109.40	1761	902	850
0.00	112.72	1840	920	851

Table 4.3
Data rate performance of SDSC (INTRA frames: QP=15; INTER frames: Base Layer QP=24, Enhancement Layer QP=31)

Overall data rate (kb/s)	Base layer (b/f)	Enh. layer (b/f)
56.33	1703	175

Table 4.4
Evaluation of FS-MDC regarding the RRD performance (referring to SDSC) with respect to the leaky factor ($0.5 \leq \alpha \leq 1.0$) in the second-order predictor (INTRA frames: QP=15; INTER frames: Base Layer QP=24, Enhancement Layer QP=31)

Leaky factor (α)	**FS-MDC over SDSC**		
	Redundancy (in INTER frames)	Difference in decoded video quality [dB]	
		BL	EL
1.00	82%	-0.10	-0.72
0.95	81%	-0.06	-0.67
0.90	78%	0.01	-0.60
0.85	77%	0.06	-0.53
0.80	75%	0.07	-0.50
0.75	76%	0.00	-0.54
0.70	73%	0.14	-0.42
0.65	72%	0.14	-0.40
0.60	72%	0.15	-0.36
0.55	71%	0.16	-0.34
0.50	78%	-0.14	-0.52

Table 4.5
Evaluation of FS-MDC regarding the RRD performance (referring to SDSC) with respect to the leaky factor ($0.0 \leq \alpha < 0.5$) in the second-order predictor (INTRA frames: QP=15; INTER frames: Base Layer QP=24, Enhancement Layer QP=31)

Leaky factor (α)	**FS-MDC over SDSC**		
	Redundancy (in INTER frames)	Difference in decoded video quality [dB]	
		BL	EL
0.45	71%	0.19	-0.24
0.40	72%	0.22	-0.20
0.35	73%	0.21	-0.17
0.30	74%	0.20	-0.15
0.25	78%	0.06	-0.22
0.20	77%	0.21	-0.08
0.15	79%	0.18	-0.05
0.10	83%	0.12	-0.05
0.05	87%	0.06	-0.04
0.00	92%	-0.10	-0.11

Table 4.6
Evaluation of FS-MDC regarding the RRD performance with respect to the quantization step for the enhancement layer (INTRA frames: QP=15; INTER frames: Base Layer QP=24; Leaky factor in the second-order predictor $\alpha = 0.15$)

QP for Enh. layer	**FS-MDC**			
	Overall data rate (kb/s)	Base layer (bits/frame)		Enh. layer (b/f)
		PEFs for INTER frames	MV2 for INTER frames	
QP=25	105.11	1645	874	862
QP=24	106.08	1645	875	891
QP=23	115.50	1653	882	1191
QP=22	139.46	1662	896	1970
QP=21	176.78	1662	907	3210
QP=20	232.48	1672	922	5050

Table 4.7
Data rate performance of SDSC with respect to the quantization step for the enhancement layer (INTRA frames: QP=15; INTER frames: Base Layer QP=24)

QP for Enh. layer	Overall data rate (kb/s)	Base layer (b/f)	Enh. layer (b/f)
QP=25	56.45	1705	176
QP=24	56.59	1709	178
QP=23	61.97	1714	352
QP=22	80.49	1723	964
QP=21	116.08	1723	2145
QP=20	172.99	1731	4034

Table 4.8
Evaluation of FS-MDC regarding the RRD performance (referring to SDSC) with respect to the quantization step for the enhancement layer (INTRA frames: QP=15; INTER frames: Base Layer QP=24; Leaky factor in the second-order predictor $\alpha = 0.15$)

QP for Enh. layer	**FS-MDC over SDSC**		
	Redundancy (in INTER frames)	Difference in decoded video quality [dB]	
		BL	EL
QP=25	80%	0.18	-0.05
QP=24	81%		-0.03
QP=23	80%		0.03
QP=22	69%		0.10
QP=21	49%		0.14
QP=20	33%		0.13

Table 4.9
Evaluation of Dual-LPLC regarding the RRD performance with respect to the leaky factor for the nested scalability (INTRA: QP=15; INTER: BL QP=24, EL QP=22; Leaky factor in the parallel scalability $\alpha = 0.15$)

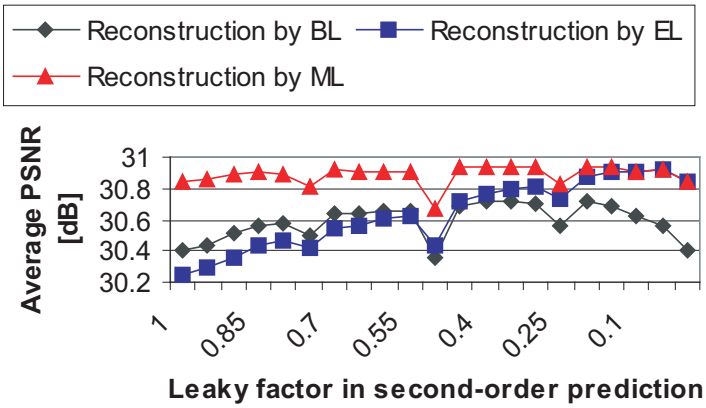
Leaky factor (θ)	**Dual-LPLC**			
	Overall data rate (kb/s)	Base layer (bits/frame)		Enh. layer (b/f)
		PEFs for INTER frames	MV2 for INTER frames	
0.00	139.46	1662	896	1970
0.10	134.88	1661	894	1820
0.20	130.39	1659	890	1675
0.30	128.01	1659	889	1596
0.40	126.61	1658	889	1550
0.50	127.33	1658	889	1575
0.60	124.96	1657	887	1498
0.70	126.83	1657	889	1559
0.80	130.78	1657	892	1688
0.90	139.53	1659	897	1974
1.00	158.24	1661	904	2593

Table 4.10
Data rate performance of SDSC (INTRA: QP=15; INTER: BL QP=24, EL QP=22)

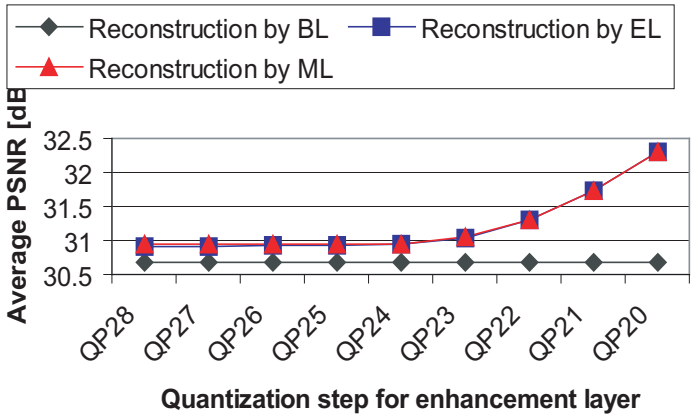
Overall data rate (kb/s)	Base layer (b/f)	Enh. layer (b/f)
80.49	1723	964

Table 4.11
Evaluation of Dual-LPLC regarding the RRD (referring to SDSC) performance with respect to the leaky factor for the nested scalability (INTRA: QP=15; INTER: BL QP=24, EL QP=22; Leaky factor in the parallel scalability $\alpha = 0.15$)

Leaky factor (θ)	**Dual-LPLC over SDSC**		
	Redundancy (in INTER frames)	Difference in decoded video quality [dB]	
		BL	EL
0.00	69%	0.18	0.10
0.10	63%		0.11
0.20	57%		0.12
0.30	54%		0.14
0.40	52%		0.16
0.50	53%		0.12
0.60	50%		0.17
0.70	53%		0.18
0.80	58%		0.16
0.90	69%		0.05
1.00	92%		-0.12

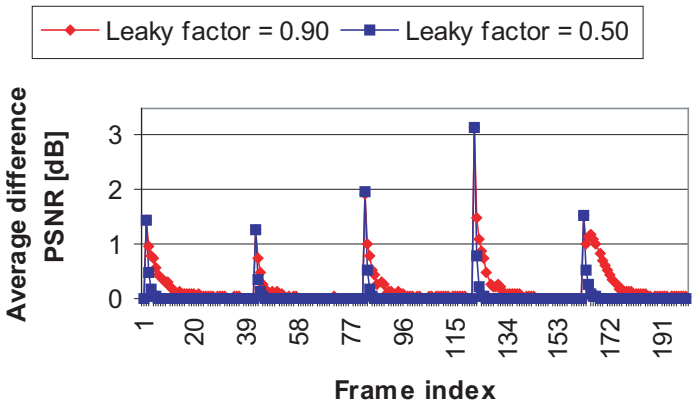


(a) Enhancement layer QP=31

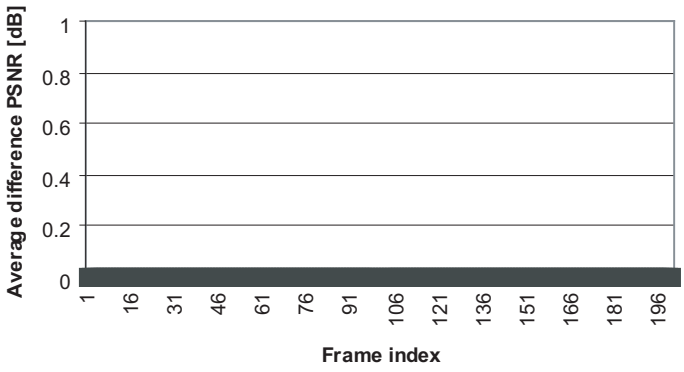


(b) Leaky factor $\alpha = 0.15$

Fig. 4.2. FS-MDC performance of *foreman* with respect to the leaky factor in the second-order predictor and the quantization step for the enhancement layer (INTRA: base layer QP=15; INTER: base layer QP=24)



(a) Description I



(b) Description II

Fig. 4.3. Error recovery capability of Dual-LPLC for *foreman* with respect to the leaky factor in the nested scalability (When the enhancement layer of the first INTER frame in each GOP is lost) (The vertical axis denotes the different PSNR of the decoded video in error from that of the intact decoded video; GOP size=80; INTER enhancement layer: QP=15; Leaky factor in parallel scalability $\alpha = 0.15$)

LAYERED SCALABLE AND LOW COMPLEXITY VIDEO ENCODING: NEW APPROACHES AND THEORETIC ANALYSIS

VOLUME 2

A Thesis

Submitted to the Faculty

of

Purdue University

by

Yuxin Liu

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

August 2004

5. LOW COMPLEXITY VIDEO ENCODING

5.1 Overview of Low Complexity Source Encoding

In this chapter, we focus on low complexity video encoding, which is developed for applications such as wireless sensor networks and distributed video surveillance systems. These systems are characterized by scarce resources for memory, computation, and energy at the video encoder but relatively abundant resources at the decoder.

Distributed source coding is a source coding paradigm that may address low complexity video encoding. Distributed source coding encodes distributed sources separately but decodes them jointly. One source symbol can be regarded as side information for the other. In distributed source coding, side information is only known to the decoder, not to the encoder. The Slepian-Wolf Theorem [102] and Wyner-Ziv Theorems have provided a theoretic basis for distributed source coding. In this section, we will briefly review the theoretic basis and practical code design for distributed source coding. We will overview state-of-the-art low complexity video encoding approaches that are implemented by the use of Slepian-Wolf and Wyner-Ziv encoding or through other techniques.

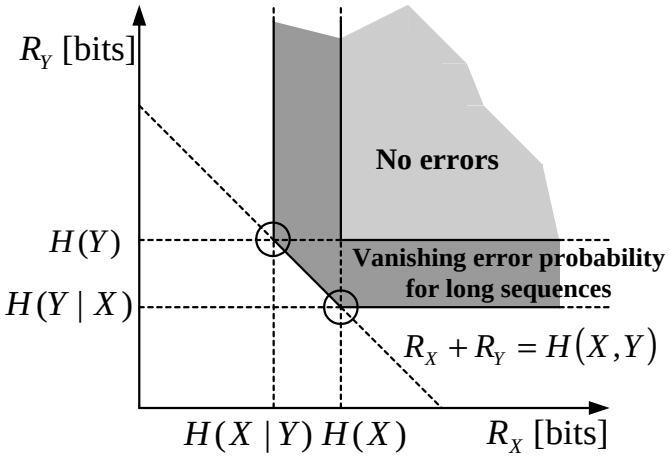


Fig. 5.1. Slepian-Wolf Theorem: theoretic basis for distributed lossless coding

5.1.1 Lossless Distributed Coding Using Slepian-Wolf Theorem

Consider a pair of source sequences X and Y, each modelled as an independent, identically distributed (i.i.d.) random sequence with a finite alphabet. Conventional lossless source coding theory claims that if X and Y are jointly encoded and jointly decoded, the achievable data rate region for error free decoding is

$$R_{X,Y} \geq H(X,Y), \tag{5.1}$$

where $H(X,Y)$ denotes the joint entropy of X and Y, and $R_{X,Y}$ represents the data rate to jointly encode X and Y [26].

The Slepian-Wolf Theorem addresses the achievable data rate region for independent encoding and joint decoding of X and Y [102]. Conventional source coding theory has shown that within the region $R_X \geq H(X)$ and $R_Y \geq H(Y)$, as depicted by the light-gray region in Fig. 5.1, the data rate combination (R_X, R_Y) is achievable

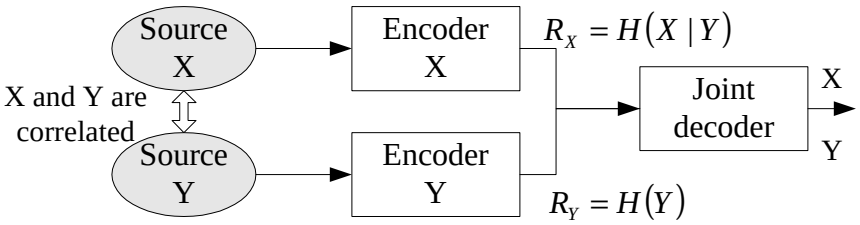


Fig. 5.2. A lossless distributed coding diagram using Slepian-Wolf Theorem

such that X and Y can be decoded with no errors. Here $H(X)$ and $H(Y)$ represent the entropy of X and the entropy of Y, respectively. Within the dark-gray region in Fig. 5.1, errors always exist in reconstructing X and Y. Nevertheless, the Slepian-Wolf Theorem shows that a vanishing error probability can be obtained if long sequences are used. Therefore, the achievable region for independent encoding and joint decoding of X and Y, if a reconstruction with an arbitrarily small error is allowed, is the combination of the two gray areas in Fig 5.1 in which

$$\begin{aligned} R_X + R_Y &\geq H(X,Y) \\ R_X &\geq H(X|Y) \\ R_Y &\geq H(Y|X), \end{aligned} \tag{5.2}$$

where $H(X|Y)$ and $H(Y|X)$ denote the conditional entropy of X given Y and the conditional entropy of Y given X, respectively.

In Fig. 5.1, the two circled points are the mostly used data rate combinations, where the data rate for the two sources, (R_X, R_Y), takes on value of $(H(X), H(Y|X))$ or $(H(X|Y), H(Y))$. For example, if X and Y are encoded at data rate $(H(X|Y), H(Y))$, i.e., source Y is encoded at data rate of its entropy but

X is encoded at data rate of its conditional entropy given Y, the total data rate is $H(X|Y) + H(Y) = H(X,Y)$, the joint entropy of X and Y. Using a joint decoder, it is shown by the Slepian-Wolf Theorem that both X and Y can be perfectly, or approximately perfectly, reconstructed, where Y serves as side information in decoding X, as shown in Fig. 5.2. Thus the Slepian-Wolf Theorem has shown that for lossless coding of two correlated sources, two independent encoders and a joint decoder may obtain the same performance as that of using a joint encoder and a joint decoder. Note that X and Y can be viewed as side information to each other.

A duality exists between Slepian-Wolf lossless coding and conventional channel coding [103]. In the case of channel coding, let X represent an arbitrary source and Y represent its channel-corrupted version. A channel encoder generates correlated redundancy bits (e.g., parity bits) from X, denoted by P. The channel decoder then uses P and Y to recover X. Analogously, for lossless distributed coding, let X represent a source symbol, and Y represent its statistically correlated side information, which is analogous to the channel-corrupted version Y in the case of channel coding. A Slepian-Wolf encoder may encode X without knowing Y and generate encoded bits P, which is analogous to the correlated redundancy bits P generated by a channel encoder. A Slepian-Wolf decoder uses P to recover X with knowledge of side information Y. Hence, a "dependence channel" may be established between a source X and its side information Y to capture their statistical correlation. Due to the analogy between Slepian-Wolf distributed coding and channel coding, a channel coding algorithm may be converted to a lossless distributed source coding algorithm.

In fact, almost all state-of-the-art practical Slepian-Wolf lossless coding algorithms were developed from the relatively more mature channel coding schemes.

Distributed source coding using syndromes, known as DISCUS, was developed in [104, 105], which was regarded as the first work in practical design of distributed source coding. The encoder of DISCUS partitions the source codewords into cosets, associates a syndrome with each coset, and encodes and transmits the syndrome, instead of the actual codeword, for the source symbol to the decoder. The decoder first identifies the coset to which the source symbol is associated from the received syndrome, and then chooses a codeword in the coset that is closest to the side information to reconstruct the source symbol. The partitioning of the source codeword space and index-labelling of the resulting cosets can be done through the framework of coset codes developed in [106, 107]. DISCUS mainly addressed the design of practical asymmetric distributed source coding. An extension of DISCUS was further developed in [108–110], where generalized cosets were designed for symmetric distributed source coding. Applications of symmetric distributed source coding to wireless sensor networks were addressed in [111, 112].

A simple example of practical distributed source coding design using DISCUS is given in Fig. 5.3 and Fig. 5.4. Let X and Y denote two 3-bit sources. Assume (1) X and Y each equally likely take on one of 8 symbols, and (2) X and Y are correlated, where the Hamming distance between X and Y is at most 1. For example, if X =[0 1 0], Y can equally likely be [0 1 0], [0 1 1], [0 0 0], or [1 1 0]. Two systems may be used to encode X when Y, serving as side information, has already been

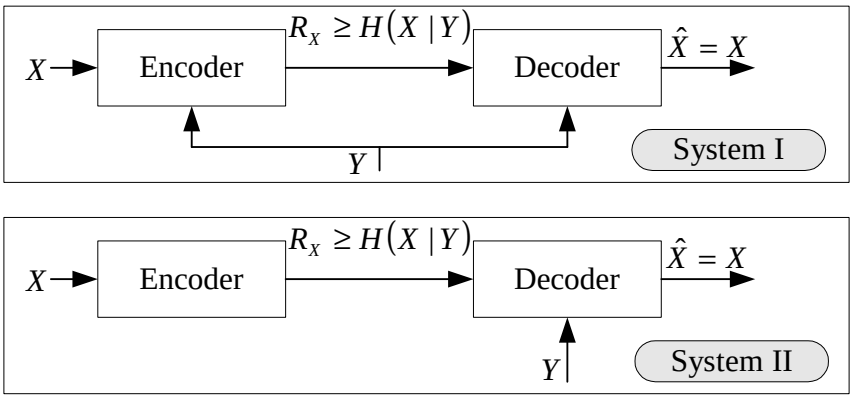


Fig. 5.3. Two systems to implement the DISCUS example

encoded and transmitted to the decoder, as shown in Fig. 5.3. System I uses the conventional source coding approach, where Y is known to both the encoder and the decoder. Due to the correlation constraint between X and Y, where

$$X + Y = \begin{cases} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{cases}, \tag{5.3}$$

2 bits are required to encode the index of $X + Y$ in order to reconstruct X at the decoder without errors.

System II, using the distributed source coding idea, where side information Y is only known to the decoder, can encode X at 2 bits per symbol. Four cosets are designed to partition the entire codeword space for X,

$$\text{Coset-1:} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \text{Coset-2:} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix},$$
$$\text{Coset-3:} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \text{Coset-4:} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \tag{5.4}$$

where the two codewords in each coset have exactly a Hamming distance of 3 in between. Accordingly, four syndromes are assigned: [0 0], [0 1], [1 1], and [1 0]. As shown in Fig. 5.4, if X takes on value [0 1 0], X belongs to Coset-3. Hence, the corresponding syndrome Z =[1 1] is transmitted to the decoder. Using the received syndrome, the decoder identifies X associated with Coset-3, implying that X may take on two possible values: [0 1 0] or [1 0 1]. With knowledge of side information Y that takes on value [1 1 0], X is finally decoded as [0 1 0]. The value [1 0 1] is discarded due to the Hamming distance constraint between the source symbol X and its side information Y.

Practical Slepian-Wolf distributed source coding design using trellis coding and lattice coding was proposed in [113], and Slepian-Wolf coding design using low-density parity check (LDPC) codes was presented in [114, 115]. In [116], irregular repeat-accumulate (IRA) was used to design joint Slepian-Wolf source coding and channel coding.

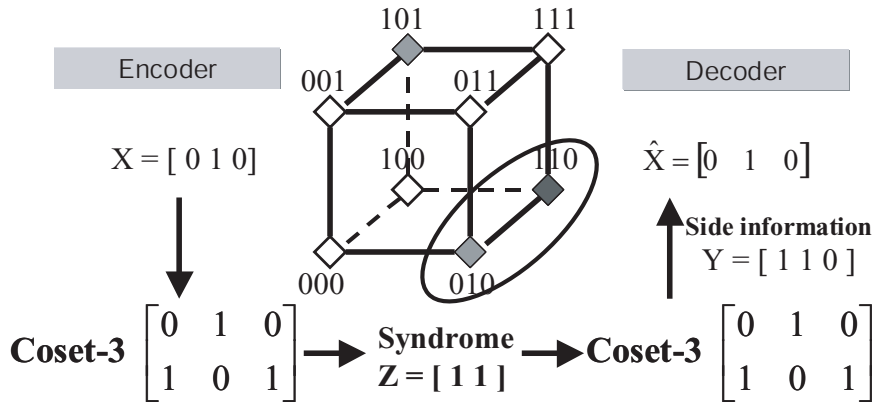


Fig. 5.4. Slepian-Wolf lossless coding using cosets and syndromes

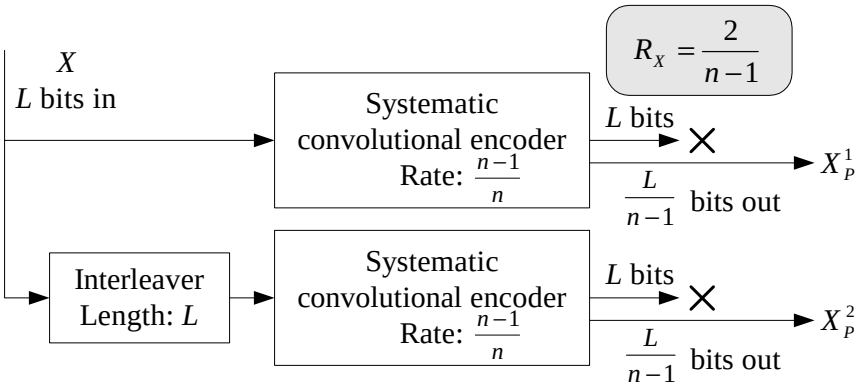


Fig. 5.5. Slepian-Wolf encoding using Turbo codes

Turbo codes, or Turbo codes in combination with other channel codes, have been widely used in the design of practical Slepian-Wolf distributed coding, including the work presented in [117–120], [121, 122], [123, 124], [125], and [126].

An example algorithm using Turbo codes is shown in Fig. 5.5 and Fig. 5.6 [126]. In this example, the systematic bits generated by the convolutional source encoders are discarded, and only the parity bits are transmitted to the decoder. Consequently, if two convolutional encoders, each with a rate $\frac{n-1}{n}$, are used, a compression ratio $\frac{2}{n-1}$ is obtained by the Slepian-Wolf source encoder that incorporates two convolutional encoders. The Slepian-Wolf decoder uses the parity bits and the side information to reconstruct the source symbol. As discussed, a "dependence channel" which characterizes the statistical correlation between the side information and the source symbol is used.

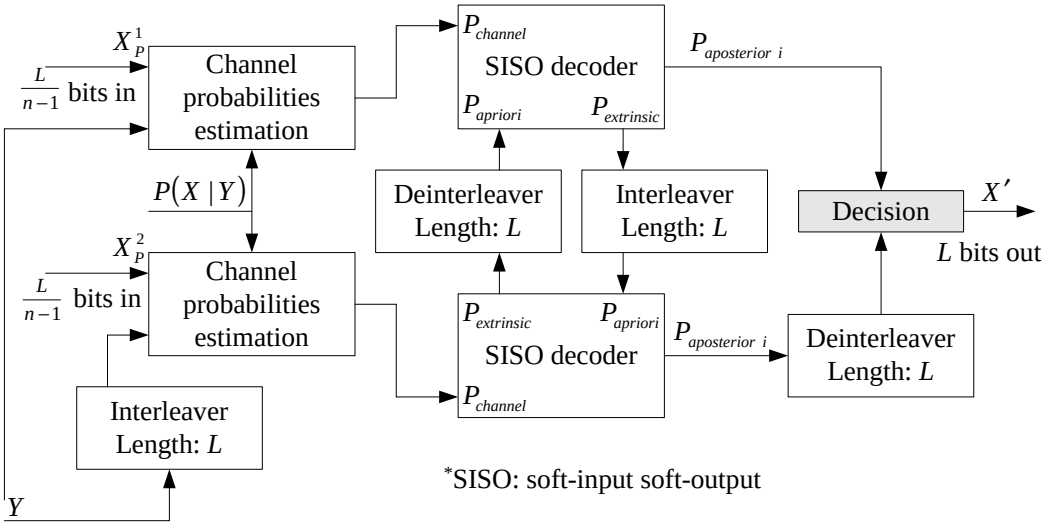


Fig. 5.6. Slepian-Wolf decoding using Turbo codes

5.1.2 Lossy Distributed Coding Using Wyner-Ziv Theorems

The Wyner-Ziv Theorems [127–129] have addressed the theoretic basis for distributed lossy coding, i.e., for lossy source coding with the side information at the decoder, as shown in Fig. 5.7. When side information is not accessible to the encoder, the Wyner-Ziv Theorems have proved that, unsurprisingly, a rate loss is incurred as follows

$$R_{X|Y}^{WZ}(D) - R_{X|Y}(D) \geq 0, \tag{5.5}$$

where $R_{X|Y}^{WZ}(D)$ represents the Wyner-Ziv rate distortion function, the achievable lower bound of a data rate to obtain a distortion D when the side information Y is not known to the encoder. $R_{X|Y}(D)$ denotes the conventional conditional rate distortion function, the achievable lower bound of a data rate to obtain the same expected distortion D when the side information Y is known to both the encoder and the decoder. Here

$$D = E\left[d(X, \hat{X})\right], \tag{5.6}$$

where $\hat{X}$ represents the reconstruction of X, $d(x, y)$ denotes a predefined distance between x and y, and $E[X]$ denotes the expectation of an arbitrary random variable X.

More importantly, it has been shown by the Wyner-Ziv Theorems that for Gaussian memoryless sources and mean-squared error (MSE) distortion, rate distortion

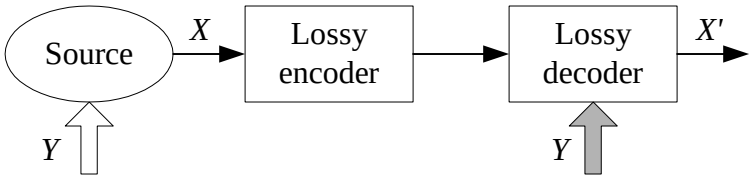


Fig. 5.7. Wyner-Ziv lossy coding with the side information at the decoder

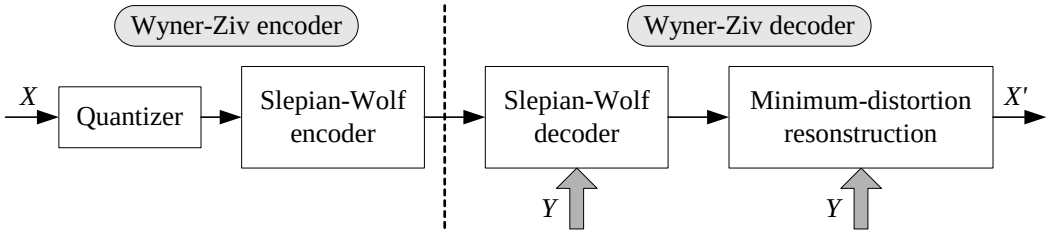


Fig. 5.8. Wyner-Ziv coding using Wyner-Ziv quantization and Slepian-Wolf coding

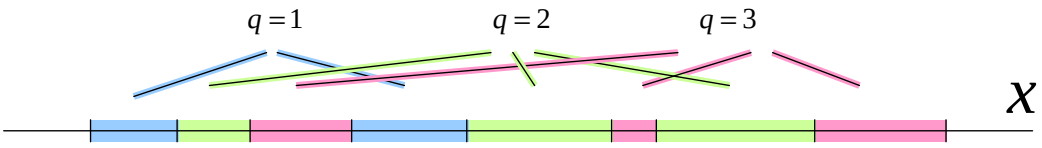


Fig. 5.9. Non-contiguous intervals for Wyner-Ziv scalar quantization

functions remain identical no matter whether the side information is known to the encoder or not. In other words, the following is achieved,

$$R_{X|Y}^{WZ}(D) = R_{X|Y}(D). \tag{5.7}$$

Further studies of the Wyner-Ziv rate distortion function were presented in [130,131] and [132].

It has been shown that under certain circumstances, linear codes and nested lattices may approach the Wyner-Ziv rate distortion bound, particularly if the source and the side information are jointly Gaussian [133, 134]. Heuristic Wyner-Ziv distributed coding design using linear codes and nested lattices were proposed in [135, 136] and [111].

Practical Wyner-Ziv coding design generally consists of a Wyner-Ziv quantizer followed by a Slepian-Wolf encoder, as shown in Fig. 5.8. Studies of Wyner-Ziv quan-

tizer design were addressed in [137, 138], [139, 140], and [141]. As shown in Fig. 5.9, a Wyner-Ziv quantizer may divide the source symbol space into non-contiguous subcells that are mapped to the same quantizer index. The Wyner-Ziv decoder then uses minimum mean-square error (MSE) reconstruction to decode the source symbol, using the statistical relation between the source symbol and its side information [141]. Fig. 5.10 shows the non-contiguous subcells that were assigned the same quantizer index by the Slepian-Wolf encoder. One subcell is finally chosen by the decoder using the minimum MSE criterion based on the conditional probability density function (p.d.f) of the source symbol X given the side information Y, namely $f_{X|Y}(x|y)$,

$$\hat{x}_{\text{opt}} = \text{argmin}_{\hat{x}} E\left[d\left(X, \hat{x}\right)|q, y\right], \tag{5.8}$$

where $\hat{x}$ denotes the centroid of each subcell under consideration and $\hat{x}_{\text{opt}}$ represents the centroid of the finally selected subcell. $d(x, y)$ denotes the $\mathbb{L}^2$-norm Euclidean distance between x and y, q denotes the quantizer index specified by the decoder, and $E[X|Y]$ denotes the conditional expectation of the random variable X given Y. A Wyner-Ziv quantizer design algorithm is given in Fig. 5.11 [138] [141], which uses a generalization of the Lloyd algorithm that has been used in the conventional source coding quantizer design [142].

Orthogonal transforms have been widely used in conventional source coding, where the quantization and entropy coding both operate on the transform coefficients. Studies of Wyner-Ziv transform coding were proposed in [143] and [144]. An example of Wyner-Ziv transform coding is shown in Fig. 5.12 [144].

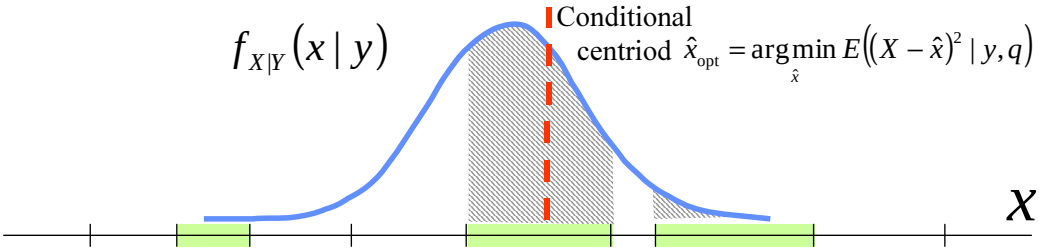


Fig. 5.10. Minimum mean-squared error (MSE) reconstruction with side information

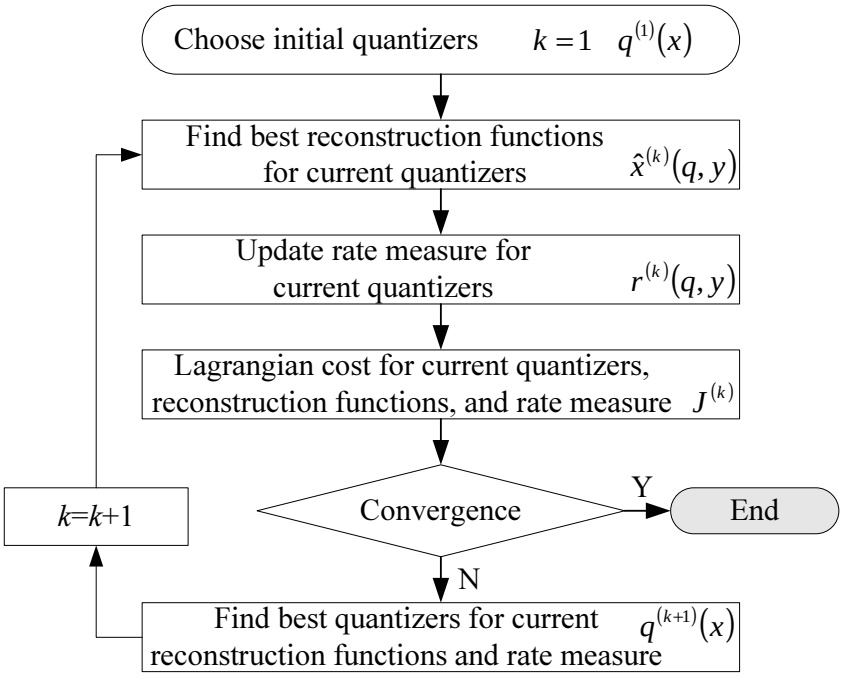


Fig. 5.11. Lloyd algorithm for Wyner-Ziv quantizer design

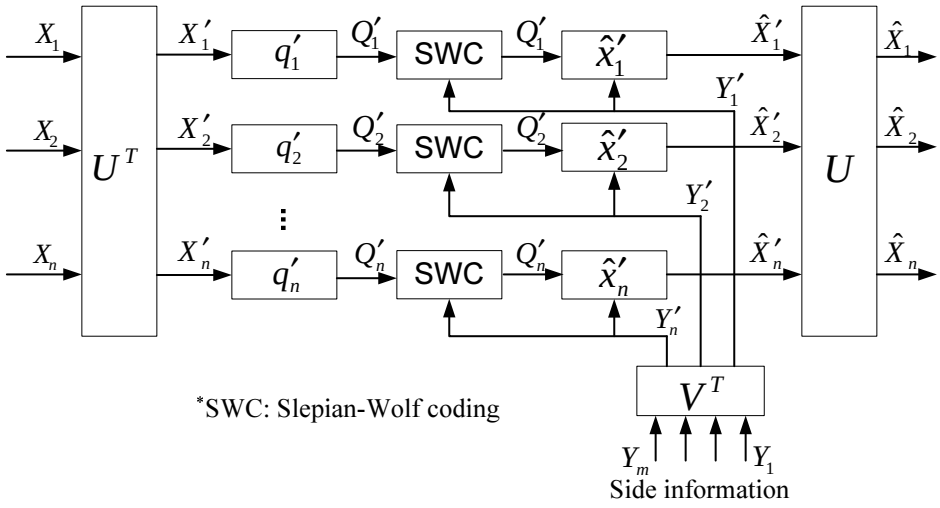


Fig. 5.12. Wyner-Ziv transform coding

In Fig. 5.12, given a source symbol X, an orthogonal transform, denoted U, is first applied to X to obtain the transform coefficients $X' = U^T X$. Each component of X', denoted by X'_i, individually undergoes its own Wyner-Ziv quantization and Slepian-Wolf coding (denoted by SWC in the figure). Accordingly, the side information, Y, also undergoes an orthogonal transform, denoted by matrix V. The transform coefficients, $Y' = V^T Y$, are used by both the Slepian-Wolf decoder and the Wyner-Ziv dequantizer for each individual component X'_i, to obtain the reconstruction of the transform coefficients X', namely $\hat{X}'$. The inverse transform, $U^{-1} = U^T$, is then used to obtain the final reconstruction of the source symbol $\hat{X}$, where $\hat{X} = U\hat{X}'$.

5.1.3 Low Complexity Video Encoding Using Wyner-Ziv

As discussed in Section 1.1.4 of Chapter 1, conventional video coding methods are highly asymmetrical, where the encoder typically requires 5 to 10 times more computational complexity than the decoder. If the motion estimation, which contributes most of the computational complexity of the encoder, could be shifted from the encoder to the decoder, a low complexity video encoder could be obtained. A video coder may implement intra-frame encoding but inter-frame decoding such that the motion vectors serving as side information are only used by the decoder. Wyner-Ziv coding has provided such a coding paradigm that may achieve a rate distortion performance similar to that of using conventional video coding approaches.

Low complexity in both ends of a video coding system can be achieved by the use of a transcoder, as shown in Fig. 5.13. With the transcoder, the decoded video from

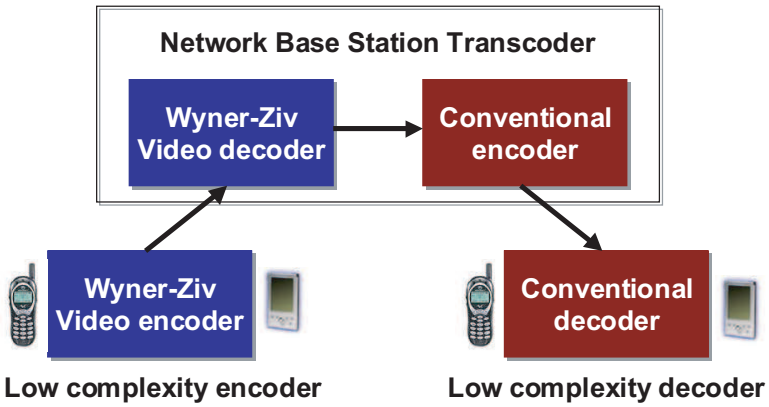


Fig. 5.13. Low complexity in both ends

the Wyner-Ziv decoder is re-encoded using a conventional video encoder. The video communication system with low complexity in both ends is particularly appealing to applications such as video messaging and video telephony with mobile terminals at both the transmitter and the receiver.

The design of low complexity video encoding includes Wyner-Ziv systematic design plus Slepian-Wolf lossless coding design. The work presented in [145–149] mainly uses Turbo codes in the design of Wyner-Ziv video coding, with an example codec given in Fig. 5.14. Using the scheme shown in the figure, the input video frames are first partitioned into two groups, each encoded using a different coding method. The frames serving as "key frames" are intra-encoded using conventional source coding methods, whereas the remaining frames are encoded by a Slepian-Wolf codec using Turbo codes. Motion estimation is performed for the reconstructed key frames at the decoder. The motion vectors are used to obtain the side information using motion compensated interpolation/extrapolation. The Turbo decoder of the Slepian-Wolf

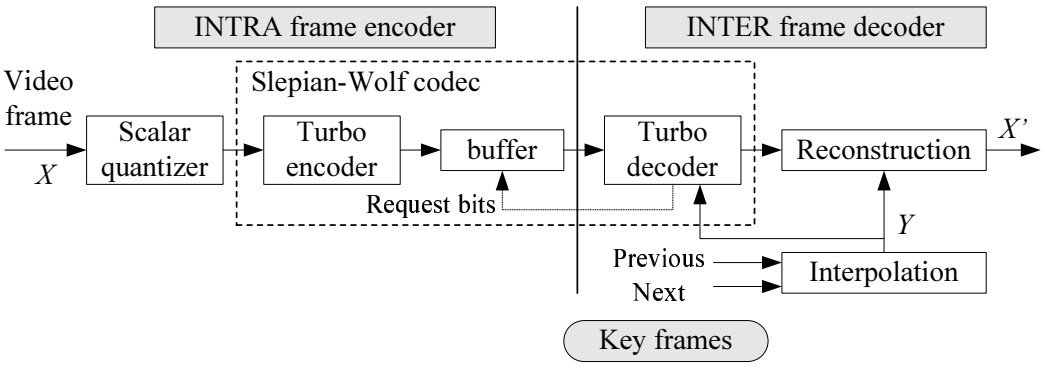


Fig. 5.14. Wyner-Ziv video coding using a Turbo coder

codec uses the side information to reconstruct the remaining frames. A rate control mechanism is built inside the Slepian-Wolf codec, where the Turbo decoder may request bits in real time from a buffer associated with the encoder through a feedback channel. It is seen that this video coding method allows flexible decoder side information, where more accurate motion compensation at the decoder does not increase the encoding data rate. With knowledge of the motion vectors obtained from the key frames, several interpolation/extrapolation schemes may be used in obtaining the side information:

- Motion-compensated interpolation;
- Average interpolation;
- Motion-compensated extrapolation;
- Previous frame extrapolation;
- Cyclic redundancy check (CRC) or robust hash coding aided motion estimation/compensation [146].

(a) Side information (b) Reconstruction using side information

Fig. 5.15. Example of Wyner-Ziv decoding with side information

An example of Wyner-Ziv decoded video frames is given in Fig. 5.15, where the Wyner-Ziv method that uses Turbo codes as shown in Fig. 5.14 is used. In Fig. 5.15, the left image is the side information obtained using motion-compensated interpolation, and the right image is the reconstructed video frame after Wyner-Ziv decoding using the side information. It is observed that symbol errors resulting from the Slepian-Wolf decoder are manifested in the form of isolated flashing pixels in the reconstructed video. It has been discussed in [145] that these flashing pixels may be avoided in practical systems.

A Wyner-Ziv video coding paradigm using syndromes was proposed in [150–153], which has been referred to as PRISM (Power-efficient, Robust, hIgh-compression, Syndrome-based Multimedia coding). The PRISM encoder and decoder are shown in Fig. 5.16 and Fig. 5.17. The rate control scheme of PRISM does not require a feedback channel, which makes it more suitable for storage applications.

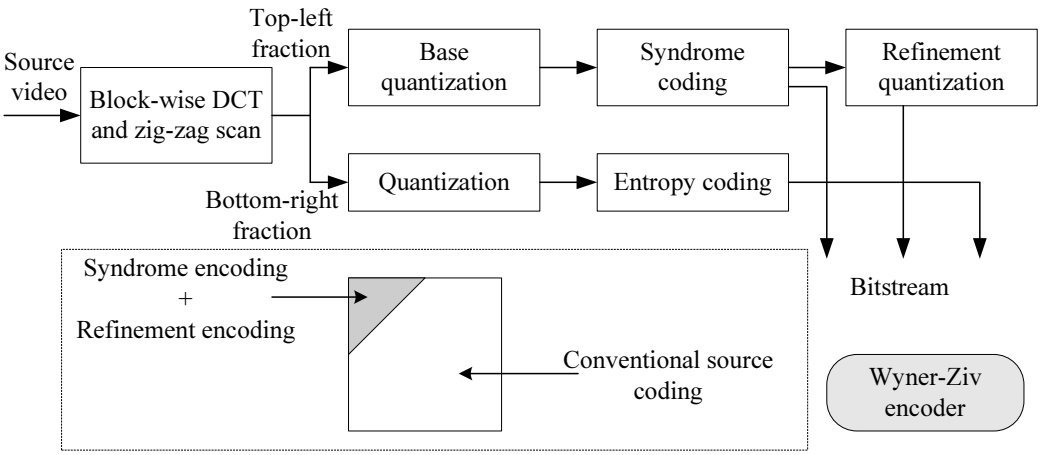


Fig. 5.16. Wyner-Ziv encoding using PRISM

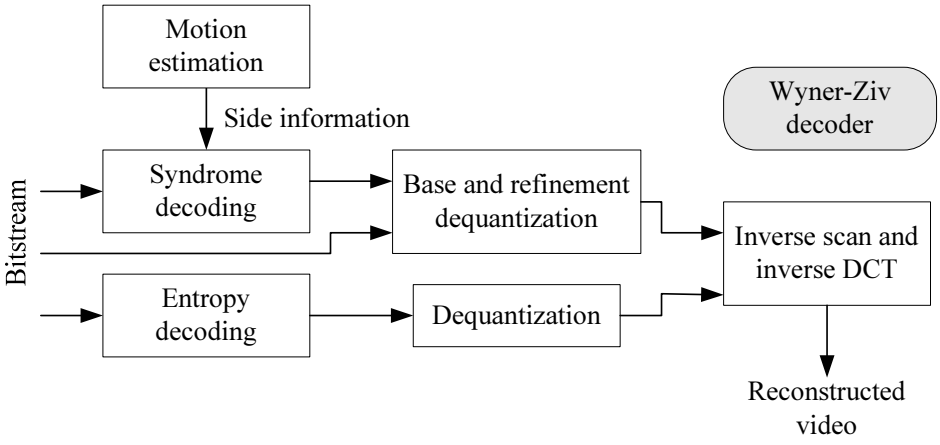


Fig. 5.17. Wyner-Ziv decoding using PRISM

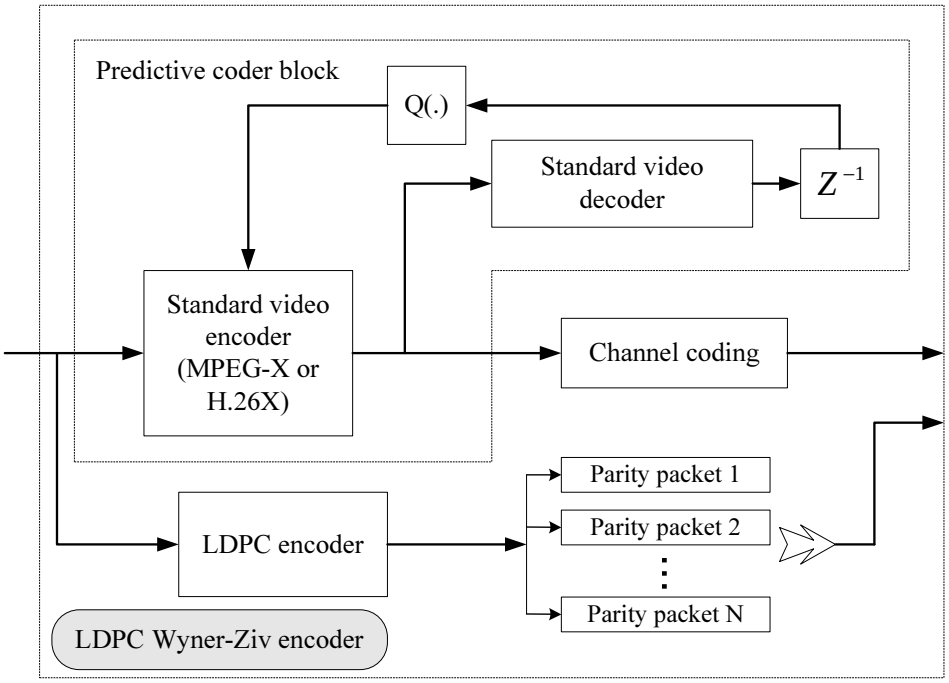


Fig. 5.18. LDPC state-free video encoding

Wyner-Ziv video coding using LDPC and coset coding was proposed in [154–157], with the encoder diagram shown in Fig. 5.18. This method provides a state-free functionality for predictive video coding, which helps alleviate the drift errors in video reconstruction when the encoder and the decoder lose their synchronization.

Layered and embedded Wyner-Ziv coding have been addressed in [145], [158], and [159, 160]. An embedded Wyner-Ziv video coding scheme is shown in Fig. 5.19 [145], where the reconstructed video from lower-layer Wyner-Ziv decoders can be used as side information for higher-layer decoders. Hence, a graceful degradation in reconstruction errors can be achieved without a layered signal representation.

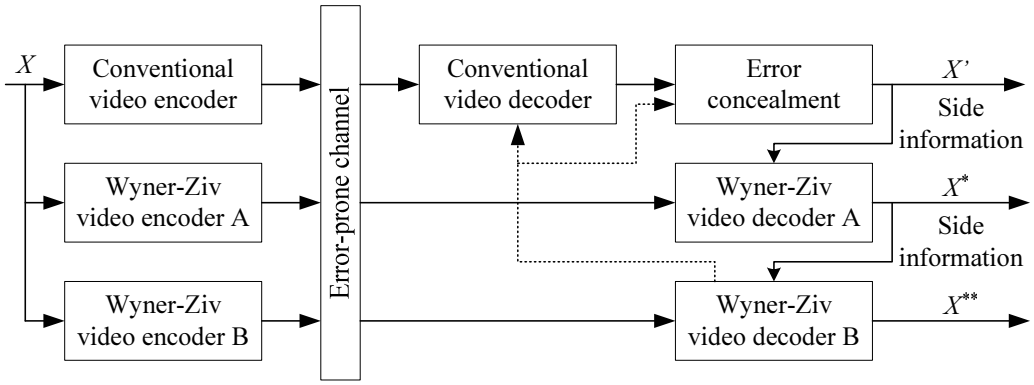


Fig. 5.19. Embedded Wyner-Ziv video coding

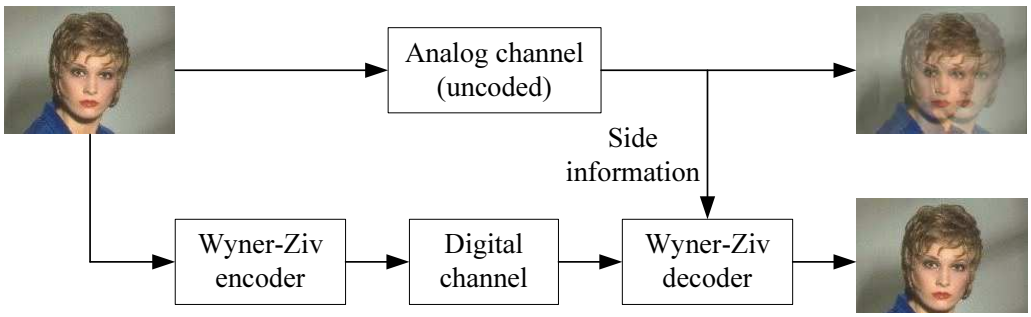


Fig. 5.20. Signal enhancement with side information

Wyner-Ziv coding applied to multiview image acquisition was addressed in [161]. Instead of motion compensated interpolation, three-dimensional (3D) geometrical models were used to extract the side information from the "key frames" or from frames already decoded by the Wyner-Ziv decoder. Signal enhancement with side information in upgrading NTSC to HDTV using a digital side-channel was presented in [162], with an example shown in Fig. 5.20.

State-of-the-art Wyner-Ziv video coding methods generally achieve a rate distortion performance somewhere between that of the intra-frame coding and the inter-frame coding using conventional video coding schemes. An example of the rate distortion performance using the Wyner-Ziv video coding method given in Fig. 5.14 is shown in Fig. 5.21 [149].

Due to the analogy between channel coding and Slepian-Wolf coding, as discussed in Subsection 5.1.1, Wyner-Ziv coding has an intrinsic capability of error resilience. In essence Wyner-Ziv coding uses the statistical relation between the source symbol and the side information, where the relation can be characterized by a hypotheti-

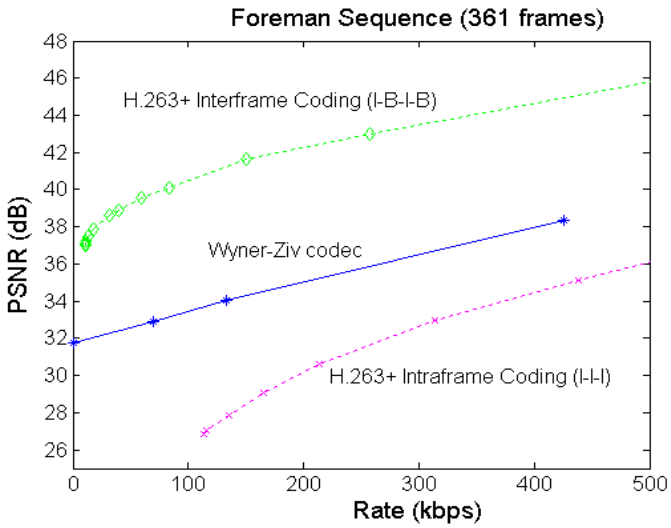


Fig. 5.21. Rate distortion performance of Wyner-Ziv video coding

cal dependence channel connecting the source and the side information. Hence, a Wyner-Ziv codec is not only capable of fulfilling the source coding task, but also possesses a channel-coding-like functionality. The Wyner-Ziv decoder may either use a stronger Slepian-Wolf code, implemented by schemes such as requiring more parity bits from the Slepian-Wolf encoder, or create more accurate side information. In either way, the Wyner-Ziv decoder is not only able to correct the discrepancies of the dependance channel, but also capable of helping correct the errors introduced during transmission of the source bitstream. It has been shown that forward error protection (FEP) realized by embedded Wyner-Ziv video coding, using the scheme presented in Fig 5.19, is more effective than forward error correction (FEC) realized by traditional channel coding schemes, especially in the scenario of high symbol error rate, as shown in Fig. 5.22 [163]. Studies of error resilience algorithm design using Wyner-Ziv and corresponding performance analysis have been presented in [163–165], [150, 151], and [154–156].

5.1.4 Other Low Complexity Video Encoding Approaches

Most state-of-the-art low complexity video encoding methods were developed based on the Wyner-Ziv coding paradigm. Other low complexity image and video encoding approaches have also been explored, including the work presented in [166], where algorithms for massively distributed image compression were studied in application to wireless sensor networks. The compact support of the wavelet transform

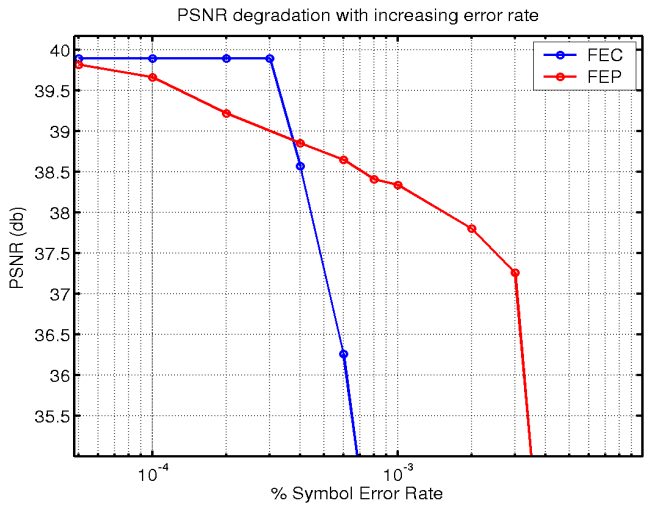


Fig. 5.22. Error resilience performance of Wyner-Ziv video coding

was used to decorrelate the source data while keeping the costly inter-communication among sensors at a minimum.

A low complexity video encoding method using network-driven motion estimation (NDME) was proposed in [167]. Using NDME, motion estimation is implemented at the decoder, while motion vectors are sent back to the encoder through a feedback channel. Fundamentally, NDME provides three coding modes to encode each macroblock of a given arbitrary video frame: intra coding mode, conditional replenishment (CR) mode, and forward motion compensation mode. CR is equivalent to motion compensation using zero-valued motion vectors. In contrast to conventional forward motion compensation, NDME obtains the forward motion vectors for each decoded frame at the decoder, with the previously decoded frame as the reference. The encoder then assumes the same motion information for consecutive frames and uses the motion vectors received from the decoder for the preceding frame to obtain the motion compensated prediction for the current frame. Once the new encoded frame is decoded, the forward motion vectors are refreshed using motion estimation applied to the new decoded frame and sent back to the encoder for encoding the next frame. NDME has demonstrated a competitive rate distortion performance, which is slightly worse than that of using forward motion estimation at the encoder, and much better than that of only using the CR and intra coding modes.

In summary, compared to conventional video encoding, low complexity video encoding is fundamentally a new coding paradigm. The Wyner-Ziv and Slepian-Wolf theorems have provided a theoretic basis for low complexity source encoding.

Nevertheless, practical algorithm design that approaches the theoretic bound still needs to be further explored. Adaptive and universal approaches have not been developed. Future studies may also involve the joint source and channel coding design coupled with appropriate network protocol design. In addition to the exploration of practical systematic design, it is also worth exploring the new insights provided by low complexity video encoding to conventional video coding. A combination or joint design of conventional video coding and low complexity video encoding would be beneficial in further improving the performance of video coding in terms of both coding efficiency and error resilience.

5.2 Low Complexity Video Encoding Using B-Frame Direct Modes

In this section, we describe a new low complexity video encoding approach using B-frame direct modes. The direct mode was originally developed for encoding the bidirectionally-coded frames, i.e., B frames, where the motion vectors for a B frame are obtained by interpolation from the motion vectors of the subsequent predictive-coded frame, i.e., P frame. The motion compensated prediction of the direct mode is a linear combination of the two predictions obtained from the forward motion compensation and the backward motion compensation. If a B frame is encoded using the direct mode, no motion vectors are needed to be encoded and transmitted. The decoder simply retrieves the motion vectors of the B frame using the same motion-vector interpolation procedure.

In our proposed approach, we extend the direct-mode idea and design new direct coding modes to encode B frames. All the new direct coding modes derive the motion vectors for any B frame from the neighboring frames. Hence, no motion estimation is needed to encode a B frame by the use of any of the direct coding modes. As discussed at the beginning of this chapter, motion estimation contributes to most of the computational complexity of a conventional video encoder. Therefore, we can obtain a low complexity video encoder if we constrain any macroblock in a B frame to be encoded only using either the intra-coding mode or one of the new direct coding modes.

Our proposed low complexity video encoding requires a feedback channel from the decoder to the encoder. We implement motion estimation at the decoder and transmit the motion vectors back to the encoder. This is similar to the NDME approach discussed in Subsection 5.1.4 [167]. In contrast to NDME where only forward motion estimation/compensation is used, we use bidirectional motion estimation and motion compensation, which provides more coding modes and a more graceful rate distortion performance. Our scheme can be further coupled with NDME or other low complexity video encoding approaches such as the ones using Wyner-Ziv to provide a practical and efficient low complexity video encoding system.

As shown in Fig. 5.23, we demonstrate a solution for video surveillance using our proposed low complexity video encoding method. The camera arrays are distributed and may be only allowed to have wireless connections. If a camera is triggered by a built-in sensor to start capturing motion events, the low complexity video

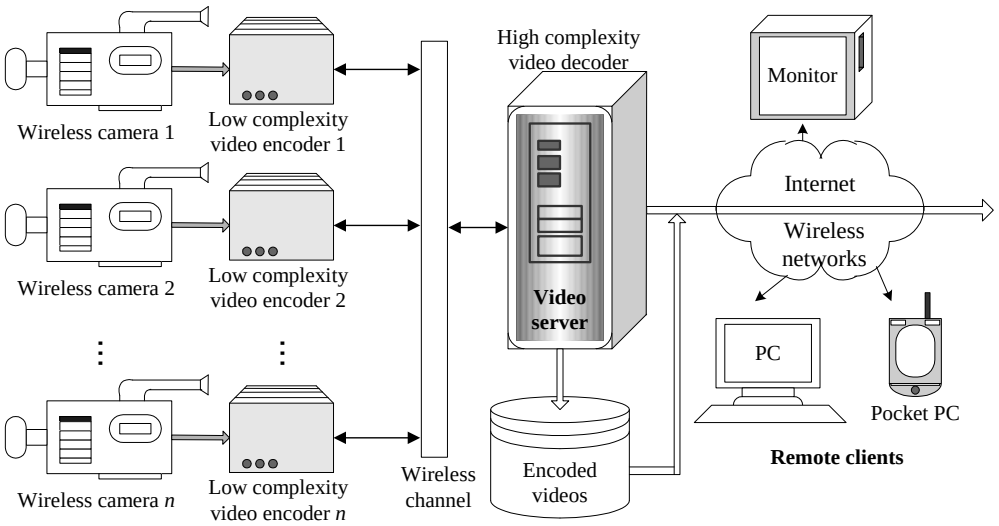


Fig. 5.23. A video surveillance system using low complexity video encoding

encoder associated with the camera may turn on a two-way communication with the video server and start the real-time encoding of whatever videos have been captured. The video server may serve as a high complexity video decoder and provide the information demanded by the low complexity video encoder through the feedback channel. The encoded video bitstreams are transmitted to the video server and stored in the video database. The stored video clips may be streamed by a user-driven mechanism to the remote clients. Real-time video streaming from the video server to the remote users is also possible.

5.2.1 An Introduction to Conventional B-Frame Direct Mode

The direct mode is designed to encode macroblocks in the B frames. The direct mode has been included in recent video coding standards such as MPEG-4 and H.264 [45]. We refer to this direct mode that is specified in the standard as the conventional B-frame direct mode, or *B direct mode I*.

B direct mode I - the direct mode that uses the forward motion vectors pointing from P to P: If a macroblock (MB) is encoded in the conventional B-frame direct mode, its motion vectors are obtained by interpolation using the motion vectors of the co-located macroblock in the subsequent P frame, as shown in Fig. 5.24. Here a co-located macroblock is referred to the macroblock located in another frame that is exactly in the same position as the current macroblock that is under consideration. Hence no motion vector information is needed to encode and transmit for the mac-

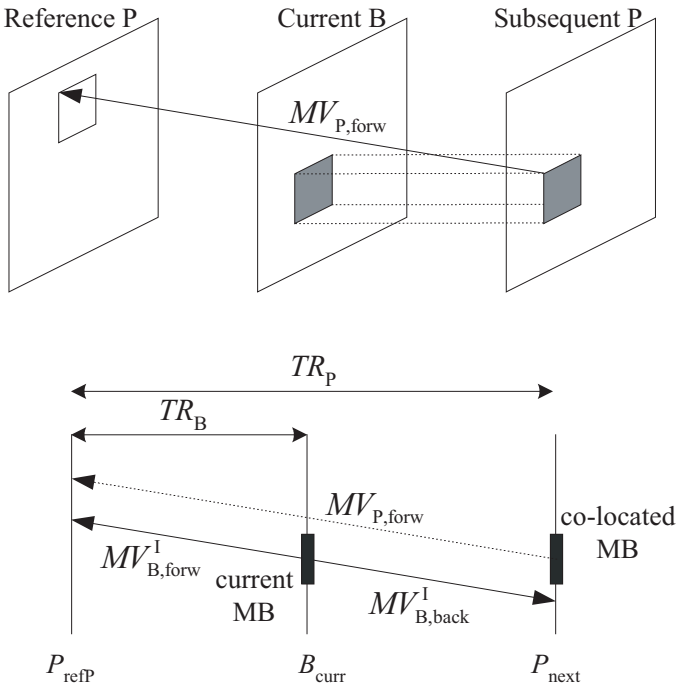


Fig. 5.24. *B direct mode I*: using the forward motion vectors pointing from P to P

roblock, and the decoder simply retrieves the same motion vectors using the same interpolation process.

In Fig. 5.24, $MV_{\mathrm{P,forw}}$ represents the forward motion vector of the co-located macroblock in the subsequent P frame referred to as P_{next}. $MV_{\mathrm{P,forw}}$ points from P_{next} to the reference frame of the co-located macroblock, namely P_{refP}. Note that the reference frame could be an intra-coded frame, i.e., I frame. TR_{P} denotes the temporal distance between P_{next} and P_{refP}, and TR_{B} represents the temporal distance between the current B frame, referred to as B_{curr}, and P_{refP}. Using *B direct mode I*,

the forward and the backward motion vectors of the current macroblock, denoted as $MV^{\mathrm{I}}_{\mathrm{B,forw}}$ and $MV^{\mathrm{I}}_{\mathrm{B,back}}$ respectively, are derived by interpolation as follows

$$MV^{\mathrm{I}}_{\mathrm{B,forw}} = \frac{TR_{\mathrm{B}}}{TR_{\mathrm{P}}} MV_{\mathrm{P,forw}}, \tag{5.9}$$

$$MV^{\mathrm{I}}_{\mathrm{B,back}} = \frac{TR_{\mathrm{B}} - TR_{\mathrm{P}}}{TR_{\mathrm{P}}} MV_{\mathrm{P,forw}}. \tag{5.10}$$

Two motion compensated predictions of the current macroblock are obtained using $MV^{\mathrm{I}}_{\mathrm{B,forw}}$ and $MV^{\mathrm{I}}_{\mathrm{B,back}}$, which are referred to as $M^{(p),\mathrm{I}}_{\mathrm{forw}}$ and $M^{(p),\mathrm{I}}_{\mathrm{back}}$ respectively. In video coding standards such as H.26L, the prediction for the current macroblock is finally obtained by simply averaging the two predictions with equal weights. In the most recent video coding standard H.264/AVC, the direct mode is improved by obtaining the prediction using a weighted superposition as follows [168]

$$M^{(p),\mathrm{I}}_{\mathrm{Bidir}} = \frac{TR_{\mathrm{P}} - TR_{\mathrm{B}}}{TR_{\mathrm{P}}} M^{(p),\mathrm{I}}_{\mathrm{forw}} + \frac{TR_{\mathrm{B}}}{TR_{\mathrm{P}}} M^{(p),\mathrm{I}}_{\mathrm{back}}. \tag{5.11}$$

It has been shown this improved direct mode is specially effective in encoding videos with fading in/fading out scenes which frequently occur in music videos or movie trailers.

5.2.2 New B-Frame Direct Modes Using Feedback from the Decoder

If a feedback channel is available, motion estimation may be implemented at the decoder and the motion vectors can be transmitted back to the encoder. The motion vectors that are obtained from the already encoded frames can then be used to interpolate/extrapolate the motion vectors for the frames to be encoded. Using

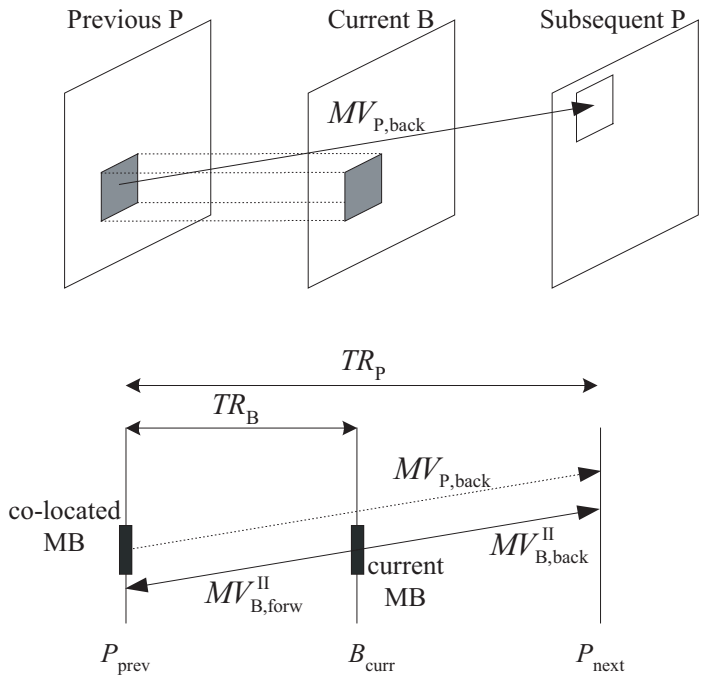


Fig. 5.25. *B direct mode II*: using the backward motion vectors pointing from P to P

this idea we design two new direct modes for encoding B frames: *B direct mode II* and *B direct mode III*.

B direct mode II - the direct mode that uses the backward motion vectors pointing from P to P: It is noted that at the time the current B frame is to be encoded, the subsequent P frame P_{next} and the previous P frame (or I frame) P_{prev} have already been encoded and transmitted to the decoder. Hence the process of motion estimation using P_{next} as the reference frame can be performed for each macroblock in P_{prev} at the decoder, obtaining the motion vectors pointing backward from P_{prev} to P_{next}, as shown in Fig. 5.25.

In Fig. 5.25, $MV_{\mathrm{P,back}}$ represents the motion vector of the co-located macroblock in P_{prev}, which is transmitted back to the encoder and used to interpolate the bidirectional motion vectors of the current macroblock, $MV^{\mathrm{II}}_{\mathrm{B,forw}}$ and $MV^{\mathrm{II}}_{\mathrm{B,back}}$, as follows

$$MV^{\mathrm{II}}_{\mathrm{B,forw}} = -\frac{TR_{\mathrm{B}}}{TR_{\mathrm{P}}} MV_{\mathrm{P,back}}, \tag{5.12}$$

$$MV^{\mathrm{II}}_{\mathrm{B,back}} = \frac{TR_{\mathrm{P}} - TR_{\mathrm{B}}}{TR_{\mathrm{P}}} MV_{\mathrm{P,back}}. \tag{5.13}$$

The bidirectional motion compensated prediction using *B direct mode II*, denoted as $M^{(p),\mathrm{II}}_{\mathrm{Bidir}}$, is obtained in the same way as Eqn. (5.11),

$$M^{(p),\mathrm{II}}_{\mathrm{Bidir}} = \frac{TR_{\mathrm{P}} - TR_{\mathrm{B}}}{TR_{\mathrm{P}}} M^{(p),\mathrm{II}}_{\mathrm{forw}} + \frac{TR_{\mathrm{B}}}{TR_{\mathrm{P}}} M^{(p),\mathrm{II}}_{\mathrm{back}}, \tag{5.14}$$

where $M^{(p),\mathrm{II}}_{\mathrm{forw}}$ and $M^{(p),\mathrm{II}}_{\mathrm{back}}$ denote the two motion compensated predictions using $MV^{\mathrm{II}}_{\mathrm{B,forw}}$ and $MV^{\mathrm{II}}_{\mathrm{B,back}}$.

Note that the temporal distances, TR_{P} and TR_{B}, may be different for *B direct mode II* compared to those in *B direct mode I*, since the reference frame of P_{next} may not be the encoded I or P frame preceding P_{next}, meaning that P_{refP} may be different from P_{prev}. This is true when the feature of multiple-reference-frame or multihypothesis is used for the inter-coded frames, which has been included in the video coding standard H.26L/H.264 [45].

B direct mode III - the direct mode that uses the bidirectional motion vectors pointing from B to P: If more than one B frames are inserted between consecutive P frames, for example, if the GOP (group of pictures) is coded using the PBBPBB pattern, the motion vectors of the first B frame, referred to as B_{prev}, can be used to interpolate/extrapolate the motion vectors for the second B frame. The bidirectional

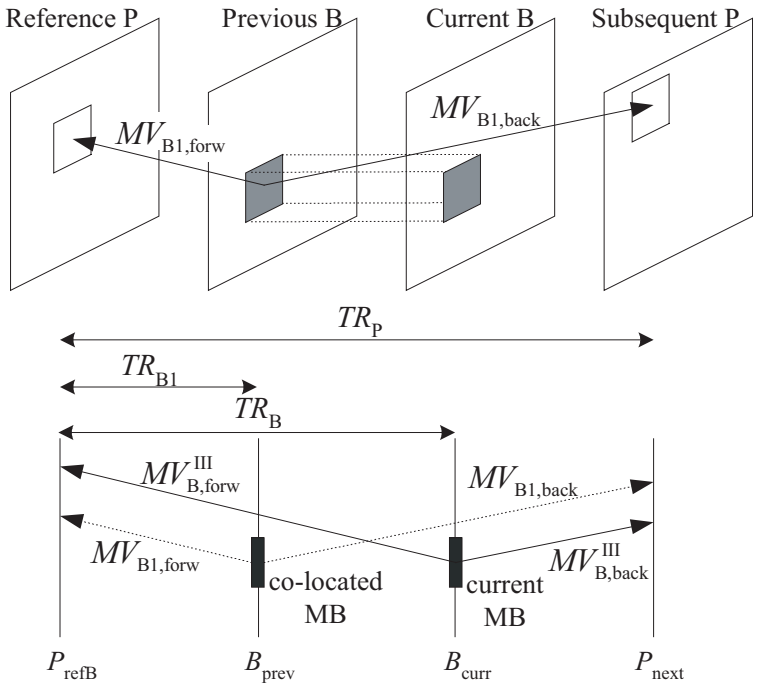


Fig. 5.26. *B direct mode III*: using the bidirectional motion vectors pointing from B to P

motion estimation for each macroblock in B_{prev} is performed at the decoder, as shown in Fig. 5.26. The frame P_{next} is used as the reference frame for the backward motion estimation, and P_{refB} denotes the reference frame used for the forward motion estimation. Again, if the multihypothesis mode is used, P_{refB} may not be the same as P_{refP}, the reference frame of the subsequent P frame, neither does it have to be the same as P_{prev}, the preceding P frame (or I frame) of the current B frame.

The obtained motion vectors of B_{prev}, pointing to P_{next} and P_{refB} respectively, are then transmitted back to the encoder. The bidirectional motion vectors of the co-located macroblock in B_{prev}, namely $MV_{\text{B1,forw}}$ and $MV_{\text{B1,back}}$, are used to derive

the bidirectional motion vectors of the current macroblock using interpolation and extrapolation as follows

$$MV^{\mathrm{III}}_{\mathrm{B,forw}} = \frac{TR_{\mathrm{B}}}{TR_{\mathrm{B1}}} MV_{\mathrm{B1,forw}}, \tag{5.15}$$

$$MV^{\mathrm{III}}_{\mathrm{B,back}} = \frac{TR_{\mathrm{P}} - TR_{\mathrm{B}}}{TR_{\mathrm{P}} - TR_{\mathrm{B1}}} MV_{\mathrm{B1,back}}, \tag{5.16}$$

where TR_{B1} denotes the temporal distance between B_{prev} and its reference frame for the forward prediction P_{refB}.

Again, the bidirectional motion compensated prediction using *B direct mode III*, denoted as $M^{(p),\mathrm{III}}_{\mathrm{Bidir}}$, is obtained in the same way as the previous two direct modes

$$M^{(p),\mathrm{III}}_{\mathrm{Bidir}} = \frac{TR_{\mathrm{P}} - TR_{\mathrm{B}}}{TR_{\mathrm{P}}} M^{(p),\mathrm{III}}_{\mathrm{forw}} + \frac{TR_{\mathrm{B}}}{TR_{\mathrm{P}}} M^{(p),\mathrm{III}}_{\mathrm{back}}, \tag{5.17}$$

where $M^{(p),\mathrm{III}}_{\mathrm{forw}}$ and $M^{(p),\mathrm{III}}_{\mathrm{back}}$ denote the two motion compensated predictions using $MV^{\mathrm{III}}_{\mathrm{B,forw}}$ and $MV^{\mathrm{III}}_{\mathrm{B,back}}$.

We summarize the three B-frame direct modes in Table 5.1. If the GOP structure takes on the pattern PBBPBB, we have six direct coding modes for coding each macroblock in the first B frame between consecutive P(I) frames, namely PPForw_Bidir, PPForw_Forw, PPForw_Back, PPBack_Bidir, PPBack_Forw, and PPBack_Back, as given in Table 5.2. For the macroblocks in the second B frame within the two consecutive P(I) frames, three more direct coding modes are provided using the third B-frame direct mode, namely PBBidir_Bidir, PBBidir_Forw, and PBBidir_Back. Note that the first direct coding mode, PPForw_Bidir, which uses the bidirectional motion compensated prediction $M^{(p),\mathrm{I}}_{\mathrm{Bidir}}$ as given in Eqn. (5.11), is the conventional B-frame direct mode specified in the video coding standard such as H.26L/H.264 [169].

Table 5.1
Three B-frame direct modes (GOP in pattern IBBPBB; B_1 representing the first B frame and B_2 the second B frame between successive P(I) frames)

Direct mode	Frm to encode	Frame of co-loc. MB	MV(s) used	Ref. frame	MVs derived	Pred. derived
B direct mode I	B_{curr} (B_1/B_2)	P_{next}	$MV_{\text{P,forw}}$	P_{refP}	$MV^{\text{I}}_{\text{B,forw}}$	$M^{(p),\text{I}}_{\text{Bidir}}$
					$MV^{\text{I}}_{\text{B,back}}$	$M^{(p),\text{I}}_{\text{forw}}$
						$M^{(p),\text{I}}_{\text{back}}$
B direct mode II	B_{curr} (B_1/B_2)	P_{prev}	$MV_{\text{P,back}}$	P_{next}	$MV^{\text{II}}_{\text{B,forw}}$	$M^{(p),\text{II}}_{\text{Bidir}}$
					$MV^{\text{II}}_{\text{B,back}}$	$M^{(p),\text{II}}_{\text{forw}}$
						$M^{(p),\text{II}}_{\text{back}}$
B direct mode III	B_{curr} (B_2)	B_{prev}	$MV_{\text{B,forw}}$	P_{refB}	$MV^{\text{III}}_{\text{B,forw}}$	$M^{(p),\text{III}}_{\text{Bidir}}$
			$MV_{\text{B,back}}$	P_{next}	$MV^{\text{III}}_{\text{B,back}}$	$M^{(p),\text{III}}_{\text{forw}}$
						$M^{(p),\text{III}}_{\text{back}}$

Table 5.2
Three B-frame direct modes and nine direct coding modes for B frames

Direct coding mode	Motion compensated prediction	B-frame direct mode
PPForw_Bidir	$M_{\text{Bidir}}^{(p),\text{I}} = \frac{TR_{\text{P}} - TR_{\text{B}}}{TR_{\text{P}}} M_{\text{forw}}^{(p),\text{I}} + \frac{TR_{\text{B}}}{TR_{\text{P}}} M_{\text{back}}^{(p),\text{I}}$	*B direct mode I*
PPForw_Forw	$M_{\text{forw}}^{(p),\text{I}}$	
PPForw_Back	$M_{\text{back}}^{(p),\text{I}}$	
PPBack_Bidir	$M_{\text{Bidir}}^{(p),\text{II}} = \frac{TR_{\text{P}} - TR_{\text{B}}}{TR_{\text{P}}} M_{\text{forw}}^{(p),\text{II}} + \frac{TR_{\text{B}}}{TR_{\text{P}}} M_{\text{back}}^{(p),\text{II}}$	*B direct mode II*
PPBack_Forw	$M_{\text{forw}}^{(p),\text{II}}$	
PPBack_Back	$M_{\text{back}}^{(p),\text{II}}$	
PBBidir_Bidir	$M_{\text{Bidir}}^{(p),\text{III}} = \frac{TR_{\text{P}} - TR_{\text{B}}}{TR_{P}} M_{\text{forw}}^{(p),\text{III}} + \frac{TR_{\text{B}}}{TR_{\text{P}}} M_{\text{back}}^{(p),\text{III}}$	*B direct mode III*
PBBidir_Forw	$M_{\text{forw}}^{(p),\text{III}}$	
PBBidir_Back	$M_{\text{back}}^{(p),\text{III}}$	

If more than two B frames are inserted between any two consecutive P(I) frames, more B-frame direct modes may be designed, since the bidirectional motion vectors of all preceding B frames can be used to interpolate/extrapolate the bidirectional motion vectors for the succeeding B frame(s) to encode.

For an arbitrary B frame, all the nine direct coding modes (six modes for the B frame following a P or I frame) can be used along with the intra mode. If none of any other mode is exploited, the motion estimation process will not be needed to encode any B frame, and a low complexity video encoder is thus obtained. We would like to point out that using our proposed low complexity video encoding approach, the total number of frames for which the motion estimation process is performed, considering both the encoder and the decoder, is identical to that using a conventional high complexity video encoder. For example, if two B frames are inserted between two consecutive P(I) frames, only one out of three frames on average requires motion estimation to be performed at the encoder. At the decoder, for every group of frames BBP, one motion estimation process needs to be performed to obtain the backward motion vectors pointing from P_{prev} to P_{next}, and another motion estimation process is needed for the bidirectional motion search for the first B frame. Nevertheless, our proposed approach shifts two-third of the motion estimation computation from the encoder to the decoder, hence resulting in a low complexity video encoder.

5.3 Evaluation of Low Complexity Video Encoding Using B-Frame Direct Modes

5.3.1 Implementation Using H.26L/H.264

We choose the ITU-T H.26L version TML9.4 [54] to implement our proposed low complexity video encoding approach. Compared to the previous video coding standards, such as MPEG-4 and H.263+, H.26L, or its follow-up standardization H.264/AVC, includes more features in its video coding layer (VCL) which further improves the coding efficiency at all data rates. H.26L also includes the network abstraction layer (NAL) to improve error resilience performance of the bitstream.

The VCL of H.26L uses seven block shapes for motion estimation and compensation, obtains motion vectors of 1/4 or 1/8 pel resolution, and utilizes multiple reference or multihypothesis for motion compensated prediction. H.26L also exploits a 4×4 integer transform with DCT-like properties to eliminate the rounding mismatch problem in the inverse transform. In addition to the conventional universal variable length coding (UVLC), two new entropy coding modes are designed, the context-based adaptive binary arithmetic coding (CABAC) and the context adaptive variable length coding (CAVLC). In the verification model of H.26L, Lagrangian rate-distortion optimization is recommended in both motion vector selection and macroblock coding mode decision, and closed-form relations between the Lagrangian parameter and the chosen quantization parameter are formulated.

In our experiments, we turned on all seven block-shape options and used the full search range of 16 for motion estimation. We adopted one slice for each frame and chose the UVLC mode for entropy coding. We intra-coded the first frame and inter-coded all successive frames as either P or B frames. We chose the 1/4 pel resolution for the motion vector data and used one reference frame for encoding the P frames. We inserted two B frames within two consecutive P frames such that the GOP structure is in PBBPBB pattern. Since no rate control was implemented in the H.26L reference software, we adjusted the quantization parameter and the frame rate to obtain various decoded video qualities.

Our proposed approach where no motion estimation is implemented for any B frame can be coupled with other low complexity video encoding approaches to achieve low complexity video encoding of the entire video sequence. For example, we may use the NDME approach [167] to encode all the P frames. In our implementation we simply kept the encoding modes of the I and P frames in H.26L/H.264 and specifically focused on the examination and evaluation of the rate distortion performance of the B frames.

We modified the H.26L TML9.4 codec such that only intra or direct coding mode is allowed for encoding a macroblock in a B frame. In particular, we provided six direct coding modes for encoding macroblocks in the first B frame that follows a P (or I) frame while nine direct coding modes for macroblocks in the second B frame,

as given in Table 5.2. We chose the best coding mode for each macroblock, denoted by Mode_{opt}, using the Lagrangian rate distortion optimization as follows

$$\begin{aligned} \text{Mode}_{\text{opt}} &= \text{argmin}_{\text{Mode}}\{J(\text{Mode}|\text{QP}, \lambda_B)\} \\ &= \text{argmin}_{\text{Mode}}\{\text{SSD}(\text{Mode}|\text{QP}) + \lambda_B R(\text{Mode}|\text{QP})\}, \end{aligned} \tag{5.18}$$

where J denotes the Lagrangian function, SSD denotes the sum of the squared differences between the original macroblock and its reconstruction, and R represents the encoding data rate. The Lagrangian parameter, λ_B, is related to the quantization parameter, QP, as follows [169]

$$\lambda_B = 20\frac{\text{QP}+5}{34-\text{QP}}\exp\left(\frac{\text{QP}}{10}\right). \tag{5.19}$$

Above rate distortion optimized mode decision is more computationally expensive than the low complexity decision rules recommended in [170]. The reason we implemented the relatively high complexity mode decision technique is because we would like to examine how good the rate distortion performance of the direct coding modes we designed can achieve as well as the relative effectiveness of each direct coding mode in terms of the coding efficiency.

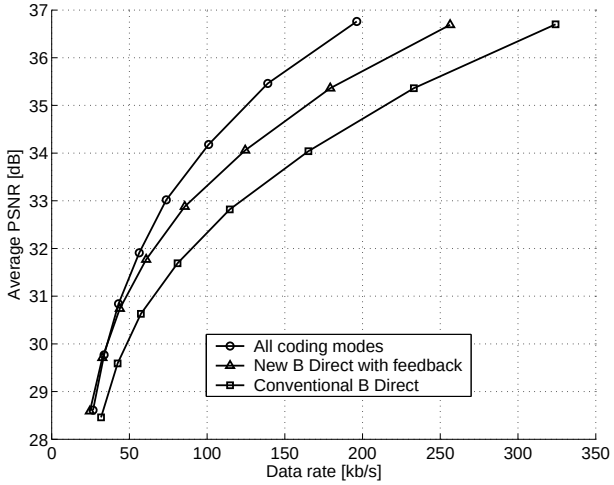
5.3.2 Experimental Results

We chose various video sequences that contain varying degrees of motions in our experiments. We used three QCIF video sequences, with frame size 176×144: *foreman* (400 frames), *coastguard* (300 frames), and *mother-daughter* (as *mthrdghtr* in short in the figure) (400 frames), three CIF video sequences, with frame size

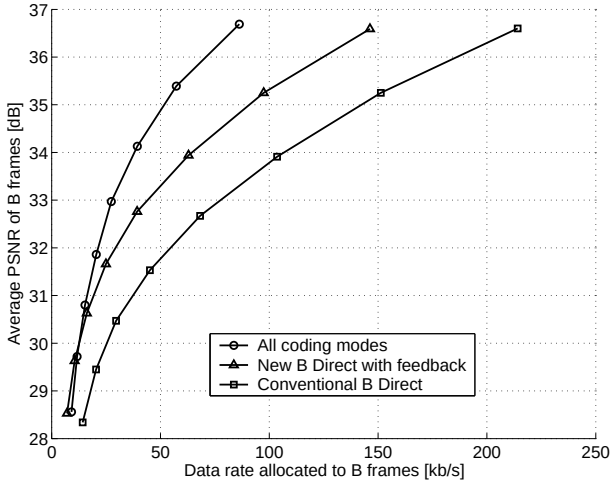
352×288: *foreman* (300 frames), *akiyo* (300 frames), and *bus* (150 frames), and one CCIR601 video sequence, with frame size 720×480: *flowergarden* (150 frames), all in YUV 4:2:0 format.

We used the peak signal-to-noise ratio (PSNR) to measure the distortion. The rate distortion performance of our proposed approach, denoted as "New B Direct with feedback" in the figure, are shown in Fig. 5.27 through 5.36. As a comparison, the rate distortion performance of two video encoding approaches are also given: one approach that encodes the B frames with all the modes specified in the H.26L/H.264 standard, denoted as "All coding modes" in the figure, and another approach where only the intra mode and the conventional B-frame direct coding mode, i.e., PPForw_Bidir, are used, denoted as "Conventional B Direct" in the figure. The approach with all the coding modes turned on for B frames is a high complexity video encoding approach, whereas the other two where only intra or direct coding mode is allowed for B frames are low complexity video encoding approaches.

It can be seen from the results that our low complexity video encoding approach using B-frame direct modes obtains a competitive rate distortion performance compared to conventional high complexity video encoding. Considering varying video characteristics, frame rates, and quantization parameters, our approach on average provides a rate distortion performance that locates somewhere between that of the high complexity video encoding approach and the approach using the conventional direct mode.

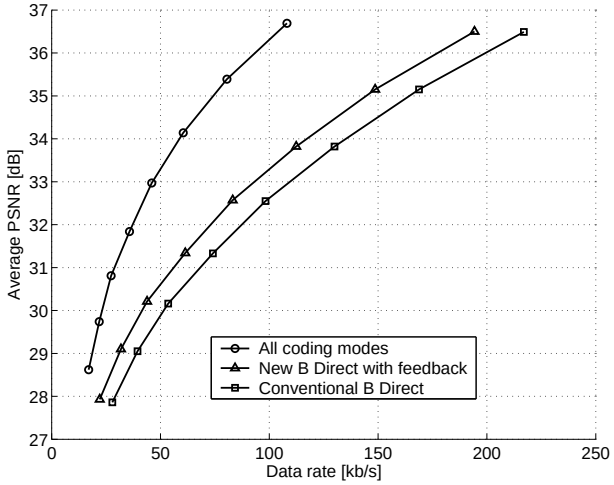


(a) For all frames

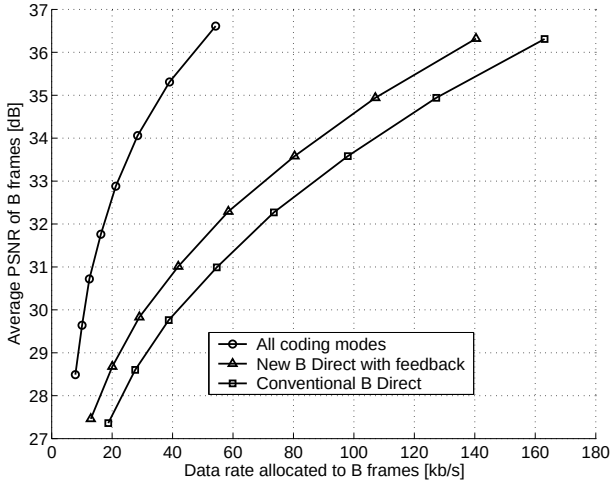


(b) For B frames

Fig. 5.27. Rate distortion performance of B-frame direct modes for QCIF *foreman* (400 frames coded in IBBPBB using modified H.26L at frame rate 30 fps)

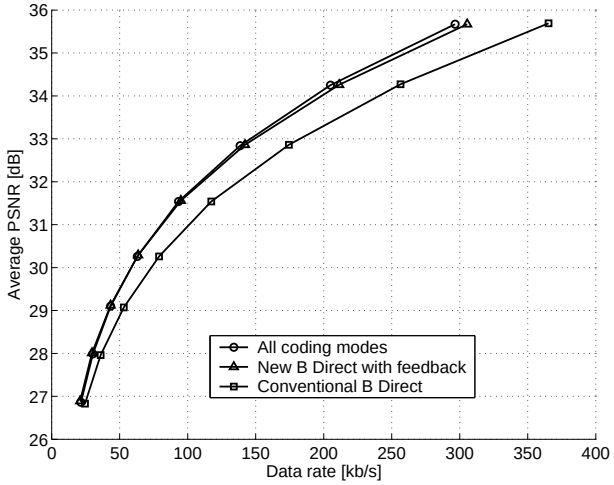


(a) For all frames

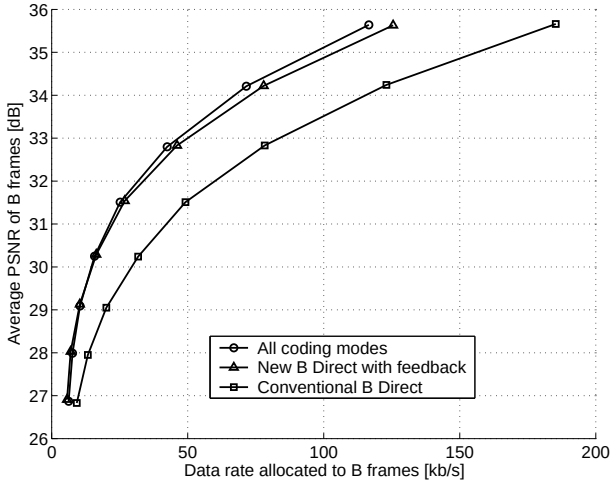


(b) For B frames

Fig. 5.28. Rate distortion performance of B-frame direct modes for QCIF *foreman* (400 frames coded in IBBPBB using modified H.26L at frame rate 10 fps)

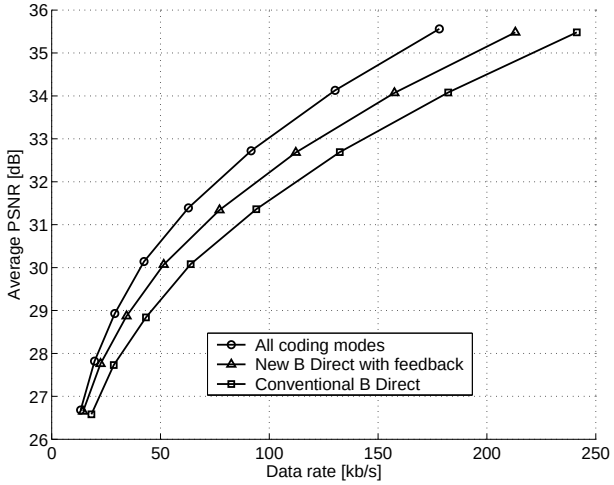


(a) For all frames

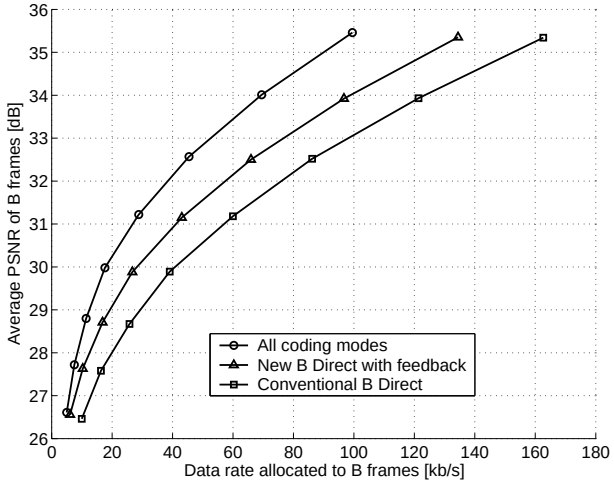


(b) For B frames

Fig. 5.29. Rate distortion performance of B-frame direct modes for QCIF *coastguard* (300 frames coded in IBBPBB using modified H.26L at frame rate 30 fps)

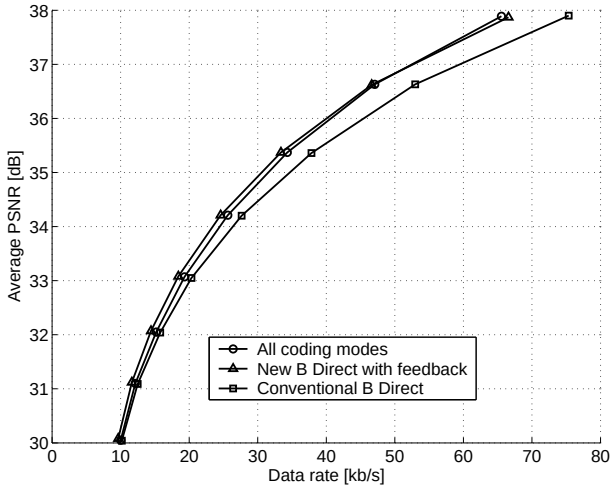


(a) For all frames

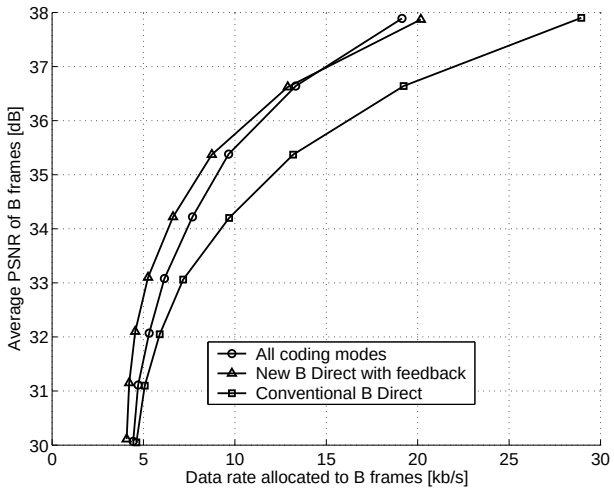


(b) For B frames

Fig. 5.30. Rate distortion performance of B-frame direct modes for QCIF *coastguard* (300 frames coded in IBBPBB using modified H.26L at frame rate 10 fps)

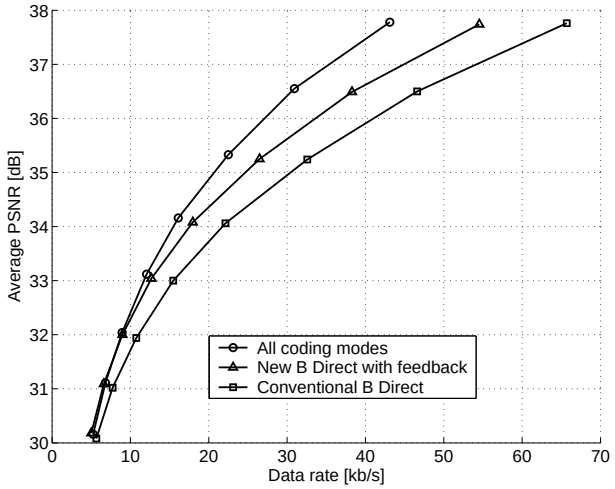


(a) For all frames

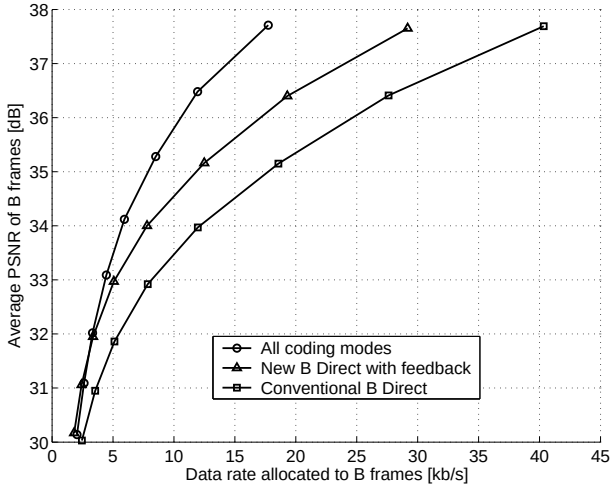


(b) For B frames

Fig. 5.31. Rate distortion performance of B-frame direct modes for QCIF *mthrdghtr* (400 frames coded in IBBPBB using modified H.26L at frame rate 30 fps)



(a) For all frames



(b) For B frames

Fig. 5.32. Rate distortion performance of B-frame direct modes for QCIF *mthrdghtr* (400 frames coded in IBBPBB using modified H.26L at frame rate 10 fps)

It is observed that our low complexity video encoding approach obtains a good performance particularly for video sequences that possess a large amount of global motions. For video sequences such as *coastguard* which contain a high-level amount of motions, our approach performs almost as good as the high complexity video encoding approach when the frame rate is relatively high. As shown in Fig. 5.29, for QCIF *coastguard* coded at 30 fps, to obtain the same quality such as 33 dB in average PSNR, our approach requires less than 2 kbps extra data rate compared to the high complexity video encoding approach but saves more than 30 kbps compared to the conventional direct coding approach.

Our approach performs well for video sequences that contain a relatively low-level amount of motions, such as *mother-daughter* and *akiyo*. For the video sequence *mother-daughter* in QCIF format coded at 30 fps, our approach outperforms the high complexity video encoding approach over a large range of data rate, as shown in Fig. 5.31. When the video sequence is encoded at a quality of 36 dB in average PSNR, our approach performs a little better than the high complexity video encoding approach, and the conventional direct coding approach results in a 5 kbps loss in overall data rate compared to the other two approaches.

When the frame rate decreases, such as the results we obtained for the QCIF video sequences where the frame rate is reduced from 30 fps to 10 fps, our approach obtains a less competitive rate distortion performance. This is not surprising since a direct coding mode uses the interpolated/extrapolated motion vectors for current B frame to be encoded from neighboring frames. When the frame rate is reduced, the

distance between consecutive coded frames is enlarged. The motion vectors obtained by interpolation/extrapolation using direct modes are hence less accurate.

The relative occurrences in percentage of the nine direct coding modes as well as the corresponding three B-frame direct modes are shown in Fig. 5.37 through Fig. 5.56. It is seen that for low-motion video sequences such as *mother-daughter* and *akiyo*, the conventional direct coding mode, PPForw_Bidir, usually dominates all the other direct coding modes. In contrast, when encoding high-motion videos such as *foreman*, *coastguard* and *flowergarden*, the dominant direct coding mode varies with the quantization parameter as well as the frame rate. For instance, instead of the conventional direct coding mode, the direct coding mode PPForw_Back, which uses the motion vectors derived by interpolation from the forward motion vectors pointing from P frame to P frame and the backward motion prediction, usually dominates all the other direct coding modes when the quantization parameter is relatively large. For the second B frame between consecutive P frames, the mode PBBidir_Bidir, which uses the motion vectors interpolated/extrapolated from the bidirectional motion vectors of the previous B frame, as implied by the first "Bidir," and the bidirectional motion compensated prediction, as implied by the second "Bidir," is usually the majority direct coding mode when the quantization parameter is relatively small.

When video sequences that contain a large amount of local motions such as *foreman*, direct modes cannot provide sufficiently accurate motion vectors, especially when the frame rate is relatively low. It is observed that in this case, the intra-coding

mode will be more frequently chosen, as depicted by the results we obtained for QCIF *foreman* that was coded at frame rate 10 fps.

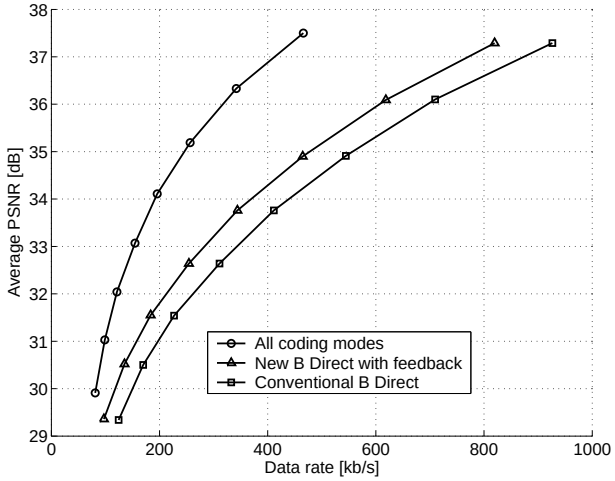
We would like to point out that one reason for *B direct mode I* to have obtained a relatively better rate distortion performance, compared to the other two B-frame direct modes, is that the motion vectors it uses for motion vector interpolation/extrapolation are estimated at the encoder. Hence, the original frame is used for the co-located macroblock, in contrast to the use of the reconstructed frame in the other two B-frame direct modes.

5.4 Conclusions

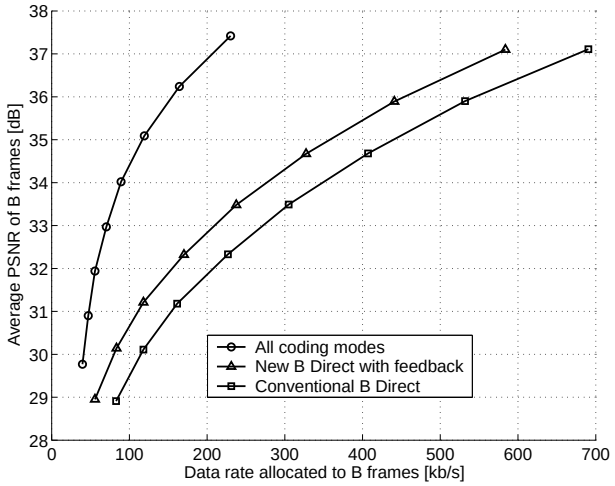
Low complexity video encoding shifts the computational complexity from the encoder to the decoder, meeting the requirement from new emerging applications such as wireless sensor networks and mobile video cameras for video surveillance. Low complexity video encoding is radically different from conventional high complexity video encoding. Studies of practical low complexity video encoding design have not been fully explored yet.

In this chapter, we contribute the following work for low complexity video encoding:

- We have proposed a low complexity video encoding approach using B-frame direct modes. The direct mode was originally developed for encoding B frames. The motion compensated prediction of the direct mode is a linear combina-

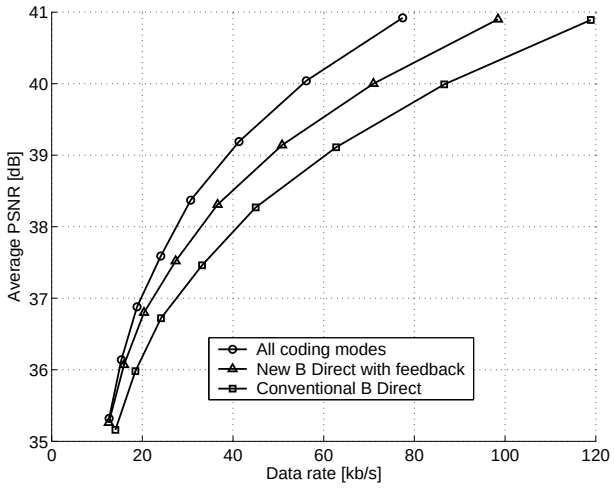


(a) For all frames

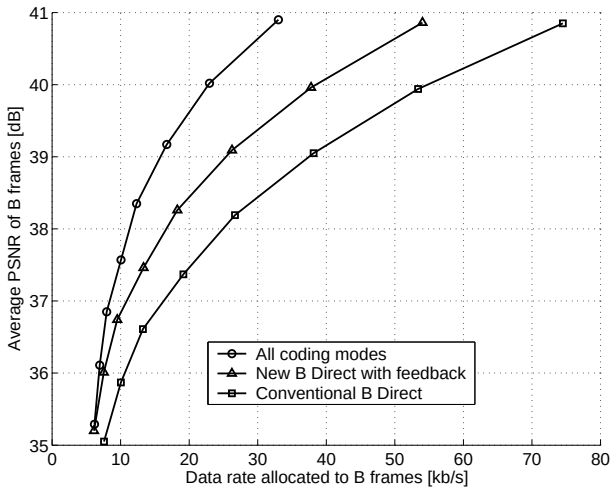


(b) For B frames

Fig. 5.33. Rate distortion performance of B-frame direct modes for CIF *foreman* (300 frames coded in IBBPBB using modified H.26L at frame rate 15 fps)

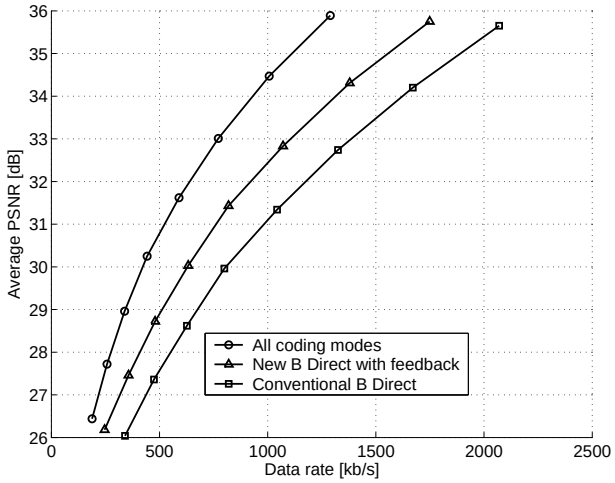


(a) For all frames

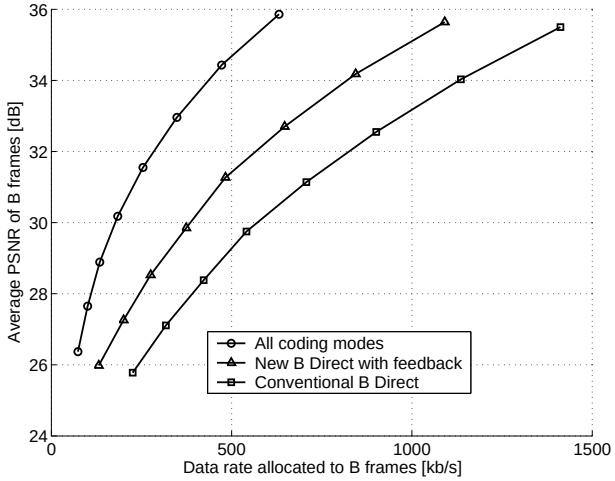


(b) For B frames

Fig. 5.34. Rate distortion performance of B-frame direct modes for CIF *akiyo* (300 frames coded in IBBPBB using modified H.26L at frame rate 15 fps)

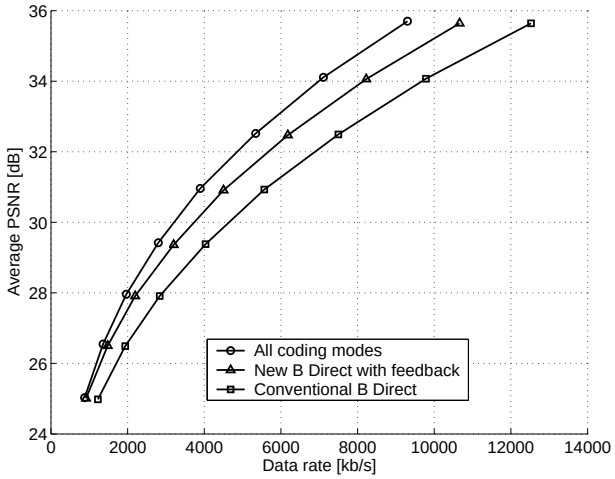


(a) For all frames

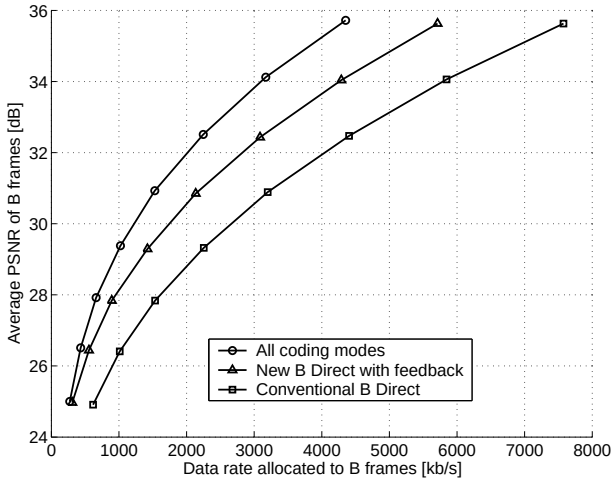


(b) For B frames

Fig. 5.35. Rate distortion performance of B-frame direct modes for CIF *bus* (150 frames coded in IBBPBB using modified H.26L at frame rate 15 fps)

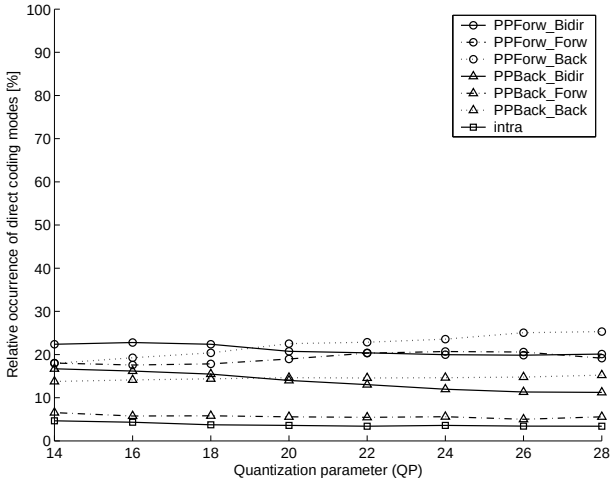


(a) For all frames

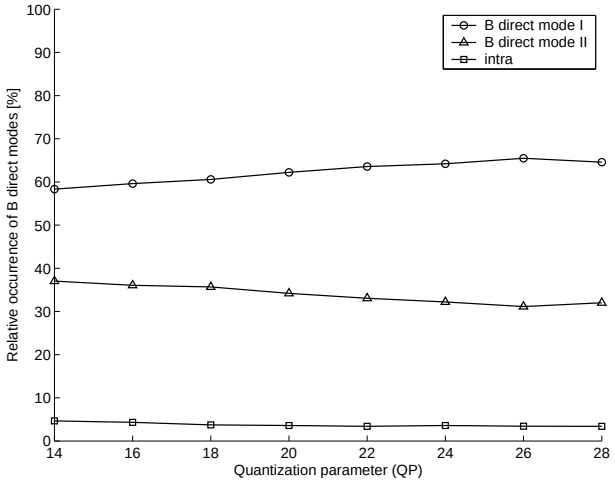


(b) For B frames

Fig. 5.36. Rate distortion performance of B-frame direct modes for CCIR601 *flowergarden* (150 frames coded in IBBPBB using modified H.26L at frame rate 30 fps)

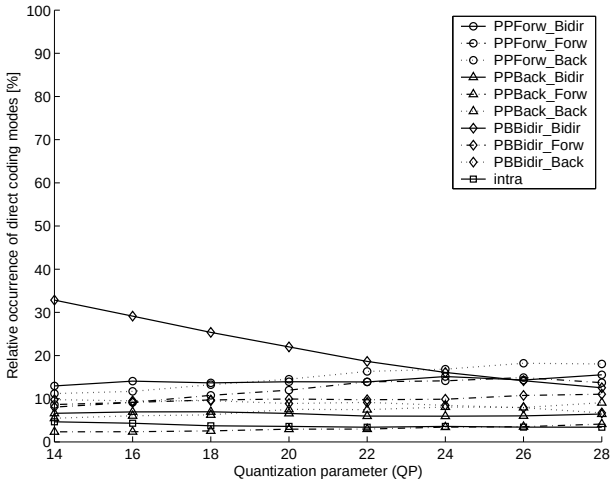


(a) Direct coding modes

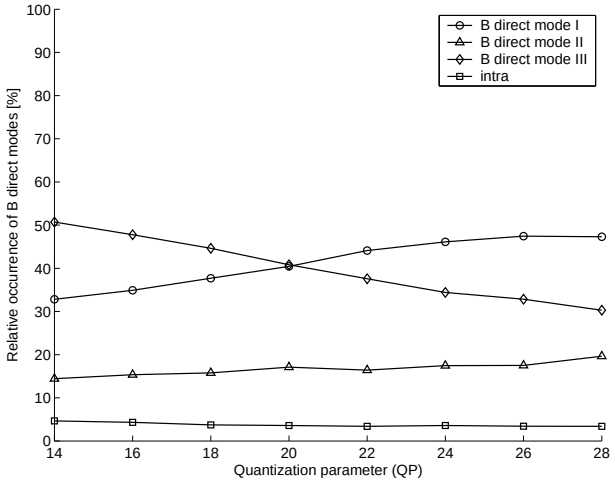


(b) B direct modes

Fig. 5.37. Relative occurrence of B-frame direct modes for the first B frame of QCIF *foreman* (400 frames coded in IBBPBB using modified H.26L at frame rate 30 fps)

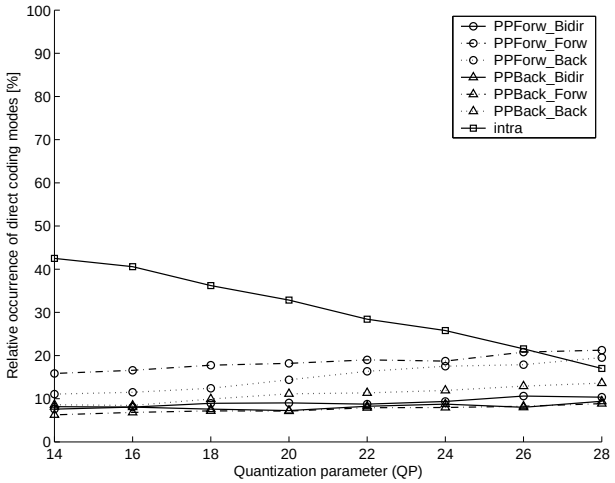


(a) Direct coding modes

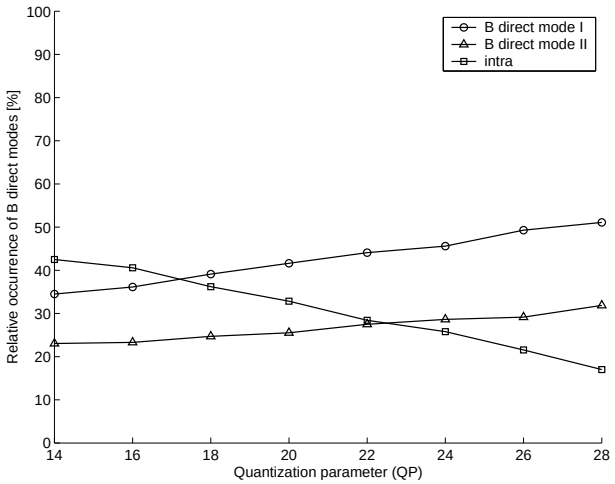


(b) B direct modes

Fig. 5.38. Relative occurrence of B-frame direct modes for the second B frame of QCIF *foreman* (400 frames coded in IBBPBB using modified H.26L at frame rate 30 fps)

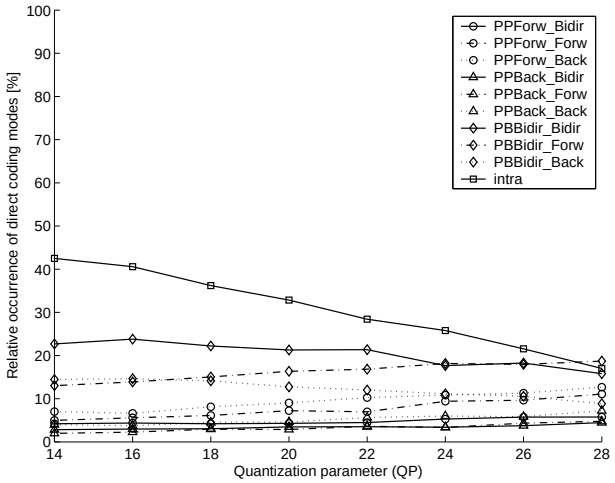


(a) Direct coding modes

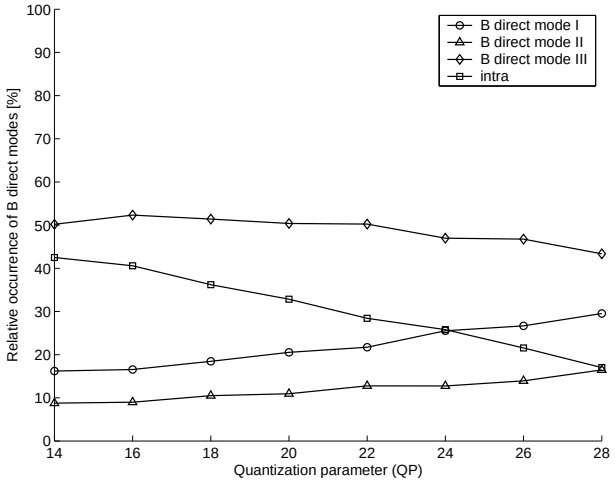


(b) B direct modes

Fig. 5.39. Relative occurrence of B-frame direct modes for the first B frame of QCIF *foreman* (400 frames coded in IBBPBB using modified H.26L at frame rate 10 fps)



(a) Direct coding modes



(b) B direct modes

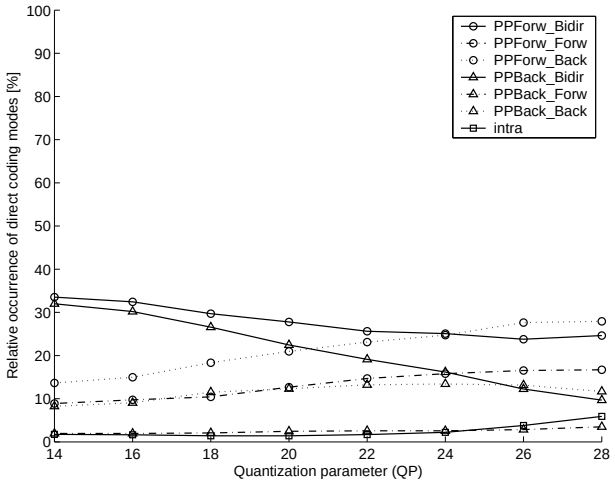
Fig. 5.40. Relative occurrence of B-frame direct modes for the second B frame of QCIF *foreman* (400 frames coded in IBBPBB using modified H.26L at frame rate 10 fps)

tion of the two predictions obtained from the forward motion compensation and the backward motion compensation. In our proposed approach, we have extended the direct-mode idea and designed new B-frame direct modes to encode B frames. We constrained any macroblock in a B frame to be encoded only using either the intra-coding mode or one of the new direct coding modes. Hence, no motion estimation is used to any B frames and low complexity video encoding is achieved. Our proposed low complexity video encoding requires a feedback channel from the decoder to the encoder. We implemented motion estimation at the decoder and transmitted the motion vectors back to the decoder. We have designed three B-frame direct modes and specified nine coding modes for B frame macroblocks: PPForw_Bidir, PPForw_Forw, PPForw_Back, PPBack_Bidir, PPBack_Forw, PPBack_Back, PBBidir_Bidir, PBBidir_Forw, and PBBidir_Back. Experimental results have shown that our approach using new B-frame direct modes with help of a feedback channel obtains a competitive rate distortion performance compared to that of the high complexity video encoding approach.

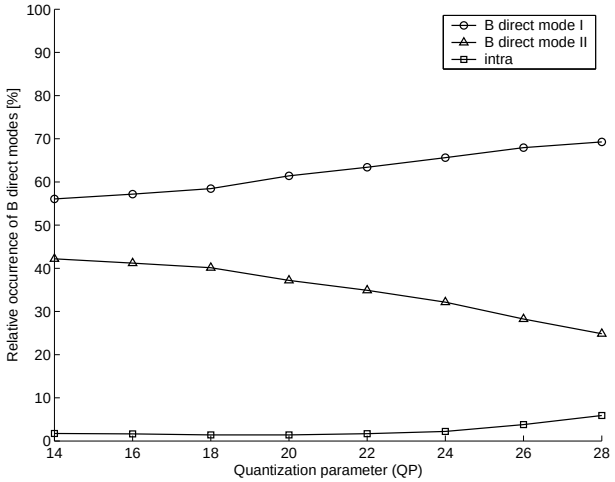
We would like to point out that the use of our new B-frame direct modes may also be beneficial in improving the performance of conventional high complexity video encoding. For example, it has been observed from Subsection 5.3.2 that the direct coding mode PPForw_Back, which simply uses the backward motion prediction, is sometimes more effective than the conventional direct mode PPForw_Bidir that uses the linearly combined prediction. If high complexity is allowed in both the encoder

and decoder, we may also use the third B-frame direct mode, *B direct mode III*, to encode succeeding B frame(s) using the bidirectional motion vectors of preceding B frame(s). If *B direct mode III* is used, the bidirectional motion vectors of the previous B frame do not have to be coded, since the decoder may obtain the same motion vectors by simply repeating the same motion estimation process. With more direct modes provided for encoding the B frames using the conventional coding paradigm, we may expect a further improvement in the performance of conventional video coding algorithms.

It is noted that if a macroblock is coded in one of the B-frame direct coding modes, its motion vectors are derived by interpolation or extrapolation from the motion vectors associated to the co-located macroblock. Hence, when video bit-streams suffer from channel errors, errors that occur to the motion vectors of the co-located macroblock, either in the forward channel or in the feedback channel, will inevitably propagate to the macroblock that is coded in direct mode. In our future work, we will focus on improving error resilience performance of our low complexity video encoding approach using B-frame direct modes.

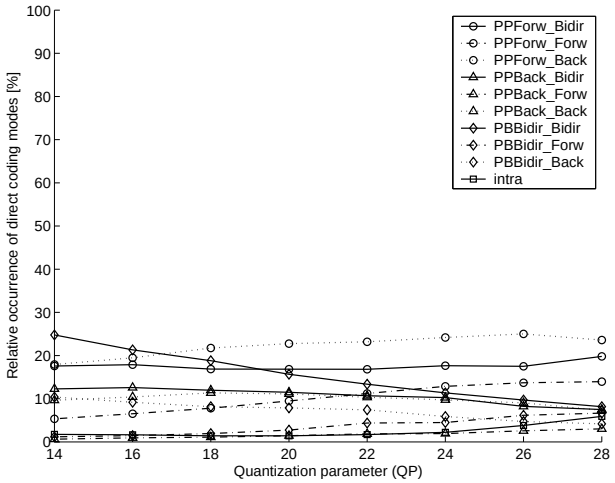


(a) Direct coding modes

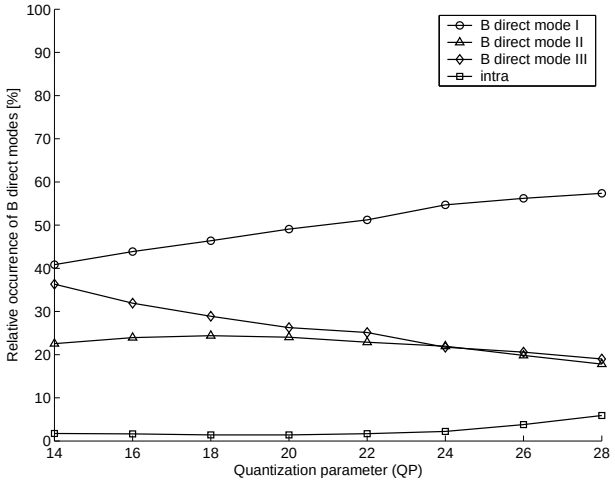


(b) B direct modes

Fig. 5.41. Relative occurrence of B-frame direct modes for the first B frame of QCIF *coastguard* (400 frames coded in IBBPBB using modified H.26L at frame rate 30 fps)

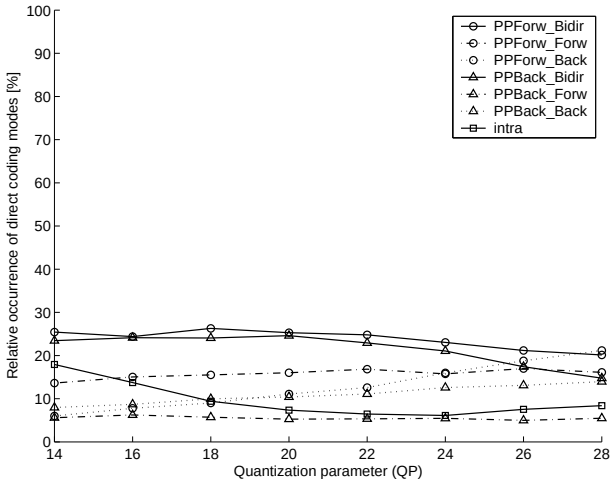


(a) Direct coding modes

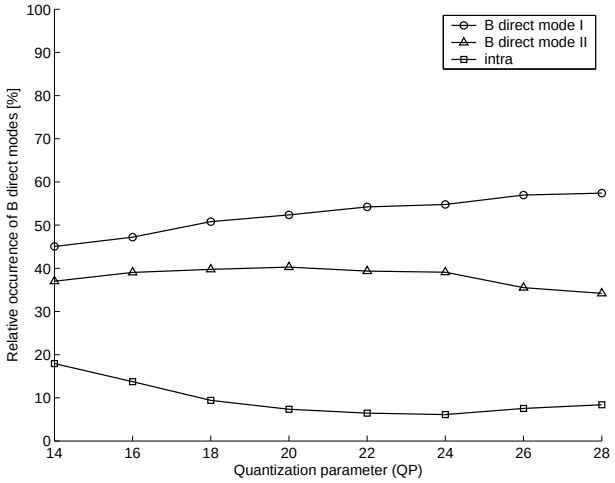


(b) B direct modes

Fig. 5.42. Relative occurrence of B-frame direct modes for the second B frame of QCIF *coastguard* (400 frames coded in IBBPBB using modified H.26L at frame rate 30 fps)

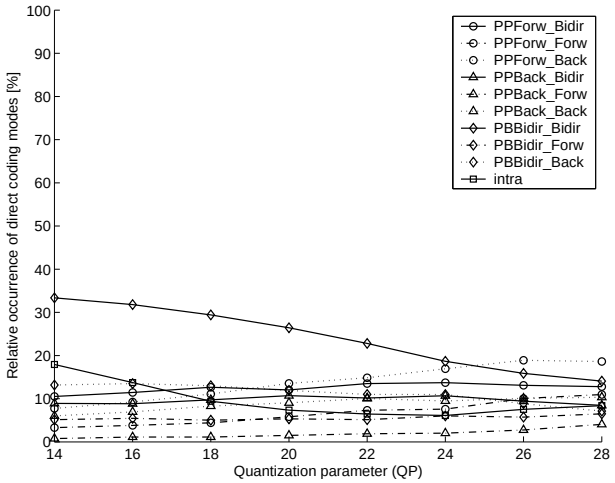


(a) Direct coding modes

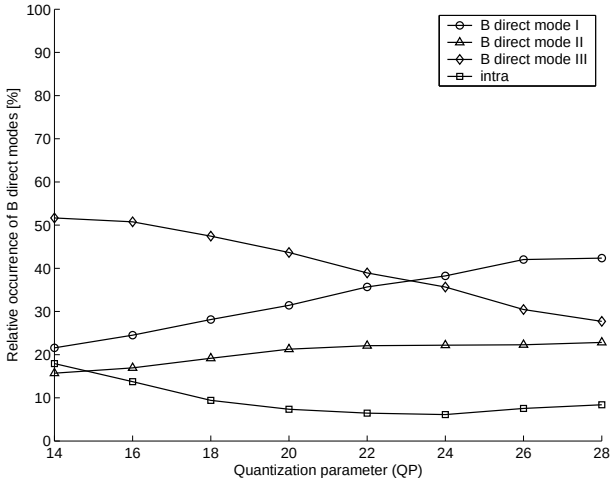


(b) B direct modes

Fig. 5.43. Relative occurrence of B-frame direct modes for the first B frame of QCIF *coastguard* (400 frames coded in IBBPBB using modified H.26L at frame rate 10 fps)

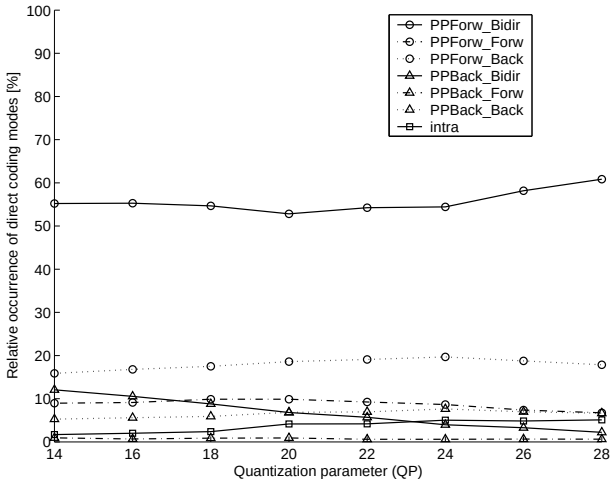


(a) Direct coding modes

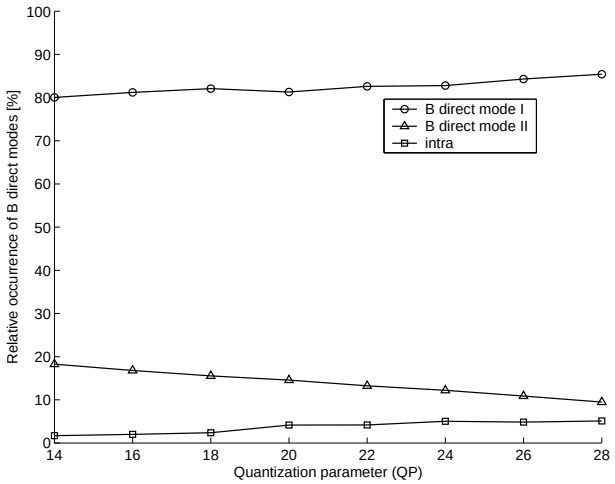


(b) B direct modes

Fig. 5.44. Relative occurrence of B-frame direct modes for the second B frame of QCIF *coastguard* (400 frames coded in IBBPBB using modified H.26L at frame rate 10 fps)

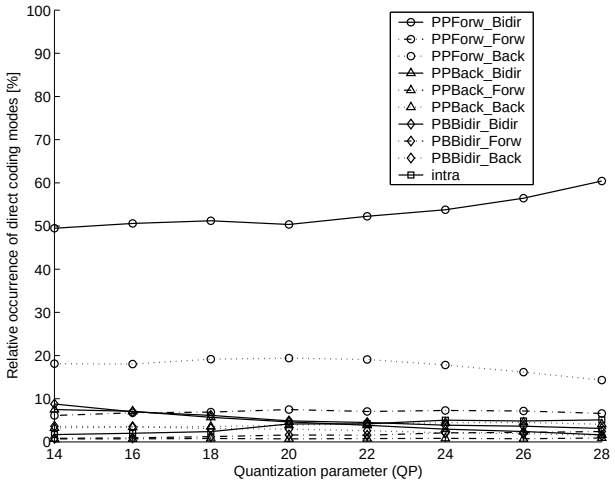


(a) Direct coding modes

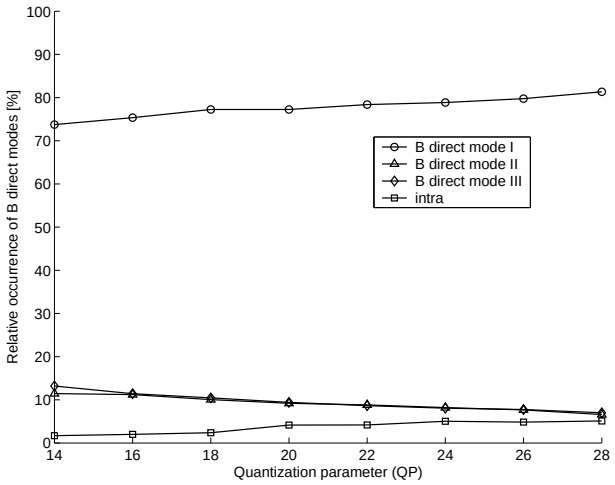


(b) B direct modes

Fig. 5.45. Relative occurrence of B-frame direct modes for the first B frame of QCIF *mthrdghtr* (400 frames coded in IBBPBB using modified H.26L at frame rate 30 fps)

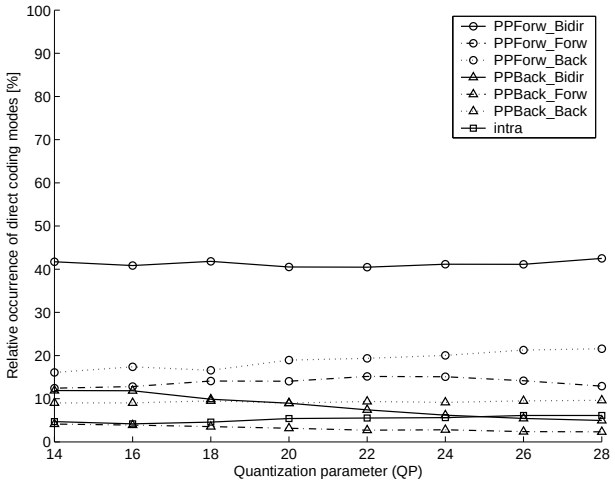


(a) Direct coding modes

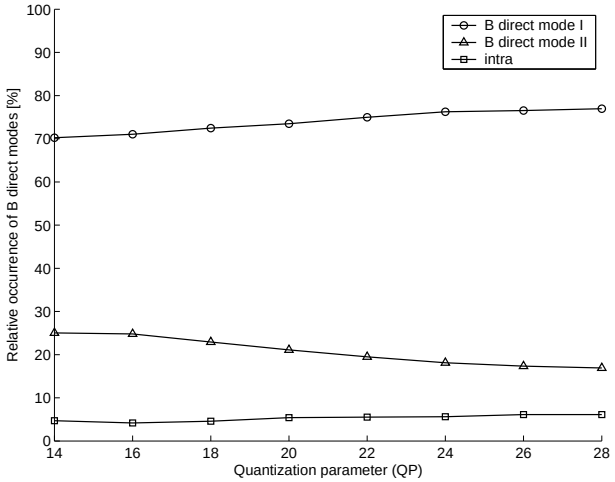


(b) B direct modes

Fig. 5.46. Relative occurrence of B-frame direct modes for the second B frame of QCIF *mthrdghtr* (400 frames coded in IBBPBB using modified H.26L at frame rate 30 fps)

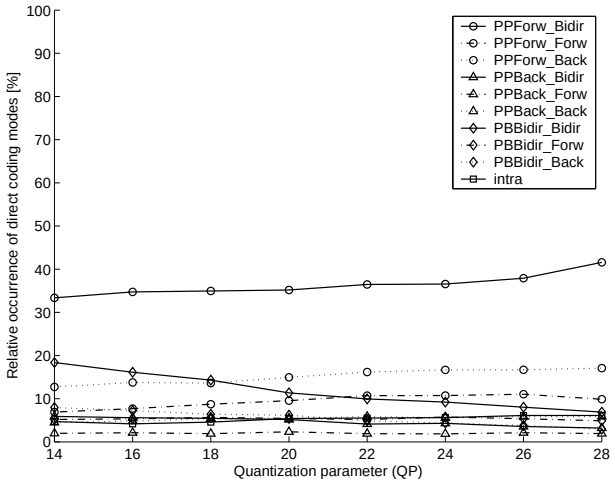


(a) Direct coding modes

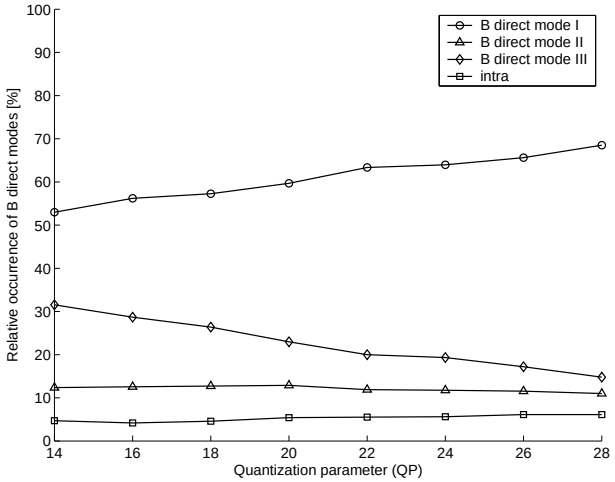


(b) B direct modes

Fig. 5.47. Relative occurrence of B-frame direct modes for the first B frame of QCIF *mthrdghtr* (400 frames coded in IBBPBB using modified H.26L at frame rate 10 fps)

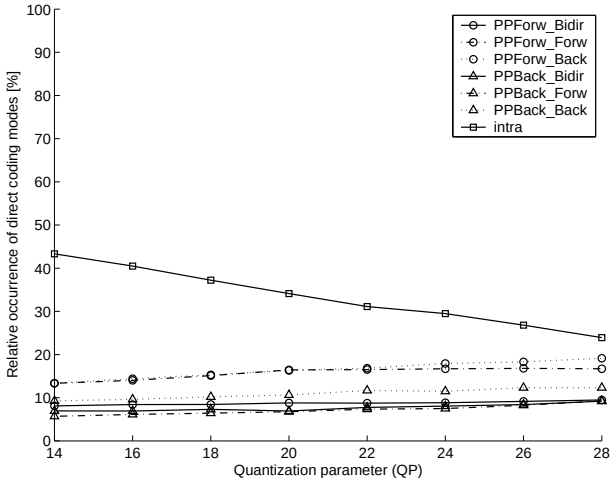


(a) Direct coding modes

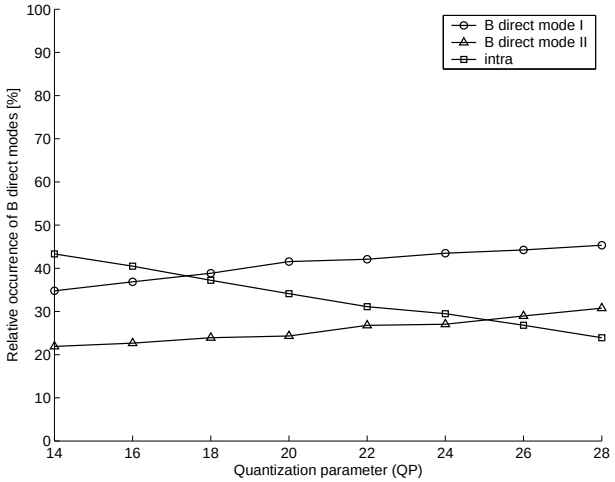


(b) B direct modes

Fig. 5.48. Relative occurrence of B-frame direct modes for the second B frame of QCIF *mthrdghtr* (400 frames coded in IBBPBB using modified H.26L at frame rate 10 fps)

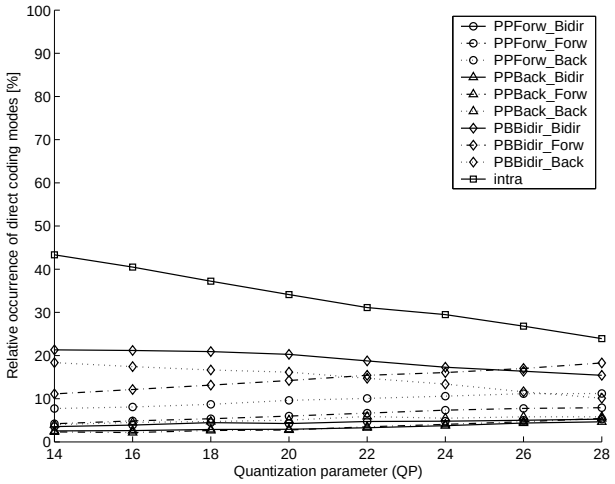


(a) Direct coding modes

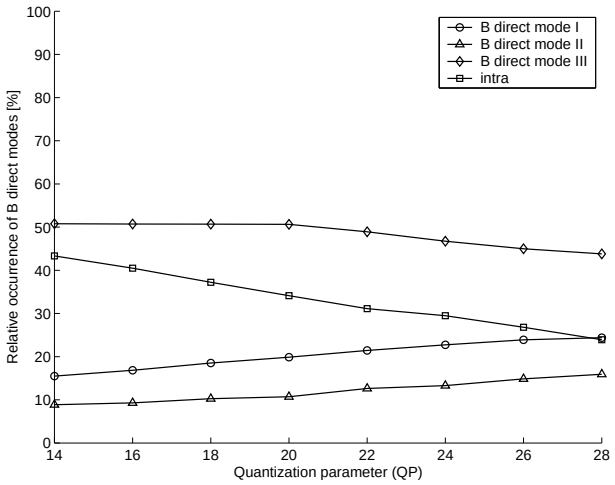


(b) B direct modes

Fig. 5.49. Relative occurrence of B-frame direct modes for the first B frame of CIF *foreman* (300 frames coded in IBBPBB using modified H.26L at frame rate 15 fps)

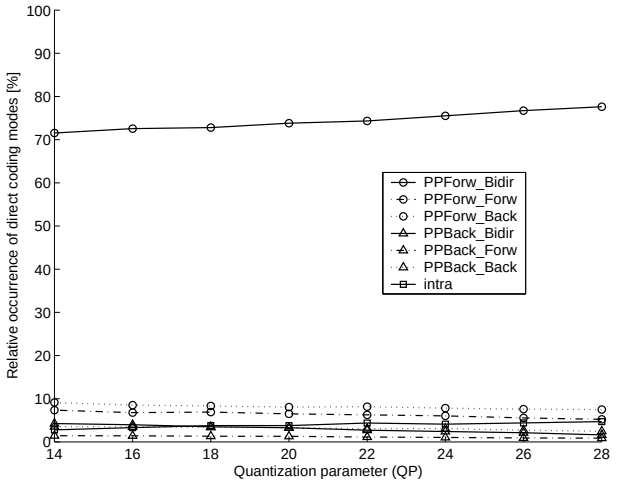


(a) Direct coding modes

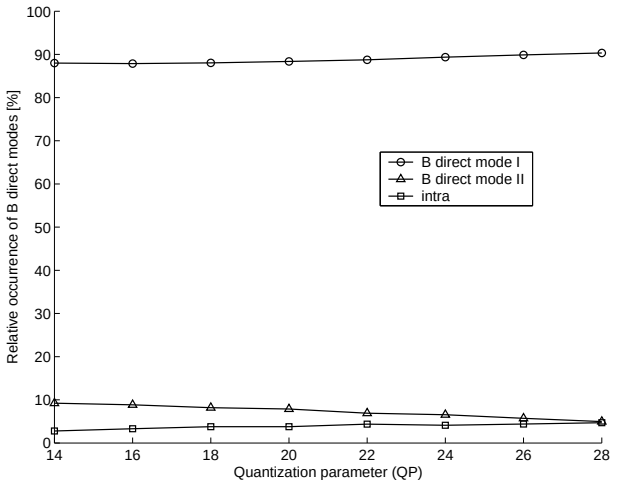


(b) B direct modes

Fig. 5.50. Relative occurrence of B-frame direct modes for the second B frame of CIF *foreman* (300 frames coded in IBBPBB using modified H.26L at frame rate 15 fps)

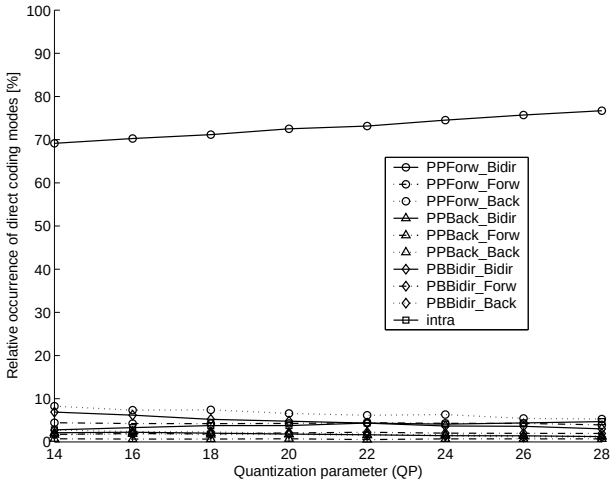


(a) Direct coding modes

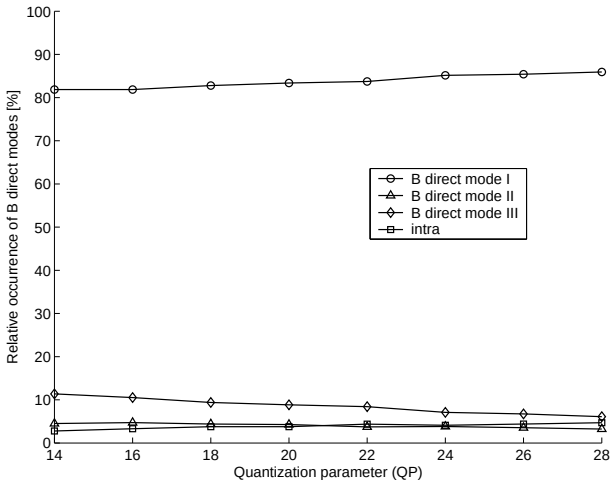


(b) B direct modes

Fig. 5.51. Relative occurrence of B-frame direct modes for the first B frame of CIF *akiyo* (300 frames coded in IBBPBB using modified H.26L at frame rate 15 fps)

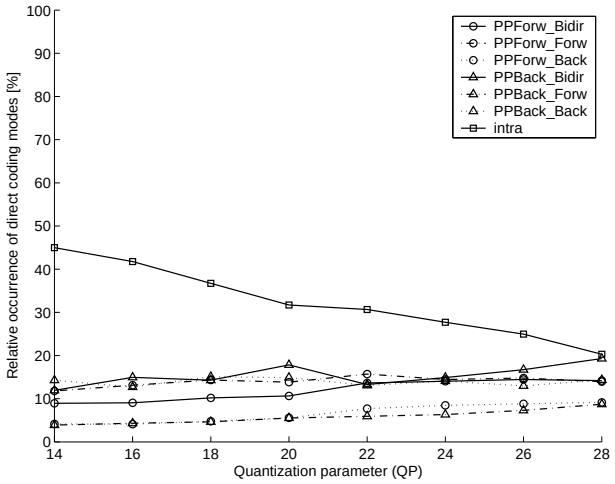


(a) Direct coding modes

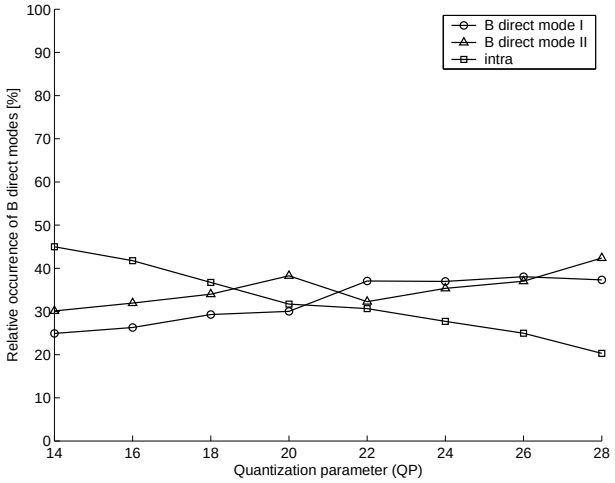


(b) B direct modes

Fig. 5.52. Relative occurrence of B-frame direct modes for the second B frame of CIF *akiyo* (300 frames coded in IBBPBB using modified H.26L at frame rate 15 fps)

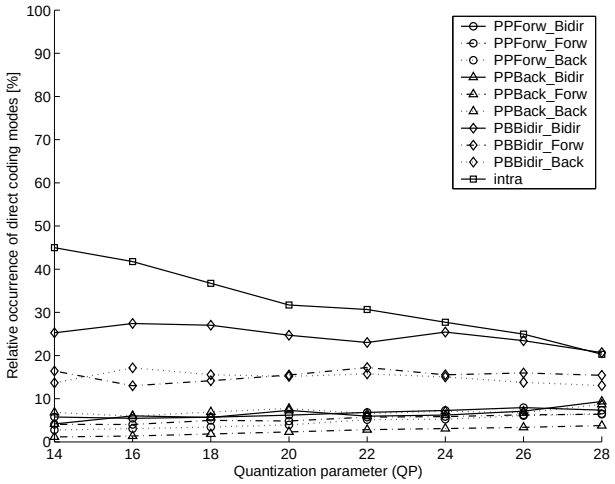


(a) Direct coding modes

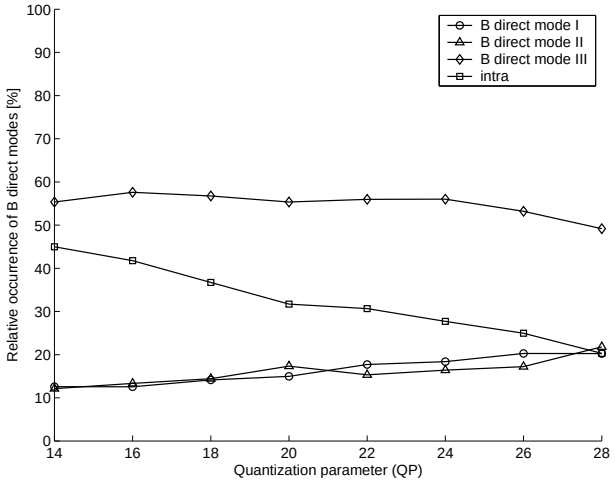


(b) B direct modes

Fig. 5.53. Relative occurrence of B-frame direct modes for the first B frame of CIF *bus* (150 frames coded in IBBPBB using modified H.26L at frame rate 15 fps)

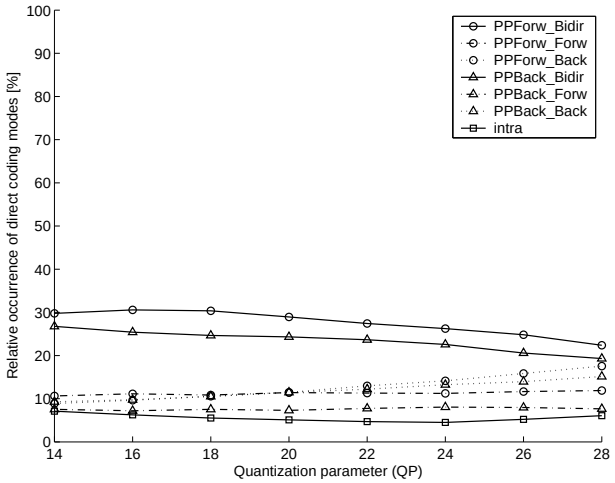


(a) Direct coding modes

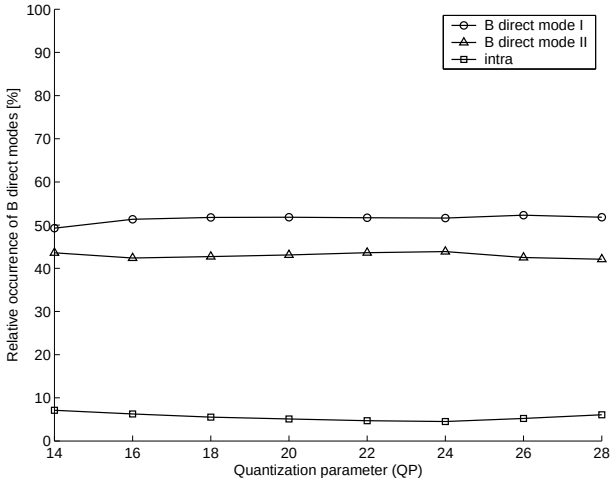


(b) B direct modes

Fig. 5.54. Relative occurrence of B-frame direct modes for the second B frame of CIF *bus* (150 frames coded in IBBPBB using modified H.26L at frame rate 15 fps)

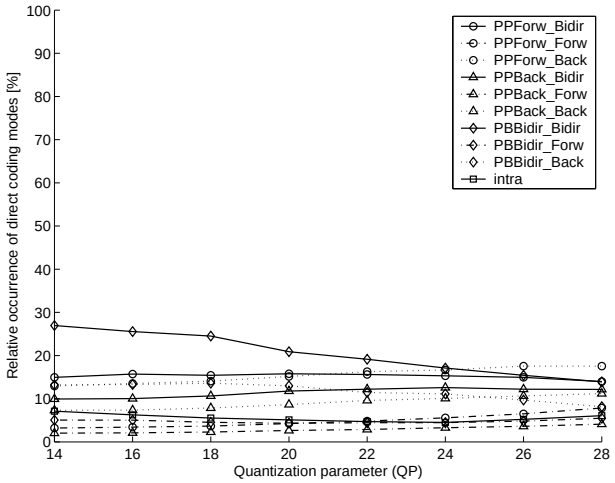


(a) Direct coding modes

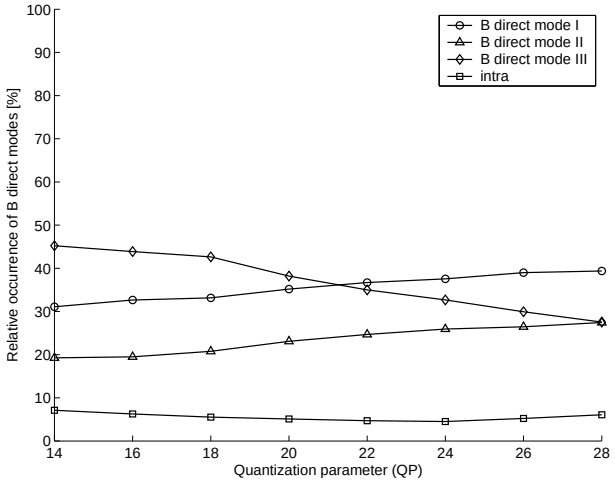


(b) B direct modes

Fig. 5.55. Relative occurrence of B-frame direct modes for the first B frame of CCIR601 *flowergarden* (150 frames coded in IBBPBB using modified H.26L at frame rate 30 fps)



(a) Direct coding modes



(b) B direct modes

Fig. 5.56. Relative occurrence of B-frame direct modes for the second B frame of CCIR601 *flowergarden* (150 frames coded in IBBPBB using modified H.26L at frame rate 30 fps)

6. EVALUATION OF JOINT SOURCE AND CHANNEL CODING OVER WIRELESS NETWORKS

6.1 Introduction

In this chapter, we examine the trade-offs in source and channel coding for video transmission over a wireless channel [171]. In particular, we explore the error resilient features as recommended by the ITU for the H.263+ coder and forward error correction (FEC) using Reed-Solomon codes. As we have discussed in Chapter 1, the advantages of using error resilience include standards compliance and no additional delay. The disadvantages include slight reduction in compression efficiency and by definition, error resilience is designed to reduce error propagation, not detect and correct errors. Channel coding using FEC has the advantage of the ability to detect and correct for errors. Channel coding can be used to design an unequal error protection scheme for high and low priority data and FEC can be added at the application layer for current networks where the physical and link layers cannot be altered and may provide channel conditions which are not acceptable for video applications. The disadvantage of FEC includes additional overhead (bandwidth), additional delay and additional software at the client in order to be able to decode and play the video.

Other work which examined the tradeoffs between source-channel coding for video over wireless includes [172] and [173].

We examine the H.263+ coder for video transmission over a wireless network. The particular annexes that we explore within the standard are annexes related to coding efficiency such as Annex D - unrestricted motion vector mode, Annex F - advanced prediction mode, and Annex I - advanced INTRA coding mode. We also examine annexes related to error resilience (the ability to recover or mitigate error propagation) such as Annex K - slice structured mode, Annex N - reference picture selection mode, and Annex V - data partitioning.

The experiments use the H.263+ standard and software which emulates the UMTS (Universal Mobile Telecommunication System), 3G network provided by [174]. As illustrated in Fig. 6.1, the overall system consists of one computer which acts as the server, both encoding and packetizing the bitstream. The bitstream is sent to the UMTS proxy machine and corrupted according to the traditional Gilbert 2-state model with parameters for the bit error rate (BER) and error burst length. Moreover, the proxy implements rate 1/3 Turbo coding as a realization of channel coding in the physical layer. The third machine acts as the client, decoding and playing the video and a fourth computer is used to initiate and monitor the proxy.

6.1.1 Channel Condition Scenarios

Two major lossy channel scenarios are considered in this dissertation: packet networks and wireless channel. Both IP (Internet Protocol) and ATM (Asyn-

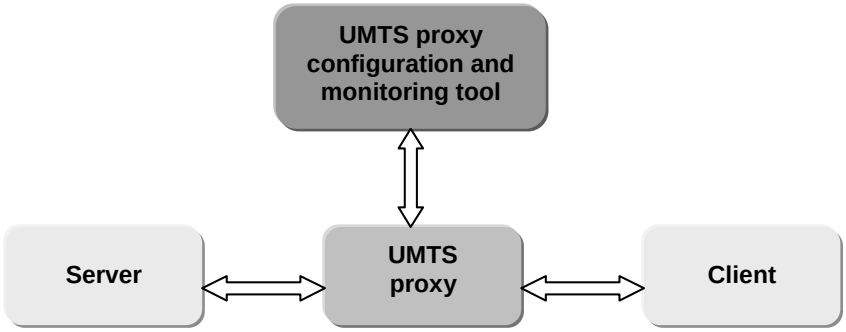


Fig. 6.1. Structure of the emulated UMTS video communication system

chronous Transfer Mode) networks are packet-switched networks, characterized by burst packet loss due to network congestion. The excessively delayed packets beyond the maximum tolerable transmission delay or the buffer size of the decoder are discarded. Wireless channels are, in contrast, characterized by burst bit errors that are mainly caused by channel fading. Although many underlying components contribute to quality degradation for video transmission over lossy channels, we mainly focus on the video signal corruption due to packet loss or bit errors. With the explosion of wireless Internet access applications and the appearance of the next generation wireless networks such as the Universal Mobile Telecommunications System (UMTS) [175, 176], encoded video signals directly interface with packetization stages in the higher layer network protocols of a wireless communication system. Therefore, we are particularly interested in the error resilience performance of video communication applications that are mainly impeded by packet loss. The transmission packet networks are associated with flexible-sized packets and a specified maximum packet size.

Burst error behaviors are mainly characterized by two parameters: the average packet loss or bit error ratio, and the average burst length. The minimum data units in wireless channels are bits, while packets construct the minimum data units in packet networks. Regarding the error behavior of the data units, two channel models are usually used in the literature: (1) Uniform and independent data error assumption; (2) Two-state discrete Markov chain model, i.e., Gilbert model. As shown in Fig. 6.2, the two states of Gilbert model are denoted as G (good) and B

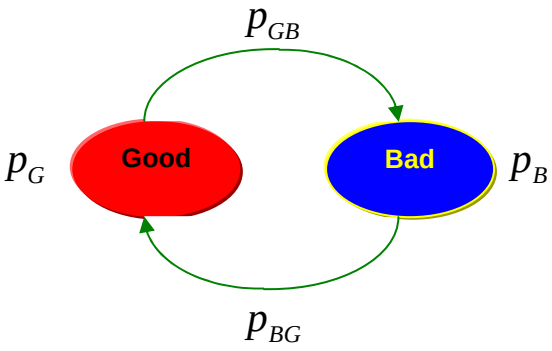


Fig. 6.2. Gilbert model for packet loss or bit errors

(bad), and the model is fully described by the transition probabilities p_{GB} from state G to state B and p_{BG} from state B to state G. In the Gilbert model, the average data error probability p_B and the burst length L_B can be obtained as follows

$$p_B = \frac{p_{\mathrm{GB}}}{p_{\mathrm{GB}} + p_{\mathrm{BG}}}, \tag{6.1}$$

$$L_B = \frac{1}{p_{\mathrm{BG}}}. \tag{6.2}$$

Other channel conditions that might be considered for video transmission applications include the availability of the feedback channel and the unicast or multicast oriented scenarios. In this dissertation, we particularly consider the scenario of video transmission over best-effort networks where no QoS (quality of service) is guaranteed, since this is the situation of current IP networks. Guaranteed network QoS is under consideration for the next generation network protocols, with an example the QoS architecture of the UMTS system, where the bitstream might be transmitted through a number of logical channels with different QoS. Another example is the Differentiated Services (DiffServ), which has been attracting more attention in the literature [177, 178]. In DiffServ, packets are classified and marked to obtain a particular per-hop forwarding behavior on nodes along their path.

6.1.2 Overview of Joint Source and Channel Coding

To further improve the error robustness of the bitstreams for video transmission over an error-prone environment, especially to combat serious channel errors and burst errors, additional error protection needs to be used to the bitstream in addition

to the error resilience elements introduced at the source coding stage. In particular, unequal error protection (UEP) techniques are needed to protect the more critical information in the bitstream. Several UEP techniques have already been addressed in the literature, such as retransmission by the so-called Automatic Retransmission on Request (ARQ) technique, channel coding in the application layer such as using FEC, or distribution through channels with different channel error characteristics [177]. For current wireless and IP networks, no data priority transmission service is provided. Therefore, it is the task of the encoder to realize priority encoding transmission (PET). ARQ requires a backward channel and hence results in a certain delay to transmit the feedback information to the encoder. Especially ARQ is not suitable for multicasting to a large number of receivers since the network traffic may increase dramatically by the use of ARQ. Compared to ARQ, error control coding using FEC in the application layer can better comply with the time constraint required by the time-sensitive video streaming applications.

As we discussed in Subsection 1.1.5 of Chapter 1, the introduction of FEC in the application layer for error protection falls into the category of the joint source and channel coding (JSCC) problem, which aims to optimize bit allocation between source coding and channel coding to balance coding efficiency and error robustness [179]. An overview of JSCC techniques for video transmission over bit-error featured wireless channels is presented in [173]. In particular, the overview focuses on the JSCC methods using the residual redundancy. The residual redundancy refers to the statistical dependency that remains in the output of the source encoder. JSCC using

the residual redundancy does not attempt to remove the redundancy, but rather to make use of this redundancy as a form of implicit channel coding to substitute the conventional channel coding that is explicitly implemented.

In [172], FEC, which is implemented by Reed-Solomon (RS) coding, is used for robust transmission over lossy channels. The lossy channel is characterized by the two-state Markov model to track the burst errors in the symbol level. An (n, k) RS code first groups the bitstream into blocks of k information symbols, and then appends $(n - k)$ parity symbols to each block. Each symbol is composed of m bits, and the maximum block length is $n_{\max} = 2^m - 1$ in units of symbols. By the use of shortened RS codes, any small value can be chosen for n, hence providing a great flexibility in the system design. The symbol errors whose locations can be exactly identified at the decoder are referred to as symbol erasures. An (n, k) RS code can correct up to $\left\lfloor \frac{n-k}{2} \right\rfloor$ symbol errors and $(n-k)$ erasures, where $\lfloor x \rfloor$ denotes the largest integer no greater than x. It has been pointed out that block codes are perfectly suited for error protection against burst errors since they are maximum distance separable codes, and no other codes can reconstruct symbol erasures from a smaller number of correctly-received code symbols [180]. In [172], the channel code rate, k/n, together with the INTRA refresh rate are considered the two most significant system parameters. A framework modeling the entire video communication system, including the video source coder, FEC, interleaving, and error concealment, is developed and optimized through selecting appropriate system parameters under different channel conditions.

Error protection can also be implemented using packet-level FEC schemes, which provide an efficient way to combat packet loss, even though the perfect recovery may be not guaranteed [181]. As shown in Fig. 6.3, block of packets (BOP) are first constructed, each containing k information data packets and $(n-k)$ parity packets. An RS encoder generates $(n-k)$ parity symbols from the k information symbols that are contained in k different information packets and all located in the same position relative to the packet. Each of the $(n-k)$ parity symbols is then placed in the same position in the respective parity packet. The k information symbols and their corresponding $(n-k)$ parity symbols are represented by the column boxes enclosed within the red rectangle in Fig. 6.3. Since block coding requires fixed-size packets in each BOP, each data packet has to be stuffed till the maximum packet size of the BOP. The BOP size in units of packets can vary from one to the other. Notice that for a BOP with fixed size n, the number of data packets k determines the error protection level and thus different transmission priorities can be realized by adjusting this parameter.

A similar work was done in [182] where RS coding across packets is implemented to scalable coded bitstreams.

In [183], an adaptive encoding structuring scheme is proposed. A hypothetical distortion that considers channel errors is first estimated at the encoder using a video packet loss model. Error resilience mechanisms are then adaptively introduced to the encoded bitstream to satisfy the distortion requirement. Error protection using FEC is further considered, and the video packet loss model is modified to take into

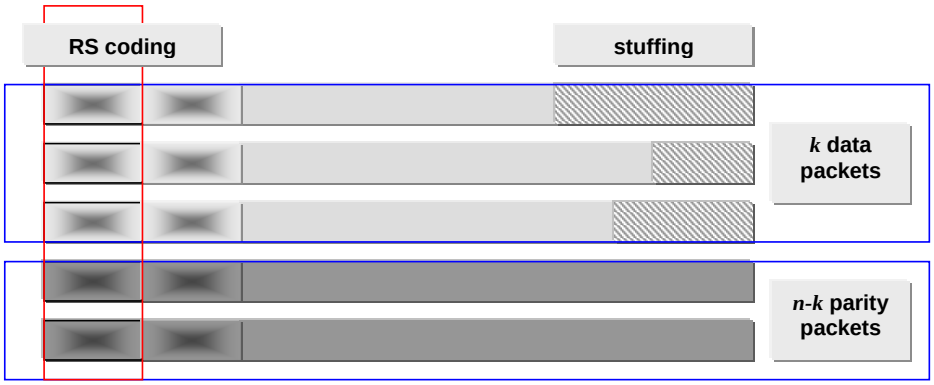


Fig. 6.3. Reed-Solomon coding across packets for error protection against packet loss

account the error recovery of FEC. A media-dependent adaptive FEC protection scheme is designed by tracking the modified hypothetical distortion. For simplicity, $(k+1, k)$ FEC code is used, where rapid XOR-based FEC is implemented instead of the general RS coding technique. To guarantee the crucial information data always get protected from error corruption, a packet is always protected using FEC whenever it contains the picture or slice header information.

In [6], high-priority protection is addressed to the base layer. Multi-levels of error protection using FEC are addressed to protect the embedded FGS bitstream of the enhancement layer. In particular, the more significant bit planes in the enhancement layer automatically benefit from more error protection due to the entropy coding properties. Each source bit plane together with its FEC bits are assumed to be encapsulated into one packet. The entropy coding assigns shorter VLC (variable length coding) codes to the more significant bit planes. Hence, if the packet size is fixed, the more significant bit planes can obtain better error protection due to the larger amount of FEC bits for error protection stuffed to the respective packets.

Two parameters are critical for erasure packet networks: the packet loss ratio p_B and the average burst length L_B, as formulated in Eqn. (6.1) and (6.2) for the Gilbert channel model. In [184], the packet transmission behavior is modelled as a renewal error process, and thus both p_B and L_B can be easily derived if FEC is used in the packet level for error protection. From the derived model, it can be observed that: (1) For a given FEC code (n, k), the packet loss ratio p_B increases with k as the amount of protection decreases; (2) FEC protection becomes less efficient for

burst loss traffic. When a very large amount of protection is used, the burst length L_B stays close to k. With the decrease of the amount of protection, L_B is getting closer to the real channel burst length pattern. If the amount of error protection is in between, a maximum exists for L_B which is less pronounced for burst packet loss.

A JSCC rate-shaping method is developed in [185], aiming to achieve the optimal rate-distortion performance. The bitstream generated by both the source coding and the channel coding in the application layer is passed through a rate shaper to satisfy the channel bandwidth requirement. The rate shaper fulfills its shaping task by selectively sending a certain portion of the bitstream to the channel. The rate shaping is done in a rate-distortion optimization manner.

An adaptive JSCC scheme is proposed in [186], where MDC is used for unequal error protection. It takes into account the concealment strategy at the decoder. A macroblock is protected only if the error concealment performance is expected to be significantly poorer than that of error protection through the use of multiple description codes. For macroblocks to be protected, motion vectors in the inter-coding mode and the DC coefficients in the intra-coding mode are repeated in two channels, while the remaining DCT coefficients are quantized using a multiple description scalar quantizer [76]. This enables that less than twice the amount of information is needed to be sent over the two channels.

Back channel information is helpful for FEC schemes since error protection can be used adaptively to variable channel characteristics. An adaptive FEC error protection scheme by using back channel information is presented in [187], where the

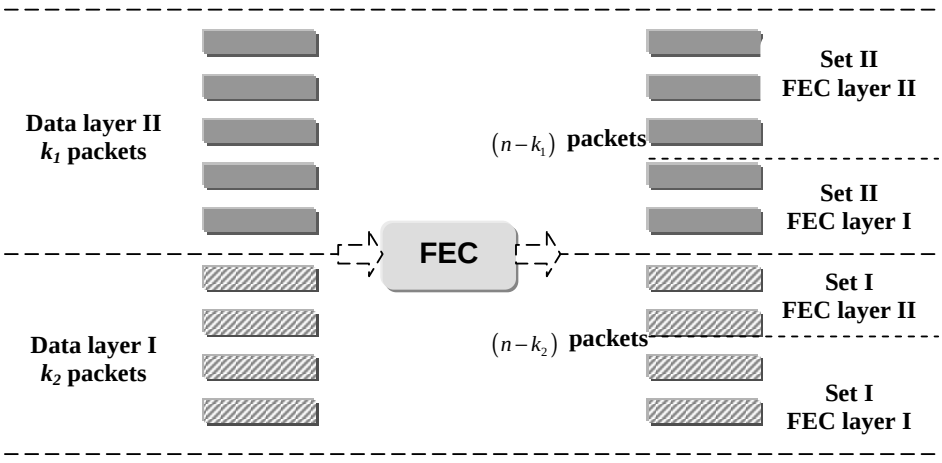


Fig. 6.4. Layered FEC for layered source coding data

channel loss process is modelled as a continuous time Markov chain. The use of FEC will inevitably cause transmission delay, since channel decoding can only proceed after receiving the complete BOP data. Hence, a trade-off exists between the time delay and the data loss rate. In [187], an analytical model that captures both time delay and data loss rate of the lossy channels is established, helping determine the number of parity packets for FEC based on the feedback information.

Independent work has been reported by [188] and [189], where layered FEC schemes are proposed for video multicast over heterogeneous networks. A pseudo-ARQ mechanism is further introduced to extend FEC to hybrid FEC/ARQ techniques in [188]. In this work, ARQ is simulated by continuously transmitting delayed parity packets from the sender to additional multicast groups.

As shown in Fig. 6.4, for k data packets, the $(n-k)$ parity packet data can be partitioned into several layers of FEC codes, with each layer still possessing the

maximum distance separable (MDS) property. A receiver can subscribe to one or more layers to obtain a specific error protection level to adapt to its own channel features such as bandwidth and packet loss conditions. For protecting layered source data that are generated by a scalable source coder, the FEC layers can be partitioned into different sets, each protecting a different layer of the source data. For a receiver, it is only needed to determine to how many FEC layers are to subscribe from each set, based on the receiver's channel characteristics. The layered FEC technique is particularly suitable for video multicast applications due to the extremely diversity of the communications channels between the sender and its receivers. The trade-off between delay and transmission quality needs to be balanced to satisfy various receiver requests.

6.1.3 ITU-T Standard - H.263+

ITU-T Recommendation H.263 Version 2, abbreviated as H.263+, is the very first video coding standard to support both circuit-switched and packet-switched networks [56]. In particular, H.263+ includes some significant features that are most suitable to very low data rate video coding. The recommendation specifies sixteen negotiable coding options, denoted as annexes, to further improve coding efficiency and support additional capabilities including error resilience and error protection. Many features provided in H.263+, especially the features provided in the annexes, have been modified and included in the most recent video coding standard H.264/AVC [37].

A. Video Compression Features in H.263+

H.263+ includes a hybrid video compression mechanism where DCT and motion compensation techniques are used. For each frame of a given video sequence, H.263+ first partitions it into macroblocks, each containing 16×16 pixels. One macroblock is further divided into four 8×8 blocks, and the two-dimensional DCT is implemented in the block level. H.263+ allows INTRA/INTER mode decision made in a macroblock-by-macroblock basis. Except the very first frame of the sequence encoded in INTRA mode, which is denoted as I frame, all the remaining frames are either P picture in forward prediction mode, or B picture in bi-directional prediction mode. A single row or multiple consecutive rows of macroblocks can be grouped into a Group of Block (GOB). Thus the video syntax of H.263+ is arranged in a hierarchical structure with four primary layers, in the order of from the top to the bottom being picture, GOB, macroblock, and block.

Annex D, Annex F, and Annex I are three optional mechanisms regarding the coding efficiency performance improvement. The unrestricted motion vector mode, known as Annex D in the standard recommendation, allows motion vectors to point outside the pictures, and it also extends the range of motion vectors from the default value $[-16, 15.5]$ to $[-32, 31.5]$.

The advanced prediction mode, known as Annex F, provides the one/four motion vector selection for each macroblock. In the default mode of H.263+, one motion vector is associated with one macroblock, and a differential coding scheme is used.

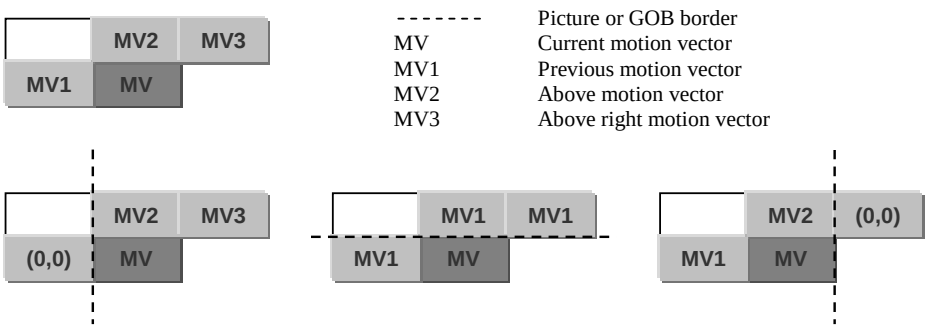


Fig. 6.5. H.263+ motion vector prediction

As shown in Fig. 6.5, the motion vector predictor for the current macroblock is the median value of three candidate predictors from the neighboring macroblocks. Only the difference between the current motion vector and the predictor is coded and transmitted.

If the four-motion-vector mode is turned on by Annex F, a motion vector is obtained for each of the four blocks in one macroblock, and motion prediction is used to each motion vector associated with each block. At the same time, the overlapped block motion compensation (OBMC) scheme is used, where motion compensation is implemented by a weighted sum of three motion compensated predictions for each luminance block. Each motion compensated prediction is specified by the respective motion vector associated with the current block or one of the two designated adjacent blocks.

The advanced INTRA coding mode, referred to as Annex I, focuses on the improvement in coding efficiency for the macroblocks coded in INTRA mode. By turning on Annex I, a prediction for an INTRA block to encode is obtained first

from the neighboring INTRA blocks. Moreover, the modified inverse quantization for INTRA coefficients is used, and a separate Variable Length Coding (VLC) table is designed.

The annexes regarding coding efficiency in H.263+ aim to further remove the redundancy inherit in the bitstream than as is in the default mode, which results in more dependency across different portions of the bitstream. If a macroblock is damaged by channel errors, it is more likely to cause error propagation through motion compensation and differential coding.

B. Error Resilience Features in H.263+

H.263+ provides annexes that are related with error resilience, including Annex K - slice structured mode, Annex N - reference picture selection mode, and Annex R - independent segment decoding (ISD) mode.

Annex K introduces the slice structure to replace the original GOB layer in the syntax. When the slice structure is exploited, every macroblock in a frame is assigned to one and only one slice. There are two submodes in Annex K: the rectangular slice submode (RS) and the arbitrary slice ordering submode (ASO). A slice can be either a rectangular area in units of macroblocks, or contain a sequence of macroblocks in lexicographic order, which is indicated by the RS submode. In contrast, all the slices of one frame can be encoded in lexicographic order, or in any arbitrary order, which is designated by the ASO submode. To guarantee the arbitrarily ordered slices to be successfully decoded when ASO submode is turned on, Annex K prohibits motion

vector prediction, OBMC, and the Advanced INTRA coding mode indicated by Annex I from being implemented across slice boundaries. Annex K provides a more flexible structure, compared to GOBs, so that frames can be segmented into slices at needed. Moreover, the headers of slices can be used as resynchronization points in the bitstream. Notice that Annex K does not prevent dependency across slice boundaries in the reference picture for motion prediction purposes.

By turning on Annex R, dependency across different segments in one picture can be further prevented. Annex R regards a single GOB, a number of consecutive GOBs, or one slice as one segment. Besides the same constraints imposed on the segment as on the slice in Annex K, Annex R requires that the segmentation for all frames that use motion estimation shall be the same as that in its reference picture. However, although Annex R allows each segment to be decoded independently at the receiver, the encoding process of the segment is not completely independent as long as motion estimation is used. Therefore, Annex R effectively prevents spatial error propagation, but it cannot prevent temporal error propagation if a segment contains INTER-mode macroblocks.

Annex N provides a method known as "NEWPRED", allowing the encoder to choose the reference picture for the INTER picture prediction. There are two scenarios in this optional mode, depending on whether the back channel information is available or not. If a feedback channel is available, the encoding process can exploit the information to select the known-as-correctly transmitted picture or segments for prediction. Back channel messages are returned either through VideoMux sub-

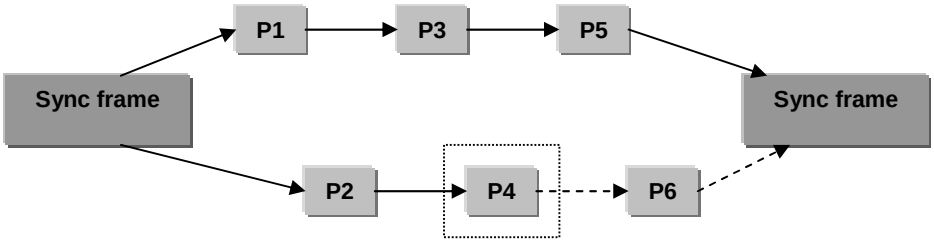


Fig. 6.6. Video Redundancy Coding (VRC) with two threads and three frames per thread

mode, where the decoder uses the same channel as the encoder to transmit the back information in the opposite direction, or by using a separate back channel.

Under the circumstances where information can only be transported in one direction, Annex N employs the so-called Video Redundancy Coding (VRC) method to suppress the temporal error propagation, as described in Fig. 6.6. VRC breaks a given source video sequence into more than one threads. Every picture is assigned to one of the threads, and thus each thread has a lower frame rate compared to that of the source video. Within a regular interval, all the threads are encoded separately, independent of each other, and then a Sync Frame is inserted regularly to merge the threads. Notice that the Sync Frame can be decoded even if only one thread within the interval of two Sync Frames stays intact. Therefore, if some threads are destroyed, the decoder can depend on those successfully received ones to reconstruct the bitstream. For example, in Fig. 6.6, if the fourth picture is lost or damaged, the second thread will not be used for the Sync Frame prediction.

H.263+ also includes Annex H - forward error correction for coded video signal for error protection by binary BCH codes (Bose-Chaudhuri-Hochquenghem codes). It is inevitable that the use of annexes for robust transmission will result in a substantial penalty with respect to coding efficiency. Therefore, it is worthy to know how the combination of the above annexes affects the overall performance of H.263+ coders in terms of the coding efficiency as well as the error resilience of the bitstream.

Notice that scalable coding is also supported by H.263+, which is mainly specified in Annex O - temporal, SNR, and spatial scalability mode.

6.2 Compression Optimization

In this chapter, we focus on the video codec with a fixed encoding data rate. All the source video sequences are in QCIF format, that is, each frame has 176×144 pixels, thus containing 11×9 macroblocks. Also, all videos are in 4:2:0 YUV format, i.e., 12 bits/pixel. For instance, if a given source video is encoded at 56 kbps and 6 fps, the data rate of the source video is $(12{\times}176{\times}144){\times}6 = 9,123,840$ bps, and the compression ratio is $9,123,840/56,000 = 32.5$, i.e., the video can be represented by 0.375 bits/pixel.

6.2.1 Evaluation of Annexes of H.263+ for Source Coding

As we discussed in Subsection 6.1.3, H.263+ provides optional modes that are related to coding efficiency and modes that are related to error resilience as well. In

this subsection, we will evaluate the combinations of five major annexes, referenced as Annex D, F, I, K, and N, to investigate how their combination affects the coding efficiency and the error resilient capabilities of the bitstream.

We know that Annex D, F, and I are the three annexes regarding the coding efficiency improvement. These optional modes aim to further remove the redundancy inherit in the bitstream than as is in the default mode, resulting in more dependency across different parts of the bitstream. Hence, the use of these annexes is more likely to cause error propagation through motion compensation and differential coding. Annexes K and N, on the other hand, are two optional modes to introduce error resilience to the H.263+ bitstream. The use of these annexes concerning robust transmission will inevitably result in a substantial penalty with respect to coding efficiency. Therefore, it is worthy to know how the combination of the above annexes affects the overall performance of H.263+ coders in balancing the coding efficiency and the error resilience of the bitstream.

In Fig. 6.7, we illustrate the performance of the H.263+ source coder with and without Annex D, F, I, K, and N turned on. Simulation results are obtained by using the video sequence *foreman* in QCIF format coded at frame rate of 6 fps and data rate of 56 kbps. The blue curve in the figure denotes the PSNR of each frame without the annexes in use, while the red curve indicates the results with all the five annexes on. It can be observed that the coding scheme with annexes in use achieves a little better performance, about 0.8 dB improvement in average. Notice that we leave the range of the motion vectors as the default value in the above two

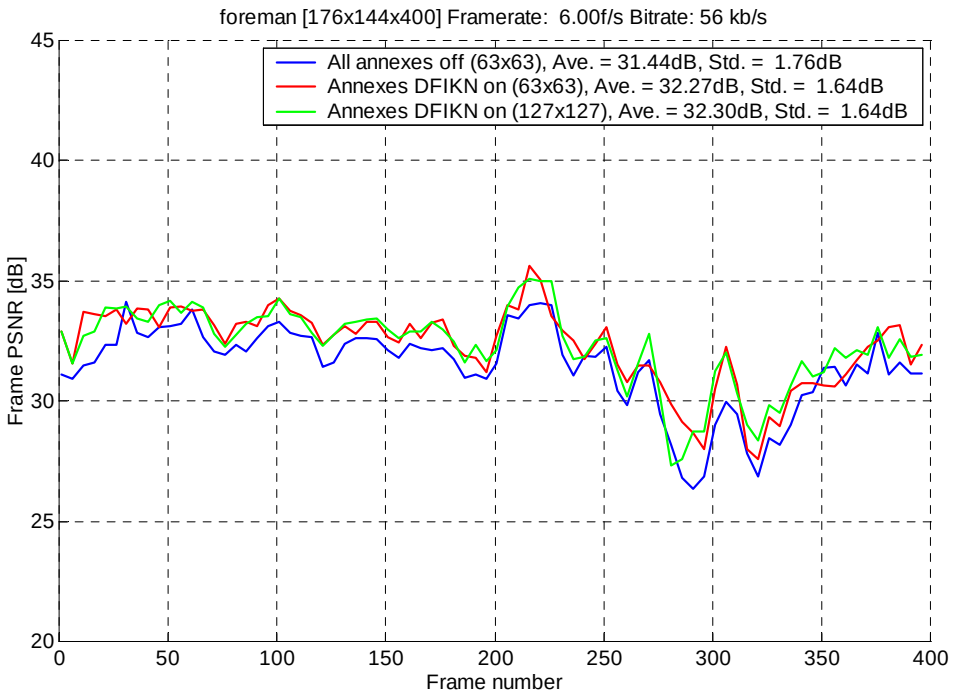


Fig. 6.7. Evaluation of annexes in H.263+ over *foreman*

experiments, which is denoted as 63×63 in the figure referring to the search area size for motion estimation. Furthermore, we take advantage of Annex D to extend the motion vector range to twice as large, which is denoted as 127×127. It is expected to achieve a better video quality by exploiting a larger search range since we have much more motion vector candidates to choose from, while at the price of paying more computation complexity. It turns out that almost the same quality level is obtained by the use of Annex D, which is believed to be a result of the mutual effects of different annexes with different functionalities.

Moreover, the above experiments are used to six other video sequences with a variety of motion change characteristics. Statistics are summarized in Table 6.1. The first column under each experiment presents the PSNR value averaged over the entire sequence, and the second column indicates the standard deviation of PSNR versus frame number. Table 6.1 demonstrates almost consistent results as obtained from *foreman*.

In summary, the combination of the efficient coding annexes with the error resilient annexes results in an overall compression performance similar to or slightly better than turning off the annexes. Considering the critical role that the error resilience capability of the bitstream plays for video transmission over error prone environment, we will leave all the five annexes on simultaneously in the following experiments.

Table 6.1
Evaluation of annexes of H.263+ over different video sequences

Video	Annexes off 63×63		Annexes DFIKN on 63×63		Annexes DFIKN on 127×127	
	Average PSNR (dB)	Standard deviation (dB)	Average PSNR (dB)	Standard deviation (dB)	Average PSNR (dB)	Standard deviation (dB)
claire	42.94	1.20	42.64	1.25	42.63	1.25
mthrdgthr	39.47	1.51	39.64	1.37	39.68	1.34
salesman	37.74	2.51	38.05	2.31	38.41	2.52
wireless	33.79	2.82	34.10	2.79	34.12	2.84
vfa	32.50	3.79	32.79	3.19	32.80	3.18
foreman	31.44	1.76	32.27	1.64	32.30	1.64
Laura	27.68	2.30	27.63	2.35	27.62	2.34

6.2.2 Evaluation of Rate-Distortion Operational Behavior of H.263+

In the previous subsection, the coding proceeds at one data rate. In this subsection, we evaluate the performance of the H.263+ codec and its annexes at various data rates through examining the rate-distortion behavior of H.263+. Instead of obtaining a distortion metric versus different data rates, average PSNR values are used to measure the decoded video quality, and hence the PSNR value curves rise with the increase of the data rates allocated to the source encoder.

As before, we use the H.263+ encoder to all seven video sequences with various motion change complexities. As shown in Fig. 6.8, each sequence is coded twice at one data rate, with and without the annexes in use in each case, and the PSNR values of a decoded video are obtained and averaged over the entire sequence.

Again, we observe similar results that the encoder with all the five annexes turned on performs a little better, but the improvement is very limited. However, if we use all the annexes, the H.263+ source coder can achieve a much higher degree of error resilience while maintaining the same level of coding efficiency compared to the case of no annexes turned on.

In addition, different video sequences have shown different distortions at a fixed data rate. For coding a source video at a particular data rate, the larger the PSNR value, the less motion activities the source video usually has, and thus the easier the video is to represent. In our experiments, *claire* is the simplest video sequence,

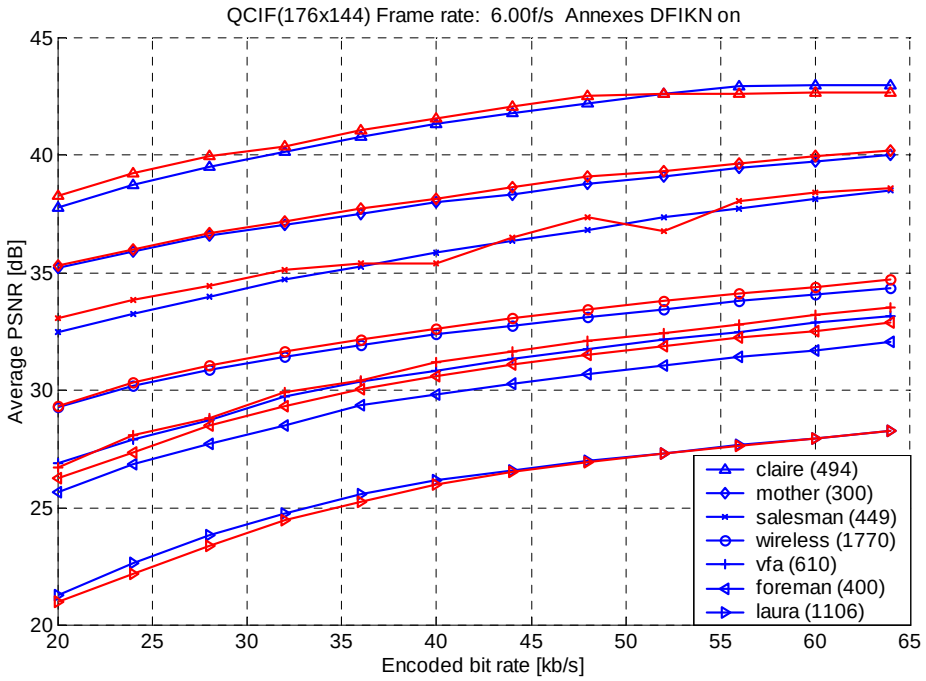


Fig. 6.8. Rate-distortion behavior of H.263+ codec over various video sequences

while *Laura* is the most complex one for it includes many scene changes and zoom-in-zoom-out motion changes that are hard to represent by motion compensation.

6.2.3 Evaluation of INTRA Refresh Period

In H.263+, INTRA/INTER decision is made in a macroblock-by-macroblock basis. As we discussed, without considering the robustness of the bitstream, the source coder usually chooses INTRA/INTER modes based on a rate-distortion sense. For the sake of robustness of the bitstream, however, it is likely to choose INTRA mode more frequently since INTRA macroblocks can serve as a resynchronization point and completely stop temporal error propagation. Nevertheless, coding efficiency will inevitably suffer from the more INTRA mode assignment for not taking advantage of motion compensation.

The macroblocks that are selected as INTER mode first by the source coder but then forced to be INTRA for the sake of robustness are referred to as forced INTRA macroblocks. In this subsection, we will examine how the forced INTRA mode affects the source coding efficiency. Later in this chapter, we will use the forced INTRA mode as an error resilience approach and present an evaluation of its performance over lossy wireless channels.

As we discussed, forced INTRA mode can be set in an adaptive way based on the video content or the channel condition if feedback channel information is available, or both, or just set regularly determined by the INTRA refreshment period in units of time. If INTRA refreshment is equal to 1/3 seconds, say, for encoding a video

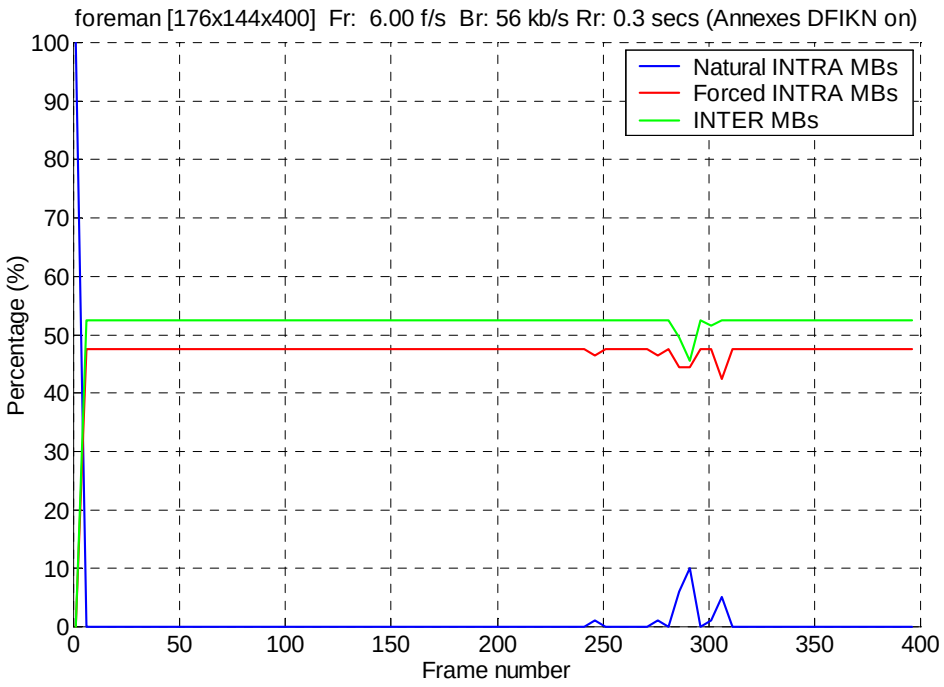


Fig. 6.9. Percentage of INTRA/INTER macroblocks of *foreman*

at frame rate of 6 fps, then every macroblock is forced to be INTRA mode exactly once every 1/3 seconds or every 2 frames. In other words, half of the macroblocks of every frame is coded as INTRA mode in average.

The percentage of naturally chosen INTRA mode, forced INTRA mode, and INTER mode macroblocks of each frame when encoding *foreman* at 56 kbps, 6 fps and with INTRA refresh period 1/3≈0.3 seconds is shown in Fig. 6.9. Considering the complex motion change nature of *foreman* and the terribly low percentage of nature INTRA macroblocks, we learn that INTER mode is much likely to be chosen by the encoder to efficiently represent the video data, implying that the use of forced INTRA might greatly impact the coding efficiency.

In Fig. 6.10, results of decoded video qualities are presented at different INTRA refresh periods. Statistics are still collected over seven video sequences. With the increase of INTRA refresh period, less INTRA modes are selected, and thus the videos are encoded in a more efficient way, resulting in higher average PSNR values of the decoded pictures. Furthermore, for those video sequences with very few motion changes such as *claire*, *mother-daughter*, and *salesman*, we can observe a more than 5 dB improvement if the INTRA refresh period is released from 1/6 secs to 1.5 secs. For those videos with much more complex motion characteristics, in contrast, such as *Laura*, forced INTRA mode has a limited impact on the coding performance, since even without forced INTRA selection, the encoder still prefers to INTRA because of the weak motion relationship between consecutive frames.

6.2.4 Matching Points between Rate-Distortion and INTRA Refresh

Moreover, we would like to know how much we have to pay for the introduction of forced INTRA modes regarding the coding data rates. We notice that the rate-distortion curves in Fig. 6.8 are obtained with the INTRA refreshment period set to the default value - 1 second but at a variety of coding data rates. In contrary, the INTRA refresh curves described in Fig. 6.10 are obtained over various INTRA refresh periods but at a fix data rate. With the decrement of the INTRA refresh period, the average PSNR value decreases accordingly, which is equivalent to a source coding process using a less data rate but keeping the INTRA refresh period constant. In other words, we introduce more error resilience to the bitstream by choosing

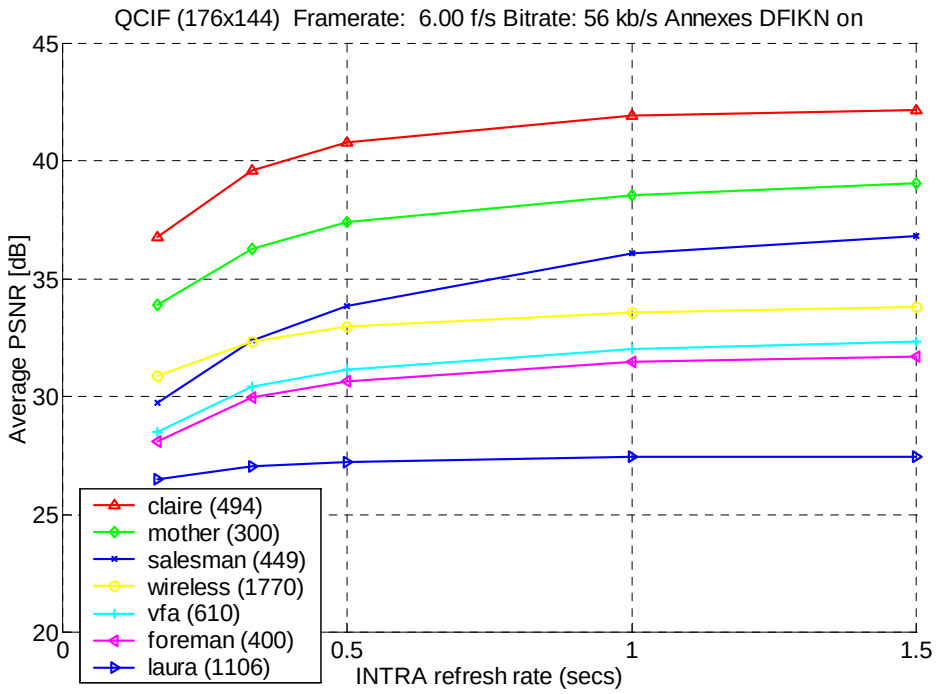


Fig. 6.10. Evaluation of H.263+ at different INTRA refresh periods

INTRA modes more frequently at a price of sacrificing data rates for pure source coding. Alternatively, we can choose a larger INTRA refresh period, achieving a better compression performance so that the same distortion can be obtained at a lower data rate and the remaining bits can be used for error protection using FEC. The goal here is to match the PSNR of the source coder with error resilience to a (lower bandwidth) version of the source coder without error resilience and additional FEC to match the final data rates.

Fig. 6.11 demonstrates how many bits have to be dropped if we choose more INTRA modes. The data in the figure are obtained by locating the "matching points" between the operational rate-distortion curves and the distortion-INTRA refresh period curves. For *wireless*, for example, if it is encoded at 56 kbps and INTRA refresh period of 0.5 seconds, the average PSNR value of the decoded pictures are around 33 dB, whereas it only requires 43 kbps to achieve the same performance with INTRA refresh 1 sec. Hence if error resilience is realized by INTRA refresh, 13 kbps in data rate has to be sacrificed.

6.3 Transmission over Wireless Lossy Channel Optimization

We have discussed that video transmission over wireless lossy channels suffers from burst bit errors caused by channel fading, which requires suitable methods to improve the robustness of the bitstream to combat the damages. We have mentioned three essential schemes in Subsection 1.1.5 of Chapter 1 to combat channel errors: error resilience, error control coding, and error concealment. The key problem we

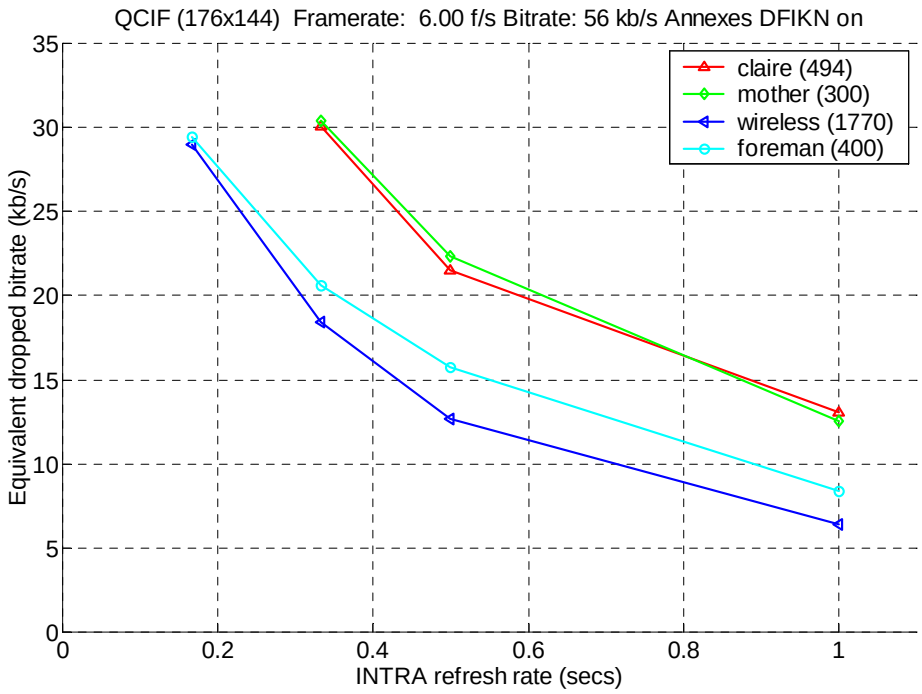


Fig. 6.11. Matching points between rate-distortion curves and INTRA refresh curves

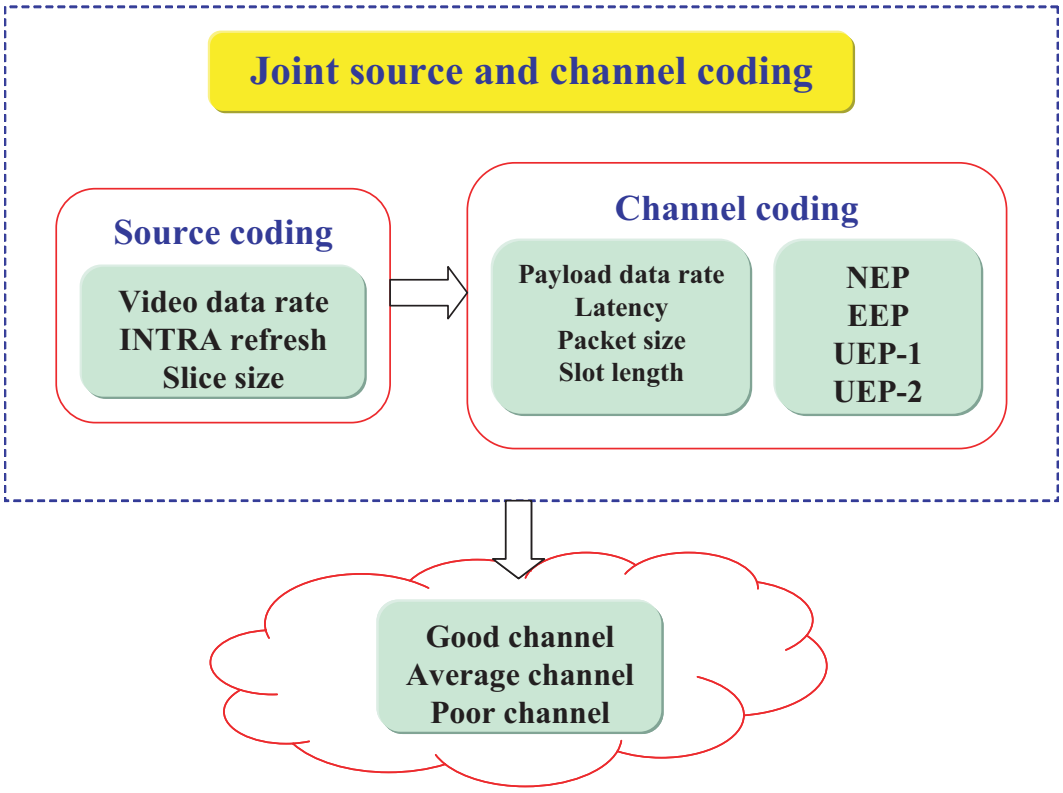


Fig. 6.12. Parameters related to joint source and channel coding optimization

need to solve is to decide the optimal bit allocation between source coding and channel coding, as well as the optimal bit allocation to introduce appropriate error resilience among the source coding elements. We fix the total data rate of the entire system, and evaluate different parameter settings for error resilience features within the source coding stage and FEC for channel coding under different channel conditions, as shown in Fig. 6.12.

In Fig. 6.12, the total data rate assigned to the encoder is denoted as the payload data rate, which includes the data rate allocated to the header information of packetization, the video data rate to source coding, and the rest data rate to FEC in the

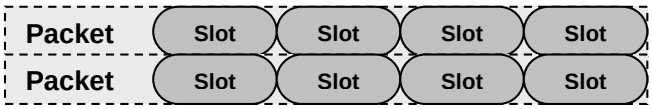


Fig. 6.13. A packet containing four slots

application layer for error protection. As shown in Fig. 6.13, bitstream generated by the source encoder is first packetized, with each packet including several equally sized slots. FEC realized by Reed-Solomon coding is then implemented across slots, and the parity slots are then grouped into packets and output to the bitstream.

Error resilience is introduced to the bitstream in the source coding stage, especially by exploiting INTRA refresh, slice structure (Annex K), and reference picture selection mode (Annex N). In particular, we are interested in adjusting the INTRA refresh period and the size of slice in order to achieve an optimal trade-off between source coding efficiency and error resilience capabilities.

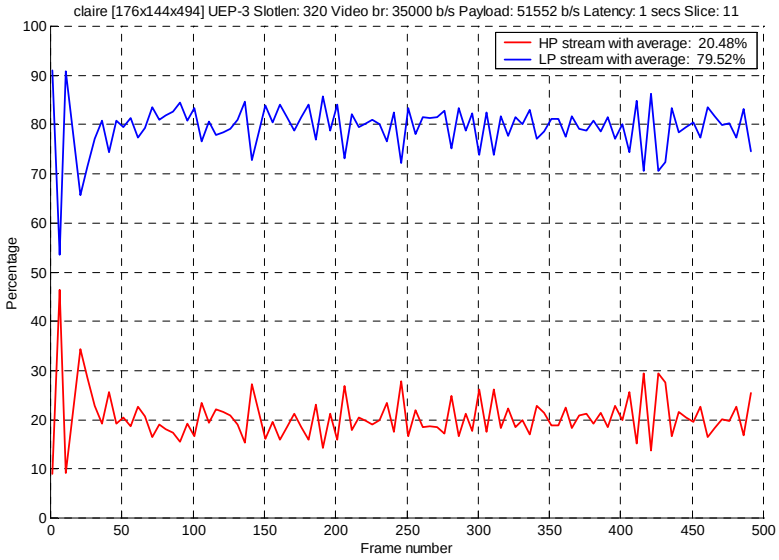
On the other hand, error control coding is implemented in the channel coding stage of the encoder. At this stage we employ FEC to introduce additional redundancy to the bitstream. We use Reed-Solomon (RS) coding as the FEC method. As we discussed in Subsection 6.1.2, Reed-Solomon codes are maximum distance separable codes, i.e., they are the only codes that are able to reconstruct erased symbols by knowing a small number of symbols at the decoder. We use Reed-Solomon coding across packets. The bitstream generated by source coding is first packetized according to the designated packet size, with each packet containing a fix number of slots

determined by the slot length. Reed-Solomon coded parity packets are then generated and transmitted following the information packets. Our system adaptively determines the total data rate of Reed-Solomon codes by obtaining the ratio of the coded video source data rate by the total payload data rate. The packet size is a critical parameter, since for the scenario of video transmission over burst bit error lossy channels, the smaller the packet is, the less chance it is impacted by channel errors, but the more bits the header information for packetization is to cost. Moreover, we implement interleaving across slots to mitigate the effects of burst bit errors, and hence slot length and latency become two important parameters, where the value of latency determines the length of a time window within which we realize the interleaving. We cannot assign a too large latency although it will benefit the interleaving procedure, since video streaming is very time sensitive. In summary, packet size, slot length, and latency are three elements that we are going to be concerned with for the channel coding stage.

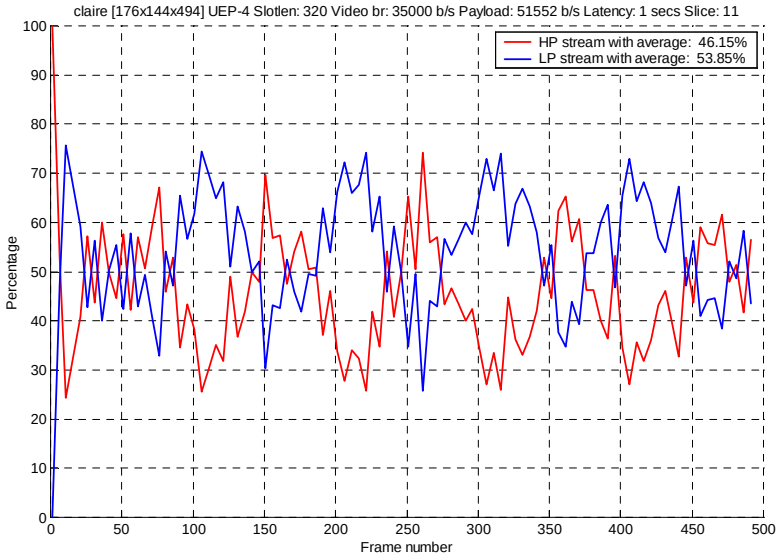
Further more, there are three modes for error protection by error control coding: None Error Protection, NEP, where error resilience is the only mechanism for the concern of the error robustness of the bitstream and no additional error protection is placed; Equal Error Protection, EEP, where all the information data produced by source coding are equally protected by FEC; Unequal Error Protection, UEP, where error protection is only used for a portion of the bitstream, or different data rates of FEC codes are used for different portions. Usually, if UEP is used, the bitstream is broken into two flows - higher priority data and lower priority data, where the

former flow is protected by a chunk of channel codes yet the latter one with no channel coding protection. By the use of UEP, the available data rates can be better taken advantage of by giving more protection to those data that are critical to the decoding procedure.

Moreover, we designed two sub-modes for UEP, UEP-1 and UEP-2. UEP-1 includes header information together with motion vectors in the higher priority flow, while UEP-2 includes header information, motion vectors, as well as slice-wise INTRA data in the higher priority flow. As discussed before, INTRA refresh data can stop temporal error propagation and thus play a more significant role than INTER data regarding error resilience. In UEP-2, we intended to place INTRA data into higher priority flow in units of macroblocks. Nevertheless, since the entire bitstream is divided into two flows, every segment of data in each flow has to be facilitated a header to indicate its relative position in the original bitstream. Therefore, we exploit a larger segment - slice to reduce the additional header information demanded by the operation of flow partition. If a slice contains at least one INTRA mode macroblock, the whole slice is taken as a part of the higher priority data. However, since the average percentage of the INTRA macroblocks in each slice fluctuates a lot from one frame to the other, the bit allocation between two flows cannot maintain a steady level, as observed from Fig. 6.14 and Fig. 6.15, where the percentages of the two flows for each frame are presented when the H.263+ coder is used to encoding *claire* and *foreman* respectively.

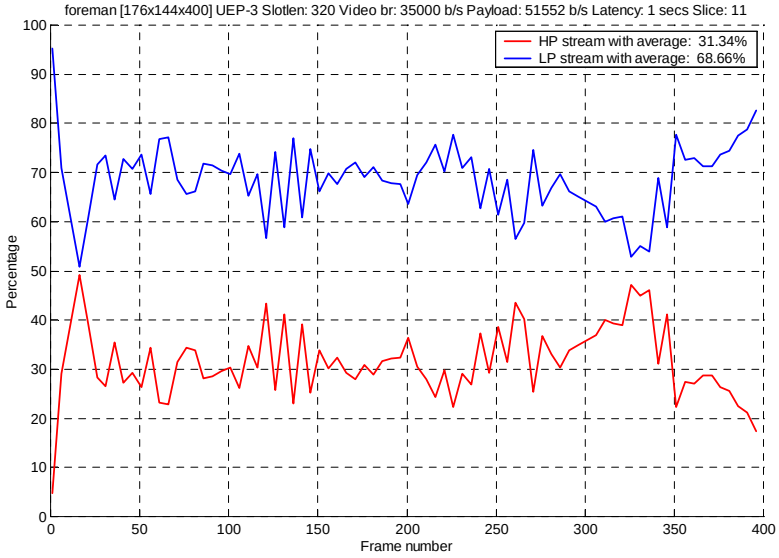


(a) UEP-1 for *claire*

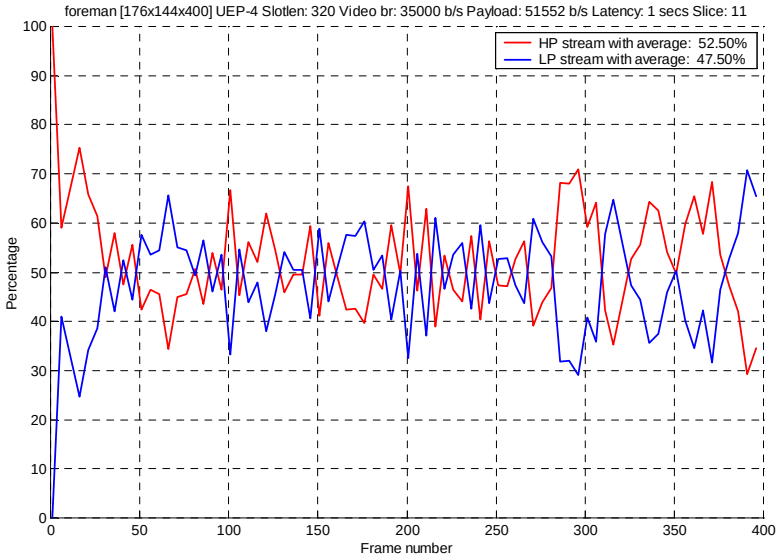


(b) UEP-2 for *claire*

Fig. 6.14. Bit allocation between two flows when UEP is used for (*claire*)



(a) UEP-1 for *foreman*



(b) UEP-2 for *foreman*

Fig. 6.15. Bit allocation between two flows when UEP is used for (*foreman*)

Notice that error concealment is used in our experiments to reduce the effects of channel residual errors, but it is beyond of the interests of our evaluation.

Three different channel conditions are considered: good channel condition with BER (bit error rate) level at 10^{-5}, average channel condition with BER at 10^{-4}, and poor channel condition with BER at 10^{-3}. Firstly, we will exploit the matching points, which are discussed in Subsection 6.2.4, to compare the performance of error resilience realized by forced INTRA with that of error control coding realized by Reed-Solomon coding for video transmission under the first two conditions. Secondly, we will see that if channel condition is very poor, the decoded video quality is unacceptable without employing error protection by channel coding. At this time, we will adjust the encoder parameters to achieve an optimal combination for error protection. Finally, we will derive several metrics to measure the distortion caused by channel errors as opposed to the average PSNR value.

6.3.1 Matching Points between Error Resilience and Error Control Coding

Recall that we obtain matching points by comparing two different schemes designed for robust video transmission in an error-prone environment in Subsection 6.2.4, one of which exploits INTRA refresh to introduce error resilience to the bitstream, while the other uses less forced INTRA in order to save data rates for error protection by error control coding. The matching points are achieved in a manner that guarantees both of the schemes demonstrate the same decoded video quality

for video transmission in an error-free environment. In this subsection, we exploit the matching points to evaluate and compare these two schemes regarding the error robustness performance of video transmission over an error-prone environment.

We choose *mother-daughter* and *wireless* for simulation, as typical representatives of two categories of source videos - the former one with a low amount of movement while the latter one with more complex motion changes and several scene changes. Since channel bit error occurs in a random way, all statistics are obtained over 10 runs. Moreover, considering that there are only 300 frames in *mother-daughter*, and thus the generated bitstream is not long enough to demonstrate the corruption impacted by channel burst bit errors, we concatenate *mother-daughter* five times to create a 1500-frame video sequence.

For all the simulations, we encode the source videos at a total data rate 56 kbps and a frame rate 6 fps. We choose two matching points for *wireless* corresponding to INTRA refresh period equal to 1/3 and 0.5 seconds, and three matching points for *mother-daughter* corresponding to INTRA refresh of 1/6, 1/3, and 0.5 seconds respectively. As to the error protection by error control coding, the EEP mode is used, i.e., the entire bitstream is equally likely to be protected by error control codes.

Simulation results of *wireless* are given in Fig. 6.16 and Fig. 6.17, each demonstrating the performances of two schemes at a special matching point under a particular channel condition. The curves in the figure illustrate the PSNR values versus frame numbers, where the red curves denote the PSNR values obtained when error-free, and the blue ones show the PSNR when error presents. If no error occurs, it is

observed that the two schemes obtain exactly the same rate distortion performance. Whenever an error occurs, blue curves will drop from the red curve, thus indicating the corruption caused by lossy channels.

In Fig. 6.16, the coded bitstreams are transmitted over the good channel with BER of 10^{-5}, and the coding parameters for both two schemes are set as in Table 6.2. It can be observed that at this matching point with INTRA refresh period equal to 0.5 seconds, there is no error occurred when the EEP mode, i.e., the error control coding scheme is used, while there is some tiny errors occurred when the NEP, i.e., the error resilience scheme realized by forced INTRA is used. Considering that the frequency of error occurrence is so few and the lasting time of each error period is so small, the resulting errors by the use of the NEP mode can be ignored. As is discussed, the NEP mode does not require additional delays at the server and additional software at the client, and also completely compliant with the standard. Therefore, based on the results shown in Fig. 6.16, it can be concluded that if the channel condition is good enough, we can approach error protection by using an appropriate error resilience scheme.

Fig. 6.17 shows the results when the bitstreams are transmitted under average channel condition with BER of 10^{-4}. We notice that if we still use the matching point when INTRA refresh is equal to 0.5 seconds, errors will occur to both schemes. When the INTRA refresh is down to 1/3 seconds, there is no error occurred when employing the EEP scheme, as shown in Fig. 6.17, since more error protection is used by allocating more bits to the error control codes. The corresponding coding

parameters are set as in Table 6.3. For the NEP scheme, however, even half of the macroblocks are forced to be encoded in INTRA mode, errors are hit much more frequently when the channel condition gets worse. We notice that each error lasts a very short time period because of frequent INTRA refresh. Nevertheless, since error resilience schemes have no capability to correct errors, if the BER of lossy channels gets larger, errors will occur more frequently. Therefore, we need to use error control coding for error protection for an average channel condition.

In summary, the error resilience scheme is easy to implement, but it works well only under good channel conditions. When the channel conditions get worse, we have to employ error control codes to improve the reliability of the bitstream. From the experimental results, we also notice that if the channel condition is not very bad, we can achieve a very good error protection performance by using a small portion of Reed-Solomon code data rate.

6.3.2 Error Protection by FEC under Poor Channel Condition

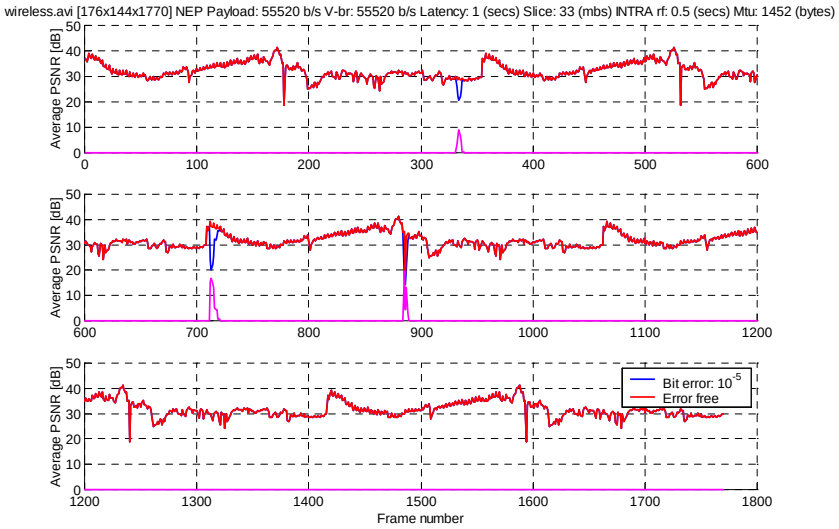
As is observed from the previous section, with the increase of the BER of the wireless channel, error protection only by the error resilience scheme results in an unacceptable decoding video quality. For the poor channel condition where the BER is equal to 10^{-3}, we have to use error control coding to obtain a better error reliability performance for the bitstream. On the other hand, we know that an (n, k) Reed-Solomon code can correct up to $\lfloor \frac{n-k}{2} \rfloor$ symbol errors and up to $(n-k)$ symbol erasures, or combinations thereof with each error counting as two erasures. If the

Table 6.2
System parameters for Wireless transmitted over good channel condition of BER 10^{-5}

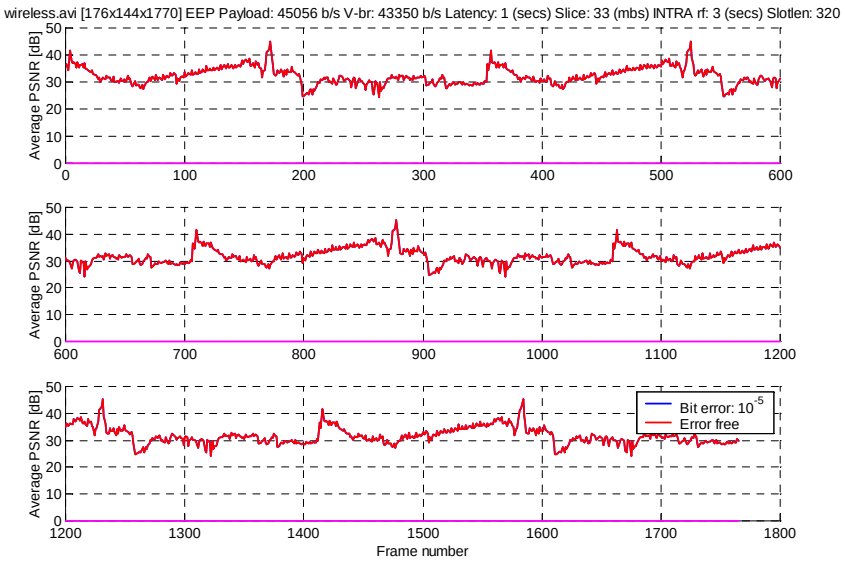
	Payload data rate (b/s)	Video data rate (b/s)	INTRA refresh (secs)	Slice size (mbs)	Latency (secs)	Packet size (bytes)	Slot length (bytes)
NEP	56,000	55,520	0.5	33	1	1,452	1,452
EEP	56,000	43,350	3.0	33	1	1,452	320

Table 6.3
System parameters for Wireless transmitted over average channel condition of BER 10^{-4}

	Payload data rate (b/s)	Video data rate (b/s)	INTRA refresh (secs)	Slice size (mbs)	Latency (secs)	Packet size (bytes)	Slot length (bytes)
NEP	56,000	52,800	1/3	33	1	1,452	320
EEP	56,000	37,546	3.0	33	1	1,452	320

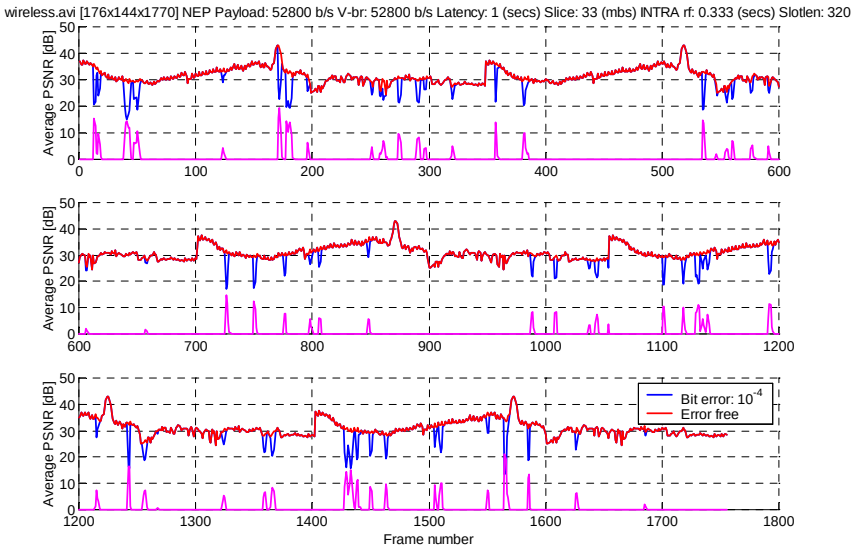


(a) NEP

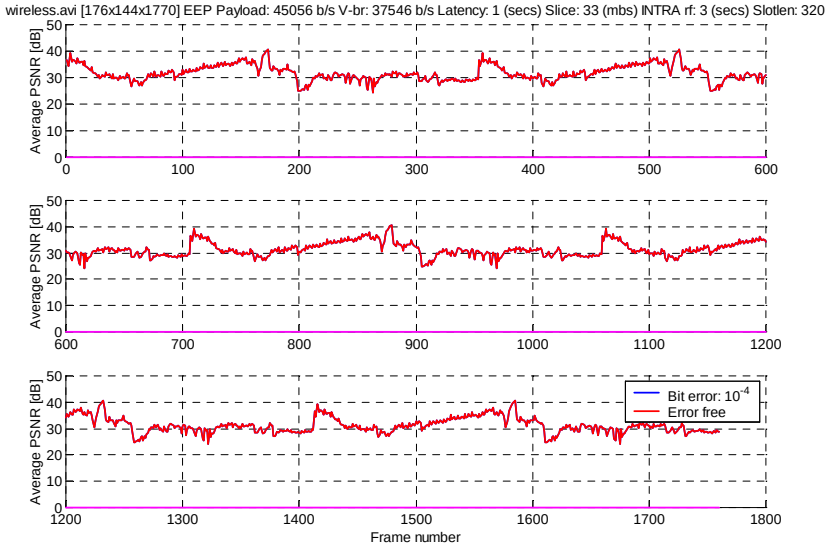


(b) EEP

Fig. 6.16. Evaluation of matching points under good channel condition (*wireless*)



(a) NEP



(b) EEP

Fig. 6.17. Evaluation of matching points under average channel condition (*wireless*)

BER of the lossy channel gets much larger beyond the error correction capability of the Reed-Solomon codes, the error control codes become ineffective and thus the decoding visual performance will seriously degrade. As is shown in Fig. 6.18(a), we leave the coding parameters set as in the first row of Table 6.4, which are almost the same as that in Table. 6.3 except that we allocate 2kbps more data rate for the error control codes, but increase the BER of the channel from 10^{-4} up to 10^{-3}. It can be observed that even though we use more error protection for the bitstream, the decoding video quality is terribly corrupted by the lossy channel burst errors.

As discussed at the beginning of this section, we can adjust the coding and packetization parameters to obtain a better performance as shown in Fig. 6.18(b), with the coding parameters set as in the second row of Table 6.4. From our experiments, we find out that there are two parameters that are most critical to the error reliability performance of the bitstream - INTRA refresh period and latency. By adjusting the INTRA refresh period, we can introduce more error resilience elements to the bitstream. Therefore, with combination of error resilience and error control coding schemes, we can exploit INTRA refresh to prevent error propagation and gain resynchronization in case the error control codes become ineffective. On the other hand, as we discussed at the beginning of this section, interleaving across slots can decentralize error effect and thus is very suitable to deal with burst errors. The higher the latency is, the larger window we can exploit to implement the interleaving, and thus the better decoding video quality can be achieved. We have to admit that, however, interleaving has two negative effects - large latency requires large memory at both

encoder and decoder as well as large time delay for the encoding process. As shown in Fig. 6.18(b), we decrease the INTRA refresh from 3 seconds to 0.5 seconds, and increase the latency from 1 second to 4 seconds, with the other parameters remaining the same as the worse case. At this time, we achieve a very good performance even though the BER of the channel is as high as 10^{-3}.

For video transmission over good and average channel conditions, we have already shown that a small portion of error control codes can achieve good error reliability and thus EEP is enough for error protection. While for the poor channel condition, it is worthy to take a look at the UEP mode since more error protection by error control coding is demanded. By adopting the UEP mode, it is possible to save more data rates for the source coding since only a portion of the data needs to be protected, so that the overall distortion is possible to be reduced.

The experimental results implemented by UEP-1 submode are given in Fig. 6.19(a), while the results by UEP-2 submode given in Fig. 6.19(b), with the coding and packetization parameters set in Table 6.5. Notice that we remain all the other parameters exactly the same as in Table 6.4 with EEP, except that we allocate more data rate for the source video coding since less amount of data are protected. It can be observed that the decoding video quality by UEP-1 is not acceptable, while with UEP-2, since we protect the header information, motion vectors, together with the slice-wise INTRA data, we achieve a much better performance regarding the error reliability of the bitstream. However, it can be observed from Fig. 6.19(b) that almost every frame has been impacted by channel errors with UEP-2 since only a

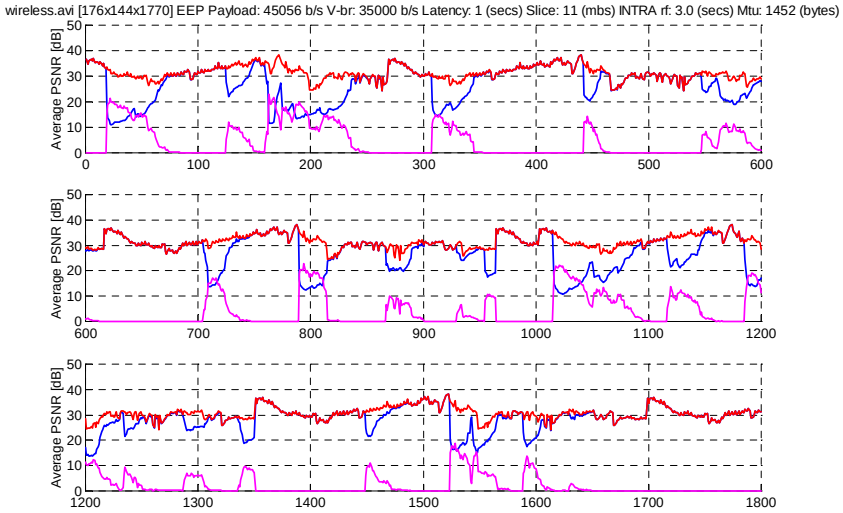
Table 6.4
System parameters for Wireless transmitted over poor channel condition of BER 10^{-3} with EEP mode

	Payload data rate (b/s)	Video data rate (b/s)	INTRA refresh (secs)	Slice size (mbs)	Latency (secs)	Packet size (bytes)	Slot length (bytes)
Worse	56,000	35,000	3.0	11	1	1,452	320
Better	56,000	35,000	0.5	11	4	1,452	320

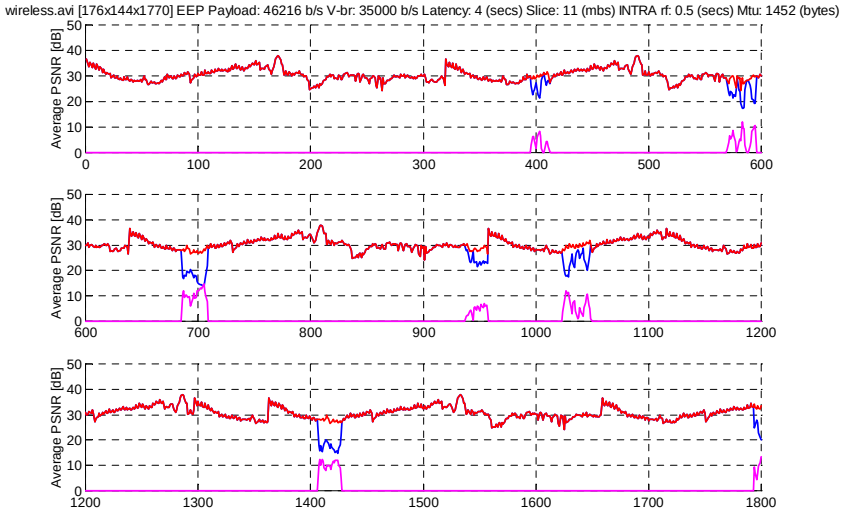
portion of the data are protected. We have to admit that the subject visual quality of the error effects that are caused by source coding due to source quantization is different from that caused by channel errors. Subjectively, distortion due to source quantization is more blurring, while distortion due to channel errors is deforming since the lost of synchronization results in block shifting. In particular, if a series of frames are consecutively corrupt by channel errors, the visual quality is very annoying. Therefore, if the channel condition is very bad, we need to make use of the combination of error resilience and error control coding for error protection, and all the information would better be protected. Notice that it is not wise to draw a conclusion that EEP might achieve a better performance than the UEP mode, since UEP mode depends on the design of the flow partition schemes.

Table 6.5
System parameters for Wireless transmitted over poor channel condition of BER 10^{-3} with UEP mode

	Payload data rate (b/s)	Video data rate (b/s)	INTRA refresh (secs)	Slice size (mbs)	Latency (secs)	Packet size (bytes)	Slot length (bytes)
UEP-1	56,000	40,000	0.5	11	1	1,452	320
UEP-2	56,000	43,000	0.5	11	4	1,452	320

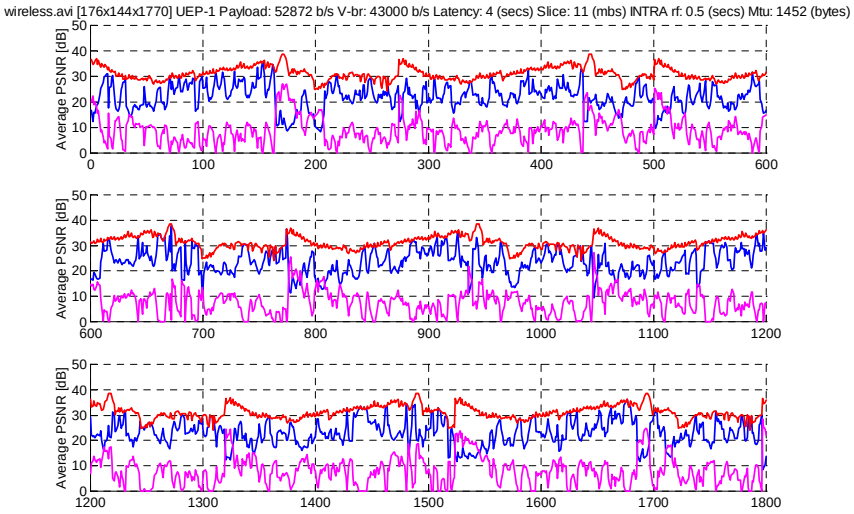


(a) Worse results

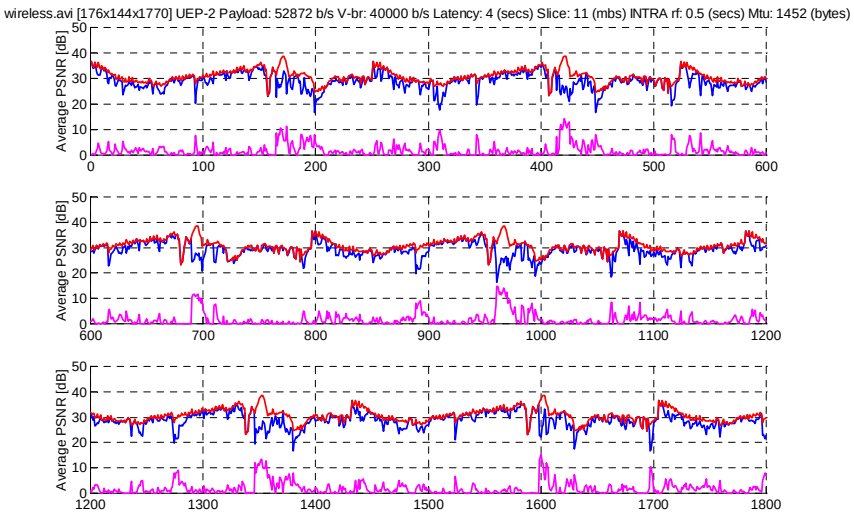


(b) Better results

Fig. 6.18. EEP mode under poor channel condition with BER 10^{-3} (*wireless*)



(a) UEP-1



(b) UEP-2

Fig. 6.19. UEP mode under poor channel condition with BER 10^{-3} (*wireless*)

6.3.3 Metrics to Measure the Distortion Caused by Channel Errors

To evaluate the distortion due to the random wireless burst channel errors, we develop five metrics as follows:

Metric I: Average degraded PSNR caused by channel error over the entire sequence:

$$D_{\mathrm{cI}} = \frac{1}{N}\sum_{i=1}^{N} \left|\mathrm{PSNR}_{\mathrm{error\_free}}(i) - \mathrm{PSNR}_{\mathrm{error\_hit}}(i)\right|, \tag{6.3}$$

where N denotes the number of coded frames, $\mathrm{PSNR}_{\mathrm{error\_free}}(i)$ denotes the PSNR value of the ith frame when error free, as indicated by the red curves from Fig. 6.16 through Fig. 6.19, and $\mathrm{PSNR}_{\mathrm{error\_hit}}(i)$ indicates the corresponding PSNR value when error is presented, as shown by the blue curves in the figures.

Metric II: Average degraded PSNR caused by channel error over those frames hit by errors:

$$D_{\mathrm{cII}} = \frac{1}{N_{\mathrm{error}}}\sum_{i\in Q_{\mathrm{error}}} \left|\mathrm{PSNR}_{\mathrm{error\_free}}(i) - \mathrm{PSNR}_{\mathrm{error\_hit}}(i)\right|, \tag{6.4}$$

where Q_{error} denotes the set of corrupted frames, which is a subset of the set containing the entire sequence of frames Q. We have $N = \mathrm{card}\{Q\}$ and $N_{\mathrm{error}} = \mathrm{card}\{Q_{\mathrm{error}}\}$.

Metric III: Percentage of frames damaged by errors:

$$D_{\mathrm{cIII}} = \frac{N_{\mathrm{error}}}{N}\times 100\%. \tag{6.5}$$

Metric IV: Probability of error corruption per frame:

$$D_{\mathrm{cIV}} = \frac{m_{\mathrm{error\_run\_length}}}{N}\times 100\%, \tag{6.6}$$

where $m_{\text{error\_run\_length}}$ denotes the number of run-length of consecutive frames that are corrupted by channel errors. In fact, this metric presents the ratio of the number of times the transmitted video signal attacked by the channel errors over the total number of frames. It is the asymptotical probability of the chance that each frame might be attacked by the channel errors.

Metric V: Average run-length in units of number of frames that are corrupted by channel error:

$$D_{\text{cV}} = \frac{1}{m_{\text{error\_run\_length}}} \sum_{i=1}^{m_{\text{error\_run\_length}}} l_i, \tag{6.7}$$

where l_i denotes the number of frames falling into the interval $\left[t_1^{(i)}, t_2^{(i)}\right]$, indicating the length of the ith run-length of consecutive corrupted frames. For each $t \in \left[t_1^{(i)}, t_2^{(i)}\right]$, we have

$$\begin{aligned}
\text{PSNR}_{\text{error_free}}(t) &> \text{PSNR}_{\text{error_hit}}(t), \\
\text{PSNR}_{\text{error_free}}(t_1^{(i)} - 1) &= \text{PSNR}_{\text{error_hit}}(t_1^{(i)} - 1), \text{ and} \\
\text{PSNR}_{\text{error_free}}(t_2^{(i)} + 1) &= \text{PSNR}_{\text{error_hit}}(t_2^{(i)} + 1).
\end{aligned}$$

We use the above five metrics to evaluate the distortion due to channel errors for the experimental results in Fig. 6.18 and Fig. 6.19, where EEP and UEP modes are adopted respectively under poor channel condition. The evaluation results are given in Table 6.6 and Table 6.7.

It can be observed that Metric V is closely related to the INTRA refresh parameters, since every time the signal is attacked by channel errors, it is the INTRA coded data to serve as a resynchronization point to prevent error propagation.

Table 6.6
Evaluation of error protection modes by various metrics - I

Error protection mode	Average PSNR when error free (dB)	Average PSNR when error present (dB)
EEP (better)	30.30	29.95
EEP (worse)	31.05	26.95
UEP-1	31.00	23.00
UEP-2	30.70	28.92

Table 6.7
Evaluation of error protection modes by various metrics - II

Error protection mode	Metric I (Average channel distortion overall)	Metric II (Average channel distortion when error > 5dB)	Metric III (Percentage of error frames %)	Metric IV (Probability of error corruption for one frame)	Metric V (Average run-length of corrupted frames)
EEP (better)	0.57	9.53	5.09	0.0093	5
EEP (worse)	4.29	11.83	33.12	0.0175	19
UEP-1	8.13	10.66	68.83	0.0929	7
UEP-2	1.85	8.60	8.68	0.0317	3

6.4 Conclusions

In this chapter, we describe the following contributions:

- We have presented a thorough evaluation of the joint source and channel video coding methodology from two points of view: source coding design for error resilience and channel coding for error detection and recovery. We investigated the current ITU-T video compression standard, H.263+, for 3G wireless transmission. In particular, we concentrated on error resilient features provided within the standard and forward error correction (FEC) to find the optimal combination of various system parameters under different lossy channel conditions. Furthermore, we investigated new metrics other than the common average PSNR to evaluate video distortion caused by the combination of source compression and channel errors.

In our future work, we will explore the combination of the five metrics we developed to obtain a better evaluation of the overall performance for a joint source and channel video coding scheme.

7. ERROR RESILIENCE OF VIDEO TRANSMISSION BY RATE-DISTORTION OPTIMIZATION AND ADAPTIVE PACKETIZATION

7.1 Introduction

In this chapter we address the problem of video transmission over packet networks. In particular, our schemes are designed to cope with packet loss during transmission across packet networks [190]. Packet loss can result in quality degradation of the transmitted compressed video stream. As we discussed in the first chapter, the current video coding standards such as H.263+/H.26L/H.264 use motion estimation and differential coding, which result in error propagation due to the widely existing dependency between different parts of the bitstream. This problem has attracted great attention recently due to the rapid growing demand for Internet video streaming services [172, 182, 191].

7.1.1 Overview of Error Resilient Video Coding

As we discussed in Subsection 1.1.5 of Chapter 1, error resilience is an error protection scheme that introduces error resilient elements at the stage of source coding to mitigate error propagation. Since source coders aim to reduce both spatial and

temporal redundancy to obtain an efficient signal representation, two kinds of error propagation are resulted when video signals are transmitted under an error-prone environment: spatial error propagation and temporal error propagation. Differential coding, in-frame prediction, and VLC cause spatial error propagation, while motion estimation results in temporal propagation. Moreover, since temporal error propagation implies that a certain amount of pixels in the current frame are damaged due to its referring to the badly decoded pixels in the reference frame, spatial error propagation that occurs to the reference frame will indirectly cause further temporal propagation to the current frame and the following inter-coded frames.

Strategies developed for error resilience can be classified into three categories: (1) Schemes used at the video analysis level of the video source coding stage, to reduce or completely prevent dependency across different portions of the bitstream and thus to combat error propagation due to in-picture prediction or motion compensation; (2) Schemes that explicitly introduce redundancy either to reduce the impact of channel errors or for the sake of error concealment; (3) Schemes developed at the entropy coding level of the video source coding stage, to combat error propagation due to the use of VLC. Error resilient entropy coding schemes include Reversible Variable-Length Coding (RVLC) [192], Error-Resilient Entropy Coding (EREC) [193], fixed-length coding [194], and semi-fixed-length coding [195]. In this dissertation, we will mainly focus on the first two schemes. Always, error resilience is realized at the sacrifice of coding efficiency. It either does not as fully remove

the redundancy inherent in the video signal as is used by the pure source coding schemes, or introduces additional redundancy to the bitstream.

It is an effective way to mitigate error propagation by judiciously inserting resynchronization points in the bitstream. Three kinds of information can serve as a resynchronization point: (1) Header information, including picture header, GOB header, or slice header; (2) INTRA coded macroblock data; (3) Synchronization markers in the bitstream. In [196], a bidirectional synchronization scheme is proposed combined with unequal error protection for different bit planes of encoded digital images. The most frequently used resynchronization point in video coding is the INTRA coded data in preventing both spatial error propagation and temporal error propagation. INTRA coded data are completely independent of any other portions of the bitstream. In spite of their error resilient capability of preventing error propagation, INTRA coded data also contribute to the most bit-consuming portion in the bitstream since INTRA coding does not use any temporal redundancy inherent in video signals. Where and when to introduce INTRA data to the bitstream is a significant and attractive problem to obtain the optimal amount of resynchronization data that maximize the end-to-end video quality. Many studies have been contributed to this problem to wisely introduce INTRA mode for balancing the coding efficiency and error resilience performance.

INTRA data can be inserted periodically, and the insertion period is determined by the INTRA refresh rate parameter. Blocks can be INTRA updated in a raster order, or in an adaptive manner based on the image characteristics and channel con-

ditions. In [197], an error sensitive metric is designed to characterize three scenarios of bit errors due to error corruption and error propagation, and INTRA modes are introduced based on this metric for encoding any macroblock by an H.263+ compliant source encoder. In [198], feedback channel information is exploited for error tracking. Whenever the decoder detects an error, it notifies the transmitter the starting address of the damaged macroblock via the feedback channel. The encoder then marks the entire area that might be affected by the damaged macroblock as zero-valued, thus preventing the following coded data from further referring to this area. A low-rate reverse channel is assumed available in [199], where error tracking is realized by a pixel-based backward motion dependency analysis.

In [183], video quality degradation caused by spatial and temporal error propagation is analytically modelled. Slice headers are inserted if the hypothetical spatial loss reaches a predefined threshold, and INTRA modes are used if the temporal loss rate increases above the given threshold. Moreover, an INTRA refresh upper bound is defined to guarantee that every pixel be INTRA coded within a certain time period. Both [200] and [182] addressed the decision of INTRA mode using a rate-distortion optimization framework, where the metrics of distortion take into account the impact of error propagation. A theoretic model of the overall video communication framework is presented in [172], where the fixed INTRA update rate and error control coding rate are considered as the two most critical system parameters. Based on this model, the optimal INTRA coding decision is made for a given channel condition.

A novel error resilience tool, known as "Scattered Slices," was presented in the video standard H.26L [201]. As opposed to the traditional slice structure that contains raster-scan ordered macroblocks, it proposes a slice structure that consists of macroblocks in a scattered manner such that one block is at least surrounded by another block that belongs to a different slice. Therefore, if one block is lost as a result of the loss of its own slice, at least a neighboring block may be available for the use of error concealment. The new slice structure inevitably reduces the coding efficiency due to the use of a less efficient in-picture prediction. Nevertheless, it generally obtains a high-quality error resilience performance at the cost of introducing less than 10% additional data rate.

A simple example of realizing error resilience by explicitly introducing redundancy for the sake of error concealment is that motion vectors are also associated with INTRA macroblocks. An error resilience scheme is developed in [202] by using the idea of data embedding to help implement error concealment. Data embedding, which is also referred to as data hiding, is a methodology to embed additional information to a host multimedia signal, usually without requiring additional transmission bandwidth and additional storage source [203]. Data embedding can be used to various applications, including copyright protection and multimedia authentication. The work in [202] aims to improve error recovery performance by protecting the most significant information such as motion vectors and macroblock coding modes. The scheme uses the half-pixel motion estimation mode in video coding standards such as H.263+ to embed two bits in each motion vector of an INTER-coded macroblock. To

protect the significant information in the current frame, a certain amount of parity bits are first generated and then embedded into the following coded frame. These parity bits can then be extracted at the decoder to help reconstruct the lost data in the current decoded frame.

In [204] and [205], redundant parity-check motion vectors or DC coefficients are inserted at the video source coding stage to combat channel burst errors. A Double-vector Motion Compensation (DMC) scheme is proposed in [206]. Instead of searching for the motion vector only in the adjacent preceding frame, DMC predicts the current block using the weighted sum of two forward motion compensated blocks from adjacent two preceding frames. Hence, an acceptable video quality can still be obtained even if only one reference frame is available, and a much better quality is obtained if both reference frames are successfully received. Moreover, an efficient DMC-based error concealment scheme is developed, by observing that the two motion compensated predictions for a same block are generally quite similar to each other. Thus one prediction can be used for error concealing the other prediction.

In [207], the importance of bidirectionally predicted frame, i.e., B frame, is in-depth analyzed. Usually the impact of data loss in B frames is barely noticeable, since they are isolated by other I or P frames and contain the smallest amount of bits such that data losses have a lower chance to affect these frames. B frames do not cause temporal error propagation since they are usually not referenced by other frames. Moreover, the bidirectionally coded macroblocks sometimes carry both forward and backward motion vectors that point to the best matched locations in two anchor

frames. The best matched location in the future reference frame can be backward tracked to the best matched location in the past reference frame through the B mode macroblocks. Based on this idea, error concealment can be realized in an inter-frame manner.

Error resilience realized by DMC is similar to that by the use of B frames, since basically both of them obtain two motion vectors for one macroblock from two reference frames. Hence the two best-matched locations in the two anchor frames can be considered close to each other. Error concealment for data recovery in the anchor frames can be implemented with the help of the two motion vectors carried by the same macroblock. Both schemes are motivated from the same concern of improving coding efficiency, and both are modified for the sake of error resilience.

As we discussed in Subsection 1.1.5 of Chapter 1, Multiple Description Coding (MDC) provides an efficient coding structure for error resilience such that the decoded video quality is only related to the amount of descriptions received instead of whichever descriptions have been really received. A multiple description motion coding algorithm is proposed in [208] to enhance the robustness of the motion vectors against transmission errors. The main idea originates from the Overlapping Block Motion Compensation (OBMC) scheme, which implements motion compensation for each block using the motion vectors associated with the current block as well as the neighboring blocks. The proposed scheme judiciously partitions the nine motion vectors belonging to the current block and its eight neighbors into two sets and transmits them separately over two channels. Motion compensated prediction

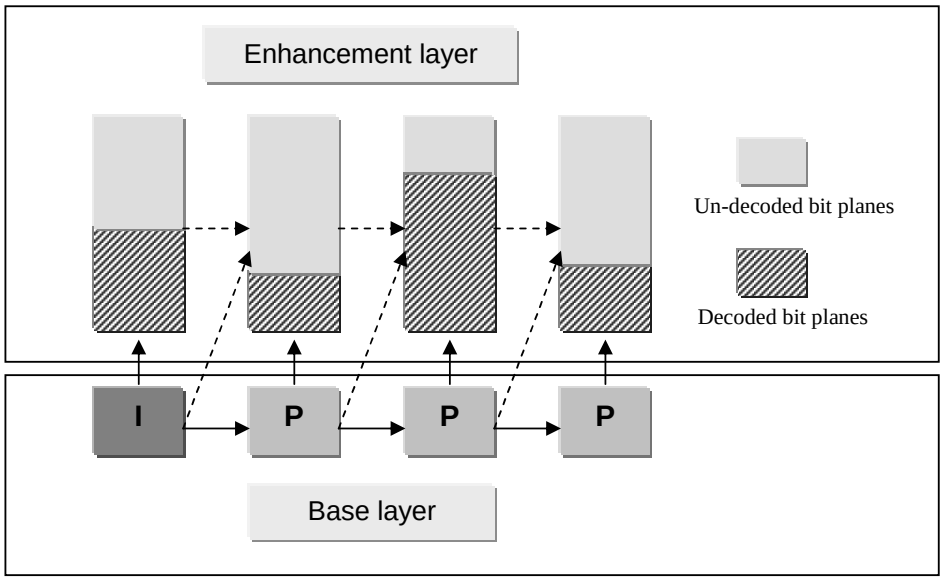


Fig. 7.1. Scalable coding structure and Fine Granularity Scalability (FGS)

can thus be implemented either by the use of the motion vectors in one of the two sets, which is called side prediction, or by using both of them, if available, to realize the so-called central prediction. Obviously with central prediction can the decoder achieve a better reconstruction, whereas a worse but acceptable video quality with side prediction if one channel has been affected by errors. Moreover, the motion estimation process is optimized in minimizing the central prediction distortion subject to the constraints imposed on both the motion vector data rate and the side prediction distortion by using a Lagrangian optimization technique.

It has been shown that scalable coding is suitable for unequal error protection, and error resilience in scalable coding has been widely studied [7, 182, 189, 209]. As shown in Fig. 7.1, the bitstream generated by a layered scalable encoder usually contains two layers: base layer and enhancement layer. The base layer contains the

lower resolution or frequency information of a video signal as well as other important data such as motion vectors, while the enhancement layer includes the refined information. In contrast to the multiple descriptions generated by MDC, the base layer plays a much more significant role. Without the availability of the base layer, the enhancement layer cannot be decoded. The coding of the base layer is the same as the non-scalable coding process. For the enhancement layer, two approaches can be used: the first one only uses the base layer information to obtain the prediction for the enhancement layer, while the second one only relies on the previous reconstructed enhancement layer information. Obviously the first approach suffers from a lower coding efficiency performance while maintaining a good error resilience capability. The second approach can encode video signals more efficiently, but inevitably suffer from temporal error propagation. H.263+ allows the prediction for the enhancement layer of each macroblock either from the base layer macroblock, or from the previous enhancement reconstruction, or a linear combination of above two predictions [56]. In [182], coding modes of the enhancement layer are determined by rate distortion optimization to trade-off the coding efficiency with robustness.

One disadvantage exists in traditional scalable coding, where the enhancement layer data are either successfully decoded or have to be completely discarded due to the channel capacity constraint or the impact from channel errors. Recently, the Fine Granularity Scalability (FGS) technique is proposed to provide a novel scalable coding mechanism with more adaptability to the variable bandwidth channel characteristics [4]. FGS consists of a single enhancement layer coded in a progressive

(fine granular) manner. The enhancement layer bitstream hence can be truncated at any data rate and still be successfully decoded, as described in Fig. 7.1. Thus the encoding process is implemented only once but provides a variety of decoding data rates ranging from the base layer data rate to the maximum coding data rate budget. FGS has already been included in the video coding standard MPEG-4 [210].

Currently, the progressive encoded enhancement layer in FGS is INTRA coded, therefore, it possesses error resilient features to combat channel errors. In [4], it has been pointed out that error resilience can be further introduced by inserting resynchronization markers in the enhancement layer bitstream. Moreover, compared to conventional scalable coding techniques, FGS is more suitable to unequal error protection. In [6], a two UEP-based aspects are addressed regarding error resilience against packet loss in FGS, which are realized by a new unequal fine-grained loss protection (FGLP) scheme. FGLP not only considers the UEP techniques used to the base layer and the enhancement layer, but also considers transmission prioritization within the enhancement layer by placing unequal packet protection. FGLP assigns bit planes in the enhancement layer to different protection level segments and applies different levels of protection using FEC in order of the significance of the bit planes.

7.1.2 Overview of Operational Rate-Distortion Optimization

As discussed in Subsection 1.1.3 of Chapter 1, the fundamental problem in rate distortion optimization is to find the asymptotically achievable bound for the fidelity of a source representation under a given data rate constraint. In practice,

operational rate-distortion optimization is used, aiming to achieve an optimum for a set of practically obtained rate-distortion points by adjusting the parameters of the overall system. Operational rate-distortion optimization schemes have been widely used in the literature of video compression [211,212]. In [213], an SNR scalable video compression scheme is proposed by partitioning the DCT coefficients into several layers. Based on the observation that setting the least significant bits of a coefficient to zero is equivalent to subtracting a certain value from it, an optimization problem is formulated as how to optimally select the subtracted values for each coefficient in a rate-distortion optimization manner. Both the Lagrangian multiplier optimization scheme and the dynamic programming algorithm are used.

As the optimization problem formulated in Eqn. (1.1), for video transmission applications, JSCC requires an overall rate-distortion optimization to achieve the optimal bit allocation between source coding and channel coding elements, as well as the optimal bit allocation among source coding elements that introduce error resilience to the bitstream.

As discussed, for a constellation of rate-distortion points, rate-distortion optimization can be realized by dynamic programming to achieve a global optimization, which is an effective way to implement an exhaust search. The disadvantage is that this method is too computationally intensive. When a sufficiently large amount of operational rate-distortion points are obtained so that an approximately continuous rate-distortion curve is obtained, the Lagrangian optimization method is usually used instead. The disadvantage of this method is, however, that it cannot reach any

rate-distortion pairs that do not reside on the convex hull of the practically obtained rate-distortion points.

A new discrete rate-distortion combination scheme is proposed in [185]. The key idea of this work is to exploit the greedy algorithm to substitute dynamic programming. Combined with certain knowledge-based principles, the proposed scheme is computationally efficient but only achieves a sub-optimal rate-distortion solution.

An objective distortion metric is a metric to computationally measure the difference between the decoded signal and the original one. The Peak Signal-to-Noise Ratio (PSNR), as formulated in Eqn. (1.2), is a commonly used objective distortion metric, due to its simplicity and relative consistency with the subjective distortion. To serve the goal of operational rate-distortion optimization, a certain form of distortion prediction has to be used at the encoder side. For video compression where signal distortion is mainly caused due to quantization, a distortion prediction is relatively easy to obtain due to the deterministic characteristic of source quantization. For video transmission over error prone channels, however, an accurate distortion prediction at the encoder has to take into account various elements such as source coding quantization, channel errors due to packet loss, spatial and temporal error propagation, and the error recovery capability of the decoder [214]. This distortion prediction is very complicated and unlikely to be predictable. Hence, how to estimate and predict the distortion at the encoder such that it is approximately consistent with the true distortion to be obtained at the decoder is a key problem for rate-distortion optimization in video communication applications.

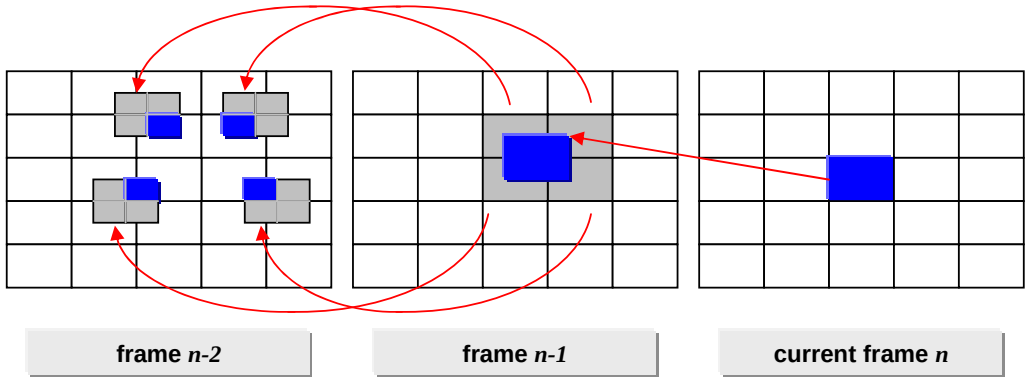


Fig. 7.2. Referenced area by motion estimation out of the constraint of macroblock boundaries

Both [183] and [200] independently address that the impact of temporal error propagation has to be analyzed in the pixel-wise resolution. This is done because motion estimation does not consider macroblock boundaries in the reference frame, but rather referencing areas of 16×16 pixels, as shown in Fig. 7.2. They both analyzed the impact of the pixel-wise error propagation behavior in a recursive manner. It is pointed out that only by estimating the reconstruction of each individual pixel to be obtained at the decoder can we accurately evaluate the error propagation and thus achieve an accurate distortion prediction at the encoder side.

A rate-distortion optimized adaptive INTRA update at the macroblock level is proposed to improve the robustness of the bitstream against packet loss in [200]. In particular, mode selection for each macroblock is implemented by jointly considering the error propagation effect for subsequent frames. The scheme thus aims to achieve a rate-distortion optimization for an entire group of frames. A recursive optimal

per-pixel estimate (ROPE) scheme is proposed for obtaining a distortion prediction at the encoder, which accurately takes into account the combining effects of source coding, error propagation, channel loss, and error concealment.

In [183], the error probability for each pixel is estimated by considering three kinds of impacts due to packet loss: (1) It is directly caused by a lost packet; (2) It is caused by a slice that partially (or completely) suffers from packet loss; (3) It is caused by referring to the damaged pixels in the reference frame. In [183], an adaptive error resilient encoding scheme is proposed. The commonly used MSE (or PSNR) is chosen as a distortion metric, but it is weighted by the likelihood of a packet loss occurrence, i.e., weighted by the estimated pixel-wise loss probability. The distortion is derived with the on-going encoding process. Whenever the accumulation of the estimated distortion leads to a video degradation above a predefined threshold, a resynchronization point is inserted into the bitstream. Furthermore, the FEC protection is considered and the video packet loss process is analyzed considering the error recovery capability of FEC. An adaptive FEC protection scheme is then designed by tracking the updated distortion estimation.

In [184], a subjective distortion is proposed which is developed based on the fact that the perceptual distortion varies exponentially with the scale quantization parameter of the encoder and is consistent with the human judgment. Furthermore, the relation between the distortion and the packet loss pattern is also derived, and an important conclusion is drawn that a uniform and independent loss process results in the worst case with respect to the amount of video quality degradation due to

packet loss. Therefore, the distortion decreases inverse-proportionally to the average burst length.

In [182], an operational rate-distortion optimized mode selection algorithm is developed for scalable coding, in which the Lagrangian method is used. An accumulated error concealment distortion is considered for the distortion prediction. The relationship between the Lagrangian multiplier and the quantization step is first formally addressed and analyzed for non-scalable coding in [28], and an approach that addresses the same relationship but for scalable coding is developed in [182]. These approaches are developed based on the observation that a close relationship exists between the rate-distortion trade-offs and the quantization levels.

In [191], rate-distortion optimization is used to packetized streaming media, where the decision of whether a packet is transmitted is made to meet a rate constraint while minimizing the end-to-end distortion. The key idea of the proposed scheme is based on mode design and selection, but candidate modes are designed for packets rather than for the original source coded data such as macroblocks [28, 182, 212]. The error cost function, similar to the error distortion, associated with each packet is derived as the cost of not delivering the packet to the destination. The optimal mode is then determined for each packet by minimizing the Lagrangian cost function. Moreover, it is addressed that the encoding and packetization of multimedia can be modelled as a single directed acyclic dependency graph, and thus a data packet is regarded as a node in the graph and the per-packet optimization is achieved to contribute to the overall optimization solution.

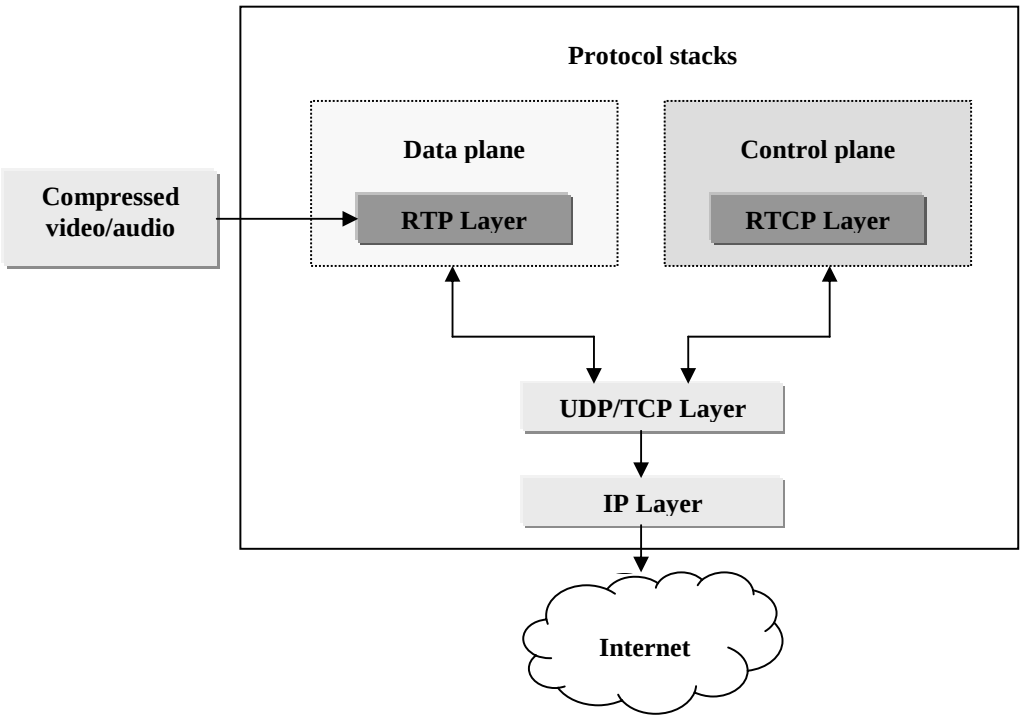


Fig. 7.3. Protocol stacks in multimedia streaming

7.1.3 Overview of Packetization

For video streaming over IP networks, bitstreams generated by the video source encoder are packetized in the transport layer and filled into the networks. Real-time transport protocol (RTP) and real-time control protocol (RTCP) provide end-to-end network upper-layer transport functions for streaming applications. They are running on top of the basic transport function providers - the user datagram protocol (UDP) and the transmission control protocol (TCP), as shown in Fig. 7.3 [215].

Functions supported by UDPs and TCPs include multiplexing, error control, congestion control, or flow control. In particular, checksum is used in all TCP and

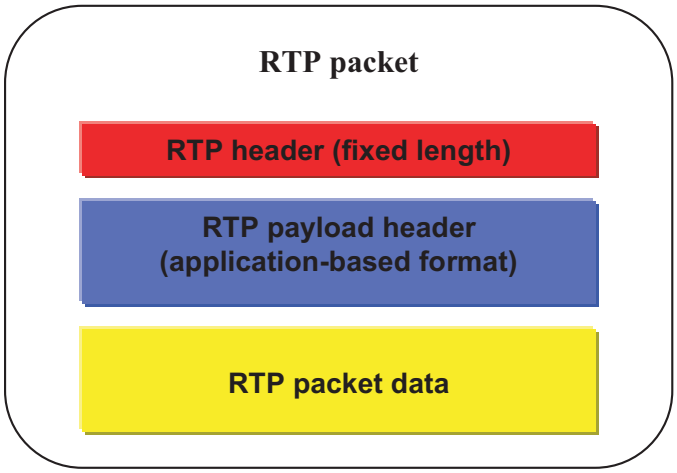


Fig. 7.4. RTP packet structure

most UDP implementations for bit error detection. If a single or multiple bit errors are detected in an upcoming packet, the packet will be discarded so that the upper layer protocols will not receive the corrupted packet. Notice that retransmission is allowed in TCP but not in UDP.

RTP, standardized by the Internet Engineering Task Force (IETF), is an Internet protocol that is standardized and designed to support end-to-end real-time media streaming applications over unicast or multicast networks [216]. RTCP is a companion protocol with RTP to provide QoS feedback information. In other words, RTP is a data transfer protocol while RTCP is a control protocol. The structure of an RTP packet is shown in Fig. 7.4, including the RTP header, the RTP payload header, and the encapsulated payload data.

RTP does not support QoS, but rather, carries important information of the packet in the RTP header to support media streaming. As described in Fig. 7.4,

the RTP header has a fixed length, which specifies significant properties including the time stamping, the sequence numbering, and the payload type identification. Therefore, RTP packets can be transmitted out of order, providing a more flexibility for video streaming. Moreover, by the inclusion of the sequence number field that is consecutive for sequentially transmitted packets, a packet loss becomes an erasure error from the perspective of the decoder, which is beneficial for error detection.

In [217] and [218], the RTP payload header format for H.263+ video bitstreams is specified. Three modes are designed for the payload header, with mode A supporting fragmentation at GOB boundaries, and a longer header in mode B or C allowing fragmentation at macroblock boundaries. Mode selection strategies are developed based on the desired network packet size and the encoding operations specified by the H.263+ encoder. The payload header contains critical coding information such as picture coding type, optional mode indication, quantization levels, temporal references, macroblock addresses, or motion vectors. The error recovery capability of a decoder hence depends on the capability of the decoder in using the payload header information provided within an RTP packet.

Ideal packetization schemes for video transmission have to trade-off packetization efficiency with error resilience performance of the packetized bitstream. For RTP packetization of H.263+ bitstream, as an example, if one frame is encapsulated into one packet, a shorter header will be resulted. However, one packet loss means the whole frame information will not be available at the decoder, thus resulting in a very poor decoded video quality for an increasing packet loss rate. In contrast, if

one GOB or slice fits into one packet, the loss of one packet will not cause much information loss, especially when facilitated by a decoder with good error concealment capabilities. Nevertheless, the smaller the packet size, the more overhead information will be generated. A large overhead is especially prohibitive in the very low data rate video transmission scenarios.

We would like to point out that another trade-off exists in terms of error resilience performance when designing the packetization strategies, which is the trade-off between the dependency and independency across packet boundaries. It is commonly acknowledged that guaranteeing every packet to be individually decodable is a key idea of packetization concerning error resilience. One way to achieve this is to partition the source video into a set of components and encode each component independently followed by forming a packet. Therefore, data packetization is an implicit way to realize resynchronization. Dependency across packets will inevitably cause error propagation. Here we give two examples of dependency as a result of the motion estimation process. The first example originates from the motion prediction included in H.263+, as we discussed in Subsection 6.1.3 of Chapter 6, where one motion vector might be predicted by three motion vectors associated to the neighboring macroblocks and only the difference between the original motion vector and its prediction is encoded. Thus, if the neighboring macroblock containing the motion vectors for prediction is lost, the decoding of current macroblock will be seriously impacted. The second example is from the Overlapped Block Motion Compensation (OBMC) scheme in Annex F of H.263+, where the motion compensated prediction

for one 8×8 block is obtained by a weighted sum of three predictions, using the motion vectors of current block and two designated adjacent blocks. Therefore, similar dependency is brought up as in the case of motion vector prediction. It seems that error resilience always requires as higher independency across packets as possible.

However, this is not always true. For example, Annex K and Annex R are highly recommended in [218]. If the Independent Segment Decoding (ISD) mode in Annex R is used in combination with the slice structure in Annex K, the rectangular slice sub-mode shall be enabled, and the dimension of each slice and the total number of slices shall remain the same between every two INTRA coded frames. Therefore, the motion estimation process of one segment in a frame is completely independent of all the remaining segments in the same picture. If one segment is lost due to the loss of the packet, there is no clue for the decoder to obtain any motion information about the lost segment from other received segments, and thus the conventional temporal-replacement error concealment method [200, 201] cannot be of any help. In this case, we might want to introduce additional redundancy to facilitate error concealment, such as associating a neighboring segment with the current segment's motion information since the independent segments behave in a manner as the INTRA coded data. Nevertheless, redundant information will unavoidably reduce the packetization efficiency.

In combination with RTP, RTCP is the control protocol to offer feedback channel messages, including the information regarding the quality of reception, such as the fraction of lost RTP packets, jitter, and delay. Therefore, RTCP facilitates the

realization of all the error protection schemes that require feedback channel information. In particular, the expected Packet Loss Ratio (PLR) p_B can be dynamically measured from the periodic RTCP receiver reports. For an (n, k) FEC code, the residual packet loss probability P_{loss} can be obtained based on PLR as

$$P_{\text{loss}} = \sum_{j=n-k+1}^{n} \frac{j}{n} \begin{pmatrix} n \\ j \end{pmatrix} p_B^j (1-p_B)^{(n-j)}. \tag{7.1}$$

An appropriate packetization approach combined with data partitioning is proposed in [7] for embedded 3D subband coding. The video bitstream is packetized into individually decodable packets of equal expected visual importance. Each subband is first partitioned into equally sized blocks, and then one block from each subband that contains different spatial information is chosen and grouped to shape one packet. Thus one packet loss only results in the loss of a segment of a specific subband.

A packetization scheme for embedded bitstream is presented in [219], and the major contribution is the proposed general paradigm for the optimal packetization design for embedded multimedia bitstreams using dynamic programming.

We would like to point out that the success of error resilience and error recovery are greatly dependent on the success of the error detection and error tracking techniques. Several mechanisms can be used to detect errors, which include but not limit to the use of a VLC parser or a syntax analyzer, a resynchronization point, or by checking whether the number of decoded DCT coefficients exceeds 64. If FEC is used, a Reed-Solomon decoder usually reports an uncorrected error. The probability

of undetected errors is usually very small, especially for a large value n chosen for FEC.

7.2 Adaptive Packetization

Techniques to address robust video transmission over packet networks need to simultaneously optimize three bit-allocation problems: the optimal bit allocation between source coding and channel coding, the optimal bit allocation to introduce appropriate error resilience into the bitstream, and the optimal bit allocation between the coded bitstream and the packetization overhead.

The use of INTRA coding mode prevents error propagation and obtains resynchronization for the bitstream. An INTRA coded frame is independent of all the other portions of the bitstream from both the encoding and decoding points of view. Nevertheless, INTRA coding is also the most bit-consuming scheme since it does not fully exploit the redundancy within the video signals.

As an alternative, we can exploit ISD (Annex R) in conjunction with the slice structure (Annex K) for the sake of error resilience. With packet fragmentation at the segment boundaries determined by ISD, we can guarantee that each packet can be independently decoded. Nevertheless, the "independency" of ISD is evaluated only from the decoder's perspective, not the encoder, since the dependency still exists across packets by the use of motion compensated prediction. If one packet is lost, all the information it carries will not be available, and thus all the packets whose motion information is based on that packet will be seriously affected.

In fact, for video transmission over packet networks, data loss always occurs in units of packets. Therefore, we only need to introduce error resilience to prevent dependency across packets, and we can take advantage of any dependency within each packet to improve the coding efficiency. Therefore, we propose a new packetization scheme, which prohibits any kind of dependency across the boundaries of packet while trying to take full advantage of the dependency within each packet.

The idea is as follows: In the source coding stage, we divide each frame into several segments, as is done in Annex R. We turn on any optional coding mode to exploit the dependency within each segment while treating the segment boundaries in a same way as with the picture boundaries. In the packetization stage, we place the segment into the same packet as its reference segment (if any). If it cannot be fit into that packet, it will be coded as INTRA instead and a new packet started. For example, if one GOB is taken as one segment, then the encoding and packetization processes obey the following principles:

- No dependency across the GOBs in one picture, which prohibits motion prediction, OBMC, and advanced INTRA block prediction across GOB boundaries;

- If there is at least one macroblock in a GOB is encoded as INTER mode, it is placed in the same packet as its reference GOB;

- If a GOB cannot fit into the packet containing its reference, all of its macroblocks are encoded as INTRA mode, and a new packet started;

- The number of GOBs in one packet is constrained by the predefined maximum packet size, and packet fragmentation is always implemented at the GOB boundaries;

- Each GOB can be referenced at most once for motion estimation.

For an H.263+ encoder with Annex R turned on, our packetization scheme is implemented to packetize a series of consecutive segments having the same position into the same packet. A packet always starts with an INTRA coded segment, or contains an INTRA segment while the remaining segments' motion vectors are obtained from that INTRA segment if backward motion estimation is employed.

7.3 Two-Layer Rate-Distortion Optimization

Similar to Annex R, we can take one or several GOBs or any rectangular slice as a segment in our scheme. For simplicity, in this chapter we take one GOB as one segment. We propose a two-layer rate-distortion optimization coding scheme to serve our packetization method. As in [182], we design four modes for each macroblock: INTRA, INTER, INTER4V, and SKIP. In Annex R, the motion vectors are only allowed to refer to the same area as the current segment in the reference picture. We loosen this constraint first to allow the current GOB to refer to any GOB at any position in the reference picture. This is done to improve the source coding performance.

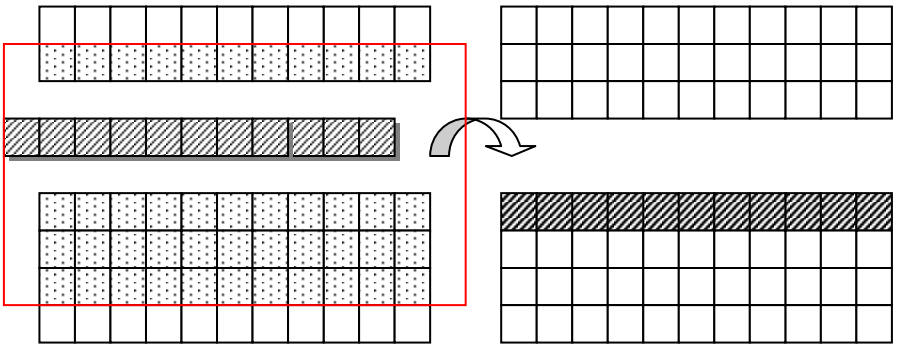


Fig. 7.5. Reference GOB selection by rate-distortion optimization

As shown in Fig. 7.5 and Fig. 7.6, for the current GOB (dark slashed blocks in the current picture) to be processed, we decide on the candidate GOBs for motion estimation based on the range of motion vectors. In H.263+, if Annex D (Unrestricted Motion Vector mode) is turned on, the search range can be as large as $[-32, 31.5]$ for QCIF pictures. Therefore, we select five GOBs - the GOB located in the same position as the current one and two above and two underneath as the candidate reference GOBs. These are the shaded regions in the reference frame enclosed within the bounding rectangle in Fig. 7.5. For each candidate reference GOB, we implement the first layer rate-distortion optimization scheme to determine the coding mode for each macroblock in the current GOB. Notice that the search window is limited within the reference GOB area. The second layer rate-distortion optimization is then used to select the final reference GOB out of the candidates by choosing the one that gives the minimum sum of the rate-distortion values of the macroblocks

$$k_{\text{opt}} = \arg\min_{k \in I} \sum_{\text{Curr\_GOB}} J_i^{(k)}, \tag{7.2}$$

where $J_i^{(k)}$ denotes the optimal Lagrangian rate-distortion value obtained in the first layer optimization, which is achieved based on the kth reference GOB for the ith macroblock in the current GOB. Finally, we encode each macroblock with the optimal mode obtained when the optimal GOB is referenced. We notice that the central area in each picture usually contains more significant information than the rest. Therefore, we start with the central GOBs and proceed to the top and the

bottom. For the nine GOBs in a QCIF picture ordered 1 through 9 from the top to the bottom, for example, we process the GOBs in the following order

$$\{5, 4, 6, 3, 7, 2, 8, 1, 9\}. \tag{7.3}$$

To further improve coding performance, we adopt Annex D and Annex F in H.263+ to treat each reference GOB. We extrapolate the edge area of the GOB, interpolate it to generate the half-pixel values, and employ the OBMC scheme. Since the unrestricted motion estimation mode is adopted, we have to signal the information regarding which GOB is selected to be the reference for the current GOB, which makes the proposed scheme not fully compliant with the H.263+ standard. As shown in Fig. 7.6, since the reference GOB has been extrapolated, its edge area is overlapped with the central area of the adjacent GOB. Therefore, from the motion vector itself, we cannot determine which GOB it refers to.

Considering that the above scheme is not fully compliant with H.263+, we simplify our scheme where the reference GOB is always the one in the same position as the current GOB, which is consistent with Annex R of the standard. From the experimental results present in the next subsection, we will see that this scheme is applicable since the encoder always tends to select the one in the same position except in the case complex motion or when scene changes occur.

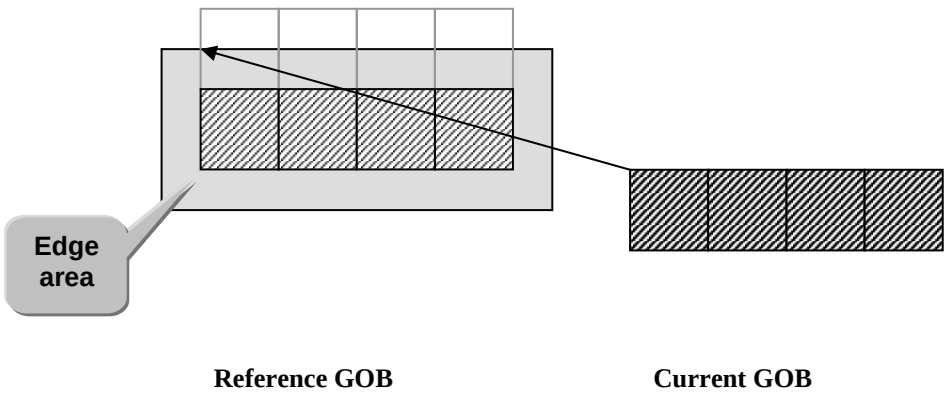


Fig. 7.6. A motion vector might point to the edge area of one GOB, or to the central area of another GOB

7.4 Experimental Results

Considering its complex motion nature with zoom-in-zoom-out motions and scene changes, we choose *foreman* in QCIF 4:2:0 YUV format in length of 400 frames as our test video sequence. For simplicity, we exploit the PSNR as the distortion metric for each decoded frame.

First we encode *foreman* at 56 kbps, 10 fps with our two-layer rate-distortion encoding method, with the results given in Fig. 7.7. Error resilience is introduced to the bitstream by our encoder in which a GOB only depends on at most one GOB in the reference picture, which results in 1.5 dB loss in PSNR. We observe that the encoder is much likely to choose the same segment in the reference picture based on the two-layer rate-distortion optimization. Only 11 frames out of the total 130 encoded P-frames include segments referring to the area other than the same segments in the reference picture. Those frames are around the 80th encoded P-frame (240th frame in the original video sequence) for *foreman* where scene changes occur. Therefore, we can just take advantage of Annex R in H.263+, which demands the same segmentation between two I-frames to replace the second layer rate-distortion optimization in our scheme. Notice that for each reference GOB, we treat it in the same way as with the reference picture, including extrapolating the edge area and realizing OBMC. The decoded video quality in PSNR of the simplified scheme is 30.12 dB in PSNR with 0.05 dB loss in average compared to the two-layer scheme.

Next we use our schemes with error-prone packet networks. We have discussed in Chapter 6 that INTRA refresh rate is a critical parameter for the error resilience concern. We map this parameter to the number of GOBs contained in each packet in our simplified scheme. In the packetization stage, we place three GOBs in one packet, which always starts with an INTRA mode GOB. Therefore, every three frames are fragmented into nine packets, altogether containing nine INTRA GOBs, which is equivalent to set the INTRA refresh rate to be 1/3 frame here, i.e., in average 1/3 of the macroblocks in each frame are forced to be INTRA. Without using the Reed-Solomon coding, the received video quality is shown in Fig. 7.8, after de-packetization and decoding. The solid curve denotes the distortion when error free, while the dashed one denotes the distortion with packet loss at rate 5%. By using Reed-Solomon coding, we can keep the packetized bitstream almost intact in a lower packet loss rate and achieving a similar performance in a higher packet loss rate as in the lower rate case without employing error correction coding.

Notice that for some frames the PSNR drops more than 5 dB each, as for the 15th to 21st encoded P frames shown in Fig. 7.8. This is because one packet loss means three consecutive frames suffer in the same location, and our current error concealment method simply copies the same located macroblock in the previous frame if the current macroblock data are not available. We can use a better error concealment method for future work to improve the performance. For example, we can choose the macroblock in the previous picture with the most matched neighboring area as the concealed image for the current lost macroblock [220].

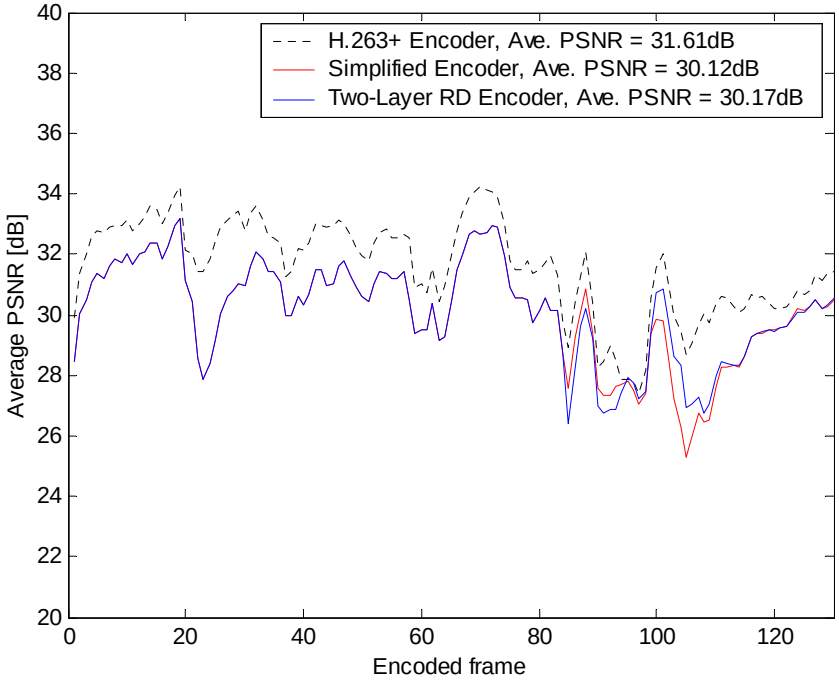


Fig. 7.7. Source encoding for *foreman*

As shown in Fig. 7.9, the packet that contains the second GOBs in three consecutive pictures have been lost. Since we have prevent any dependency across the packet boundaries, motion estimation of one GOB in a frame is completely independent of all the other GOBs in the same picture. Therefore, the decoder does not have any information to be referenced to regarding the motion information of the lost GOB, and thus the conventional temporal-replacement error concealment method [200,201] cannot be of any help. At this time, we might consider the trade-off of dependency and independency across segments when designing the data partitioning strategies.

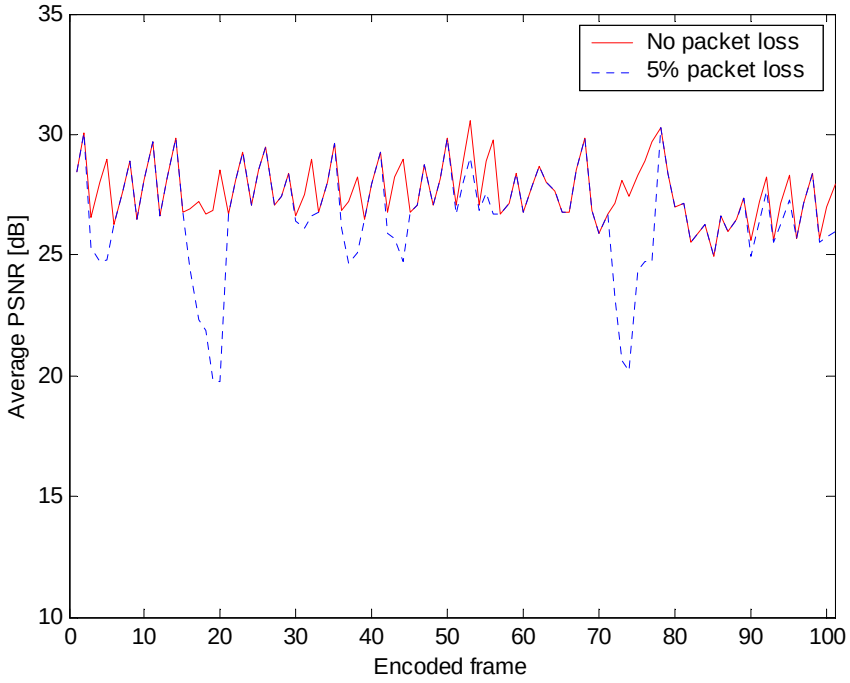
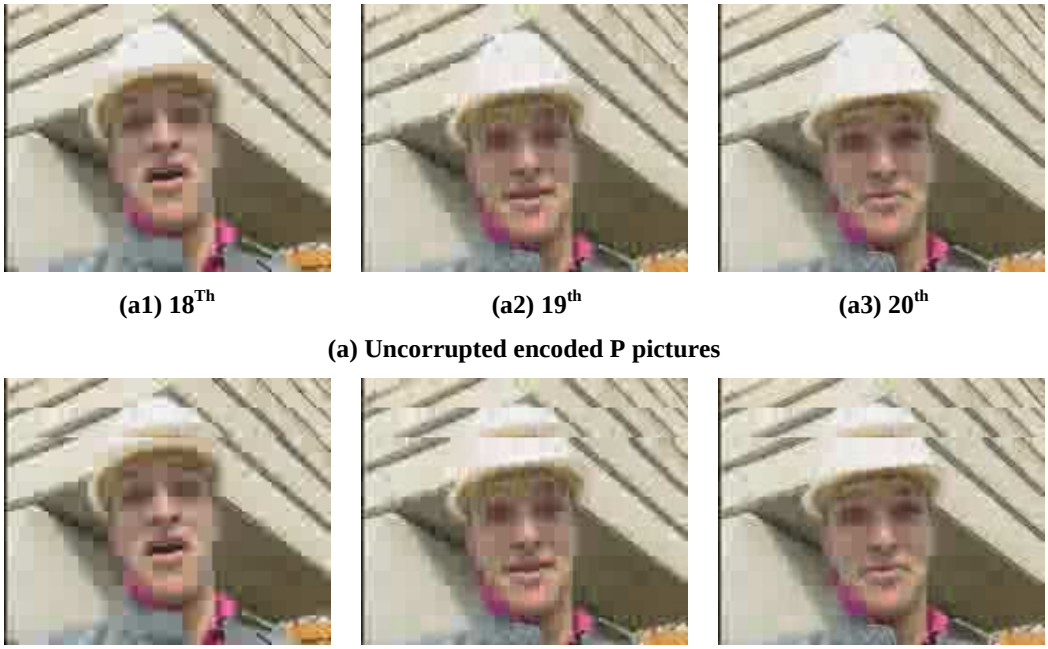


Fig. 7.8. Transmitted over 5% packet loss network (*foreman*)



(a1) 18Th **(a2) 19th** **(a3) 20th**

(a) Uncorrupted encoded P pictures

(b1) 18Th **(b2) 19th** **(b3) 20th**

(b) Corrupted encoded P pictures

Fig. 7.9. Packet loss recovery by simple error concealment to *foreman*

7.5 Conclusions

In this chapter, we describe the following contributions:

- We have proposed new schemes to introduce error resilience into the compressed video bitstreams for transmission over packet networks. First, we developed an adaptive packetization scheme that prohibits any dependency across packets, for error resilience purposes, while exploiting the dependency within each packet to improve the source coding performance. Secondly, we addressed a two-layer rate-distortion optimization scheme to serve our packetization method. Finally, we presented a simplified version of our schemes to make it fully compliant with H.263+.

In our future work, we will explore appropriate error concealment methods to better serve our error resilience schemes.

8. CONCLUSIONS

As we have discussed in Chapter 1, transmission of digital video signals over current data networks demands efficient, reliable, and adaptable video coding techniques due to the heterogeneous nature of current wired and wireless networks. In this dissertation, we have mainly addressed the scalable video coding structure, in particular the leaky prediction layered video coding (LPLC) and low complexity video encoding techniques.

Leaky prediction layered video coding, known as LPLC, includes a scaled version of the enhancement layer within the motion compensation loop to improve the coding efficiency while maintaining graceful recovery in the presence of error drift. We have completed the following work regarding LPLC:

Rate distortion performance of LPLC:

- We have examined a deficiency inherent in the LPLC structure, namely that the reconstructed video quality from both the enhancement layer and the base layer cannot be guaranteed to be always superior to that of using the base layer alone, even when no drift occurs. In other words, the enhancement layer does not always "enhance" the performance. We highlighted this deficiency using a formulation that describes LPLC.

- We have proposed a general framework that applies to both LPLC and a multiple description coding scheme using motion compensation, known as MDMC. We have addressed two types of similarities, namely Similarity I and Similarity II, that exist between LPLC and MDMC. We refer to the similarity between the enhancement layer in LPLC and the central loop in MDMC, considered from the leaky prediction point of view, as Similarity I. Similarity I conforms with the well-accepted supposition in the literature that the reconstructed quality using both layers in LPLC shall be superior to that using the base layer alone. We refer to the similarity between the enhancement layer in LPLC and the side loops in MDMC, considered from the *mismatch* point of view, as Similarity II. The mismatch in LPLC is the difference between the enhancement layer predicted error frame (PEF) and the reconstructed version of the base layer PEF. Similarity II between MDMC and LPLC confirms the deficiency that might exist in LPLC. The seemingly disagreement between the above two types of similarities is consistent with our analysis that the superiority of the enhancement layer in LPLC is dependent on the leaky factor as well as the accuracy of the encoded enhancement layer itself.

- We have proposed an enhanced LPLC based on maximum-likelihood (ML) estimation, termed ML-LPLC, to address the previously specified deficiency in LPLC. ML-LPLC is capable of addressing the deficiency in LPLC in terms of the coding efficiency regardless of the characteristics of the encoded videos, the data rates allocated to either of the two layers, or the choices of the leaky fac-

tors. The implementation of ML-LPLC requires reasonable extra computation. Once the ML-coefficients are derived, to obtain the ML reconstruction at the decoder only requires two scaling and one addition for each pixel. Moreover, the transmission of the side information for ML-LPLC, i.e., the transmission of the ML coefficients requires negligible additional data rate compared to the payload data rates.

- To theoretically analyze the rate distortion performance of LPLC, we have developed an alternative block diagram of LPLC. Similar to the original LPLC framework, the alternative block diagram includes two motion compensated prediction (MCP) loops, where the base layer PEF is encoded in the base layer MCP step and the mismatch is encoded in the enhancement layer MCP step. Different from the original framework, the leaky factor is only present in the enhancement layer MCP step in the alternative block diagram, which significantly simplifies the theoretic analysis. We have addressed the theoretic analysis of LPLC using two different approaches, namely the one using rate distortion theory and the one using a quantization noise model, based on the alternative block diagram.

- In the first approach for theoretically analyzing LPLC, we have used the optimum forward channel derived from rate distortion theory to model the encoding of a 2D image and obtain the parametric rate distortion functions for LPLC in closed form for one scenario where the enhancement layer is intact

and the other where it has drift. For each scenario, the closed form rate distortion functions are in relation to three parameters: the power spectral density (PSD) of the input video frame, the probability distribution of the motion vector estimation errors, as well as the leaky factor.

- In the second approach for theoretically analyzing LPLC, we have addressed the rate distortion performance of LPLC by using quantization noise modeling. Since the optimum forward channel used in the first approach is derived from rate distortion theory, it provides the rate distortion bound at which a 2D stationary, Gaussian random signal is encoded. The quantization noise model in the second approach is a heuristic model, which was obtained from the operational results. The optimum forward channel specifies parametric rate distortion functions, whereas the use of the quantization noise model provides a closed formulation of the mean-square-error (MSE) distortion that is explicitly related to the data rate. We also obtain closed form rate distortion functions for the two scenarios of LPLC.

- We have evaluated both closed form expressions and demonstrated that the leaky factor is critical in the performance of coding efficiency. We have validated that with the partial or full inclusion of the enhancement layer in the MCP loop, LPLC does improve the coding efficiency as opposed to the conventional layered scalable coding. When the enhancement layer has no error drift, it is shown that at a specific leaky factor value between 0 and 1, the

decoded video quality increases with the increase of the data rate. When the data rate is sufficiently large, LPLC achieves a better rate distortion performance with increasing leaky factor. It is interesting to note that when the enhancement layer data rate is small, it might be possible that a larger leaky factor yields a less efficient codec, especially when the leaky factor is close to 1. We have also shown that the leaky factor is critical in error resilience performance when the enhancement layer in LPLC suffers from error drift. When drift occurs in the enhancement layer, it is observed that larger leaky factors yield a larger drop in the rate distortion performance, especially when the leaky factor approaches 1. We have simulated both scenarios, with and without drift in the enhancement layer, using SAMCoW and evaluated the operational rate distortion performance of LPLC associated with various leaky factors for video sequences containing varying degrees of motions. It is shown that the theoretical results conform with the operational results.

We have discussed that two types of scalabilities exist in scalable video coding: (i) nested scalability, in which different representations of each frame are generated using layered scalable coding and have to be decoded in a fixed sequential order, an example of which is LPLC; (ii) parallel scalability, which is used in multiple description coding (MDC) where different descriptions are mutually refinable and independently decodable. We have completed the following work regarding the parallel scalable video coding:

Nested scalability in the parallel scalable coding structure:

- We have used the framework that applies to both LPLC and MDMC to introduce the nested scalability into each description of the MDC stream. We have proposed a fine granularity scalability (FGS) based MDC approach, termed fine-scalable-MDC, FS-MDC. The essential framework of FS-MDC is identical to that of MDMC, however, FS-MDC views the functionality of the mismatch transmitted in the side loops as the enhancement information, instead of as compensation to the central loop in MDMC. From the overall descriptions point of view, the MD structure in the base layer has error resilient capability since any single description of the base layer is decodable.

- We have proposed a coding structure characterized as "dual-leaky," and referred to it as Dual-LPLC. The "dual-leaky" feature is defined in the sense that one leaky prediction is exploited in the parallel scalable coded base layer and a second leaky prediction is included in the nested scalable coded enhancement layer. The leaky prediction in the base layer is manifested by the second-order prediction that originated from the MD structure, while the leaky prediction in the enhancement layer is implemented by incorporating a scaled version of the enhancement layer in the motion compensation loop of each single description. Dual-LPLC combines nested scalability and parallel scalability under one framework and maintains a good balance between scalable video coding and error resilient video coding due to its "dual-leaky" prediction feature.

We have also addressed the new emerging low complexity video encoding technique, which is developed for applications such as wireless sensor networks and distributed video surveillance systems where resources for memory, computation, and energy are scarce at the video encoder side whereas resources at the decoder are relatively abundant. We have introduced the Slepian-Wolf and Wyner-Ziv theorems which provided a theoretic basis for this new coding paradigm. We have completed the following work regarding low complexity video encoding:

Low complexity video encoding using B-frame direct modes:

- We have proposed a low complexity video encoding approach using B-frame direct modes. The direct mode was originally developed for encoding B frames. The motion compensated prediction of the direct mode is a linear combination of the two predictions obtained from the forward motion compensation and the backward motion compensation. In our proposed approach, we have extended the direct-mode idea and designed new B-frame direct modes to encode B frames. We constrained any macroblock in a B frame to be encoded only using either the intra-coding mode or one of the new direct coding modes. Hence, no motion estimation is used to any B frames and low complexity video encoding is achieved. Our proposed low complexity video encoding requires a feedback channel from the decoder to the encoder. We implemented motion estimation at the decoder and transmitted the motion vectors back to the decoder. We have designed three B-frame direct modes and specified nine coding modes for B frame macroblocks: PPForw_Bidir, PPForw_Forw, PPForw_Back,

PPBack_Bidir, PPBack_Forw, PPBack_Back, PBBidir_Bidir, PBBidir_Forw, and PBBidir_Back. Experimental results have shown that our approach using new B-frame direct modes with help of a feedback channel obtains a competitive rate distortion performance compared to that of the high complexity video encoding approach.

We have also completed the following work regrading reliable video transmission over error prone networks:

Reliable transmission of digital videos:

- We have presented a thorough evaluation of the joint source and channel video coding methodology from two points of view: source coding design for error resilience and channel coding for error detection and recovery. We investigated the current ITU-T video compression standard, H.263+, for 3G wireless transmission. In particular, we concentrated on error resilient features provided within the standard and forward error correction (FEC) to find the optimal combination of various system parameters under different lossy channel conditions. Furthermore, we investigated new metrics other than the common peak signal-to-noise ratio (PSNR) to evaluate video distortion caused by the combination of source compression and channel errors.

- We have proposed new schemes to introduce error resilience into the compressed video bitstreams for transmission over packet networks. First, we developed an adaptive packetization scheme that prohibits any dependency

across packets, for error resilience purposes, while exploiting the dependency within each packet to improve the source coding performance. Secondly, we addressed a two-layer rate-distortion optimization scheme to serve our packetization method. Finally, we presented a simplified version of our schemes to make it fully compliant with H.263+.

The future work can be explored from the following two perspectives:

Future research:

- For LPLC, we have observed that the leaky factor is critical and has three functionalities: (1) It affects the coding efficiency; (2) It affects the error resilience performance; (3) It determines the superiority of the reconstruction by both layers. Hence, which value should be chosen for the leaky factor in LPLC is closely related to the application. Adaptively adjusting the leaky factor thus becomes a good alternative. We would be exploring the mode-adaptive ML-LPLC approach facilitated with the drift-managing mechanism in our future work.

- For low complexity video encoding, distributed video coding based on the Wyner-Ziv structure has been actively revisited recently, such as the one using Turbo codes. Our scheme using B-frame direct modes can be further coupled with Wyner-Ziv distributed encoding to provide a practical and efficient low complexity video encoding system.

- We will explore error resilient low complexity video encoding using B-frame direct modes. When video bitstreams suffer from channel errors, errors that occur to the motion vectors of the co-located macroblock, either in the forward channel or in the feedback channel, will inevitably propagate to the macroblock that is coded in direct mode. In our future work, we will consider interpolation/extrapolation of motion vectors with errors for direct mode coded B-frames. UEP may be used to protect the motion vectors. Since motion estimation is implemented at the decoder in our low complexity video encoding approach, we would also pursue appropriate motion compensated error concealment techniques.

LIST OF REFERENCES

LIST OF REFERENCES

[1] Yao Wang, Jorn Ostermann, and Ya-Qin Zhang. *Video Processing and Communications*. Prentice Hall, 2002. ISBN: 0-13-017547-1.

[2] Hong Man, R. L. de Queiroz, and M. J. T. Smith. Three-dimensional subband coding techniques for wireless video communications. *IEEE Transactions on Circuits and Systems for Video Technology*, 12:386–397, June 2002.

[3] B. G. Haskell, A. Puri, and A. N. Netravali. *Digital Video: An Introduction to MPEG-2*. Chapman and Hall, 1997.

[4] W. Li. Overview of fine granularity scalability in MPEG-4 video standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 11:301–317, March 2001.

[5] K. Shen and E. J. Delp. Wavelet based rate scalable video compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 9:109–122, February 1999.

[6] Mihaela van der Schaar and Hayder Radha. Unequal packet loss resilience for fine-granular-scalability video. *IEEE Transactions on Multimedia*, 2:381–394, December 2001.

[7] Wai tian Tan and Avideh Zakhor. Real-time Internet video using error resilient scalable compression and TCP-friendly transport protocol. *IEEE Transactions on Multimedia*, 1:172–186, June 1999.

[8] K. Shen. *A Study of Real Time and Rate Scalable Image and Video Compression*. Ph.D. Thesis, School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, December 1997.

[9] E. J. Delp, P. Salama, E. Asbun, M. Saenz, and K. Shen. Rate scalable image and video compression techniques. In *Proceedings of the 42nd Midwest Symposium on Circuits and Systems*, pages 635–638, Las Cruces, New Mexico, August 8-11 1999.

[10] E. Asbun, P. Salama, K. Shen, and E. J. Delp. Very low bit rate wavelet-based scalable video compression. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 948–952, Chicago, Illinois, October 4-7 1998.

[11] E. Asbun, P. Salama, and E. J. Delp. Encoding of predictive error frames in rate scalable video codecs using wavelet shrinkage. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, volume 3, pages 832–836, Kobe, Japan, October 24-28 1999.

[12] E. Asbun, P. Salama, and E. J. Delp. Preprocessing and postprocessing techniques for encoding predictive error frames in rate scalable video codecs. In *Proceedings of the International Workshop on Very Low Bitrate Video Coding*, pages 148–151, Kobe, Japan, October 29-30 1999.

[13] E. Asbun, P. Salama, and E. J. Delp. A rate-distortion approach to wavelet-based encoding of predictive error frames. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, volume 3, pages 154–157, Vancouver, British Columbia, September 10-13 2000.

[14] E. Asbun. *Improvements in Wavelet-Based Rate Scalable Video Compression*. Ph.D. Thesis, School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, December 2000.

[15] K. Shen and E. J. Delp. A control scheme for a data rate scalable video codec. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, volume 2, pages 69–72, Lausanne, Switzerland, September 16-19 1996.

[16] M. L. Comer, K. Shen, and E. J. Delp. Rate-scalable video coding using a zerotree wavelet approach. In *Proceedings of the Ninth Image and Multidimensional Digital Signal Processing Workshop*, volume 3, pages 162–163, Belize City, Belize, March 3-6 1996.

[17] K. Shen and E. J. Delp. Color image compression using an embedded rate scalable approach. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, volume 3, pages 34–37, Santa Barbara, California, October 26-29 1997.

[18] M. Saenz, P. Salama, K. Shen, and E. J. Delp. An evaluation of color embedded wavelet image compression techniques. In *Proceedings of the SPIE International Conference on Visual Communications and Image Processing (VCIP)*, volume 3653, pages 282–293, San Jose, California, January 23-29 1999.

[19] V. K. Goyal. Multiple description coding: Compression meets the network. *IEEE Signal Processing Magazine*, 18:74–93, September 2001.

[20] H.C. Huang, C.N. Wang, and T. Chiang. A robust fine granularity scalability using trellis-based predictive leak. *IEEE Transactions on Circuits and Systems for Video Technology*, 12:372–385, June 2002.

[21] S. Han and B. Girod. Robust and efficient scalable video coding with leaky prediction. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, volume 2, pages 41–44, Rochester, NY, September 22-25 2002.

[22] W.-H. Peng and Y.-K. Chen. Error drifting reduction in enhanced fine granularity scalability. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, volume 2, pages 61–64, Rochester, NY, September 22-25 2002.

[23] F. Wu, S. Li, and Y.-Q. Zhang. A framework for efficient fine granularity scalable video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 11:332–344, March 2001.

[24] Wen-Shiaw Peng and Yen-Kuang Chen. Mode-adaptive fine granularity scalability. In *Proceedings of the IEEE International Conference on Image Processing*, volume 2, pages 993–996, Thessaloniki, Greece, October 7-10 2001.

[25] A. R. Reibman, L. Bottou, and A. Basso. Scalable video coding with managed drift. *IEEE Transactions on Circuits and Systems for Video Technology*, 13:131–140, February 2003.

[26] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 623–656, July, October 1948.

[27] A. Ortega and K. Ramchandran. Rate-distortion methods for image and video compression. *IEEE Signal Processing Magazine*, pages 23–50, November 1998.

[28] Gary J. Sullivan and Thomas Wiegand. Rate-distortion optimization for video compression. *IEEE Signal processing Magazine*, pages 74–90, November 1998.

[29] K. Ramchandran and M. Vetterli. Best wavelet packet bases in a rate-distortion sense. *IEEE Transactions on Image Processing*, 2:160–175, April 1993.

[30] Y. Wang and Q.F. Zhu. Error control and concealment for video communication: A review. *Proceedings of the IEEE*, 86:974–997, May 1998.

[31] Xin Li and M. T. Orchard. Novel sequential error-concealment techniques using orientation adaptive interpolation. *IEEE Transactions on Circuits and Systems for Video Technology*, 12:857–864, October 2002.

[32] Paul Salama. *Error concealment in encoded images and video*. Ph.D. Thesis, School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, August 1999.

[33] Paul Salama, Ness Shroff, and Edward Delp. Error concealment in encoded video. *IEEE Journal on Selected Areas in Communications*, 18:1129–1114, June 2000.

[34] Fan Zhai. *Optimal cross-layer resource allocation for real-time video transmission over packet lossy networks*. Ph.D. Thesis, Northwestern University, Evanston, IL, June 2004. [Online]. Available: http://www.ece.northwestern.edu/∼fzhai/publications/Fan_thesis.pdf.

[35] R. Schafer and T. Sikora. Digital video coding standards and their role in video communications. *Proceedings of the IEEE*, 83(6):907–924, June 1995.

[36] A. Luthra, G. J. Sullivan, and T. Wiegand. Introduction to the special issue on the H.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13:557–559, July 2003.

[37] T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra. Overview of the H.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13:560–576, July 2003.

[38] H.264/AVC software coordination, 2004. [Online]. Available: http://bs.hhi.de/∼suehring/tml/.

[39] D. Marpe, H. Schwarz, and T. Wiegand. Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13:620–636, July 2003.

[40] M. Karczewicz and R. Kurceren. The SP- and SI-frames design for H.264/AVC. *IEEE Transactions on Circuits and Systems for Video Technology*, 13:637–644, July 2003.

[41] S. Wenger. H.264/AVC over IP. *IEEE Transactions on Circuits and Systems for Video Technology*, 13:645–656, July 2003.

[42] T. Stockhammer, M. M. Hannuksela, and T. Wiegand. H.264/AVC in wireless environments. *IEEE Transactions on Circuits and Systems for Video Technology*, 13:657–673, July 2003.

[43] P. List, A. Joch, J. Lainema, G. Bjntegaard, and M. Karczewicz. Adaptive deblocking filter. *IEEE Transactions on Circuits and Systems for Video Technology*, 13:614–619, July 2003.

[44] T. Wedi and H. G. Musmann. Motion- and aliasing-compensated prediction for hybrid video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 13:577–586, July 2003.

[45] M. Flierl and B. Girod. Generalized B pictures and the draft H.264/AVC video-compression standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13:587–597, July 2003.

[46] H. S. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky. Low-complexity transform and quantization in H.264/AVC. *IEEE Transactions on Circuits and Systems for Video Technology*, 13:598–603, July 2003.

[47] J. Ribas-Corbera, P. A. Chou, and S. L. Regunathan. A generalized hypothetical reference decoder for H.264/AVC. *IEEE Transactions on Circuits and Systems for Video Technology*, 13:674–687, July 2003.

[48] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan. Rate-constrained coder control and comparison of video coding standards. *IEEE Transactions on Circuits and Systems for Video Technology*, 13:688–703, July 2003.

[49] M. Horowitz, A. Joch, F. Kossentini, and A. Hallapuro. H.264/AVC baseline profile decoder complexity analysis. *IEEE Transactions on Circuits and Systems for Video Technology*, 13:704–716, July 2003.

[50] V. Lappalainen, A. Hallapuro, and T. D. Hamalainen. Complexity of optimized H.26L video decoder implementation. *IEEE Transactions on Circuits and Systems for Video Technology*, 13:717–725, July 2003.

[51] Y. Liu, Z. Li, P. Salama, and E. J. Delp. A discussion of leaky prediction based scalable coding. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, volume 2, pages 565–568, Baltimore, MD, July 6-9 2003.

[52] Y. Liu, P. Salama, Z. Li, and E. J. Delp. An enhancement of leaky prediction layered video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, submitted for publication.

[53] Y. Wang and S. Lin. Error-resilient video coding using multiple description motion compensation. *IEEE Transactions on Circuits and Systems for Video Technology*, 12:438–452, June 2002.

[54] T. Wiegand. H.26L test model long-term number 9 (TML-9). ITU-T Q.6/SG 16, VCEG-N83d1, December 2001.

[55] S. Wenger. Video redundancy coding in H.263+, September 15-16 1997.

[56] ITU-T Recommendation H.263: Video coding for low bit rate communication, February 1998.

[57] X. Tang and A. Zakhor. Matching pursuits multiple description coding for wireless video. *IEEE Transactions on Circuits and Systems for Video Technology*, 12:566–575, June 2002.

[58] G. W. Cook. *A study of scalability in video compression: Rate-distortion analysis and parallel implementation*. Ph.D. Thesis, School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, December 2002.

[59] J. Prades-Nebot and G. W. Cook. Analysis of the performance of predictive SNR scalable coders. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, volume 3, pages 861–864, Barcelona, Spain, September 14-17 2003.

[60] J. Prades-Nebot, G. W. Cook, and E. J. Delp. Analysis of the efficiency of SNR-scalable strategies for motion compensated video coders. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, Singapore, October 24-27 2004.

[61] Y. Liu, P. Salama, G. W. Cook, and E. J. Delp. Rate-distortion analysis of layered video coding by leaky prediction. In *Proceedings of the SPIE International Conference on Video Communications and Image Processing (VCIP)*, volume 5308, pages 543–554, San Jose, CA, January 18-22 2004.

[62] Y. Liu, J. Prades-Nebot, P. Salama, and E. J. Delp. Performance analysis of leaky prediction layered video coding using quantization noise modeling. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, Singapore, October 24-27 2004.

[63] T. Berger. *Rate distortion theory: A mathematical basis for data compression*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1971.

[64] B. Girod. The efficiency of motion-compensation prediction for hybrid coding of video sequences. *IEEE Journal on Selected Areas in Communications*, 5:1140–1154, August 1987.

[65] G. W. Cook, J. Prades-Nebot, Y. Liu, and E. J. Delp. Rate-distortion analysis of motion compensated rate scalable video. *IEEE Transactions on Image Processing*, submitted for publication.

[66] G. W. Cook, J. Prades-Nebot, and E. J. Delp. Rate-distortion bounds for motion compensated rate scalable video coders. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, Singapore, October 24-27 2004.

[67] B. Girod. Motion-compensating prediction with fractional-pel accuracy. *IEEE Transactions on Communications*, 41(4):604–612, April 1993.

[68] M. Flierl, T. Wiegand, and B. Girod. Rate-constrained multihypothesis prediction for motion-compensated video compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(11):957–969, November 2002.

[69] M. Flierl and B. Girod. Video coding with motion compensation for groups of pictures. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, volume 1, pages 69–72, Rochester, NY, September 22-25 2002.

[70] M. Flierl and B. Girod. Multihypothesis motion estimation for video coding. In *Proceedings of the IEEE Data Compression Conference (DCC)*, pages 341–350, Snowbird, Utah, March 27-29 2001.

[71] M. S. Pinsker. *Information and information stability of random variables and processes*. Holden-Day, San Francisco, 1964.

[72] P.-Y. Cheng, J. Li, and C.-C. J. Kuo. Rate control for an embedded wavelet video coder. *IEEE Transactions on Circuits and Systems for Video Technology*, 7:696–702, August 1997.

[73] Z. Li and E. J. Delp. Statistical motion prediction with drift. In *Proceedings of the SPIE International Conference on Video Communications and Image Processing (VCIP)*, volume 5308, pages 416–427, San Jose, CA, January 18-22 2004.

[74] Z. Li and E. J. Delp. Channel-aware rate-distortion optimized leaky motion prediction. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, Singapore, October 24-27 2004.

[75] Y. Liu, P. Salama, and E. J. Delp. Multiple description scalable coding for error resilient video transmission over packet networks. In *Proceedings of the SPIE International Conference on Image and Video Communications and Processing (IVCP)*, volume 5022, pages 157–168, Santa Clara, CA, January 20-24 2003.

[76] V. A. Vaishampayan. Design of multiple description scalar quantizers. *IEEE Transactions on Information Theory*, 39:821–834, May 1993.

[77] S.D. Servetto, K. Ramchandran, V. A. Vaishampayan, and K. Nahrstedt. Multiple description wavelet based image coding. *IEEE Transactions on Image Processing*, 9:813–826, May 2000.

[78] Y. Wang, M. T. Orchard, V. Vaishampayan, and A. R. Reibman. Multiple description coding using pairwise correlating transforms. *IEEE Transactions on Image Processing*, 10:351–365, March 2001.

[79] W. Jiang and A. Ortega. Multiple description coding via polyphase transform and selective quantization. In *Proceedings of the SPIE International Conference on Visual Communications and Image Processing (VCIP)*, volume 3653, pages 998–1008, San Hose, CA, January 1999.

[80] A. R. Reibman, H. Jafarkhani, Y. Wang, M. T. Orchard, and R. Puri. Multiple-description video coding using motion-compensated temporal prediction. *IEEE Transactions on Circuits and Systems for Video Technology*, 12:193–204, March 2002.

[81] A. R. Reibman, H. Jafarkhani, Y. Wang, M. T. Orchard, and R. Puri. Multiple description coding for video using motion compensated prediction. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, volume 3, pages 837–841, Kobe, Japan, October 24-28 1999.

[82] A. Sehgal, A. Jagmohan, and N. Ahuja. Wireless video conferencing using multiple description coding. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, volume 5, pages 303–306, Sydney, Australia, May 6-9 2001.

[83] C.-S. Kim and S.-U. Lee. Multiple description coding of motion fields for robust video transmission. *IEEE Transactions on Circuits and Systems for Video Technology*, 11:999–1010, September 2001.

[84] S. L. Regunathan and K. Rose. Efficient prediction in multiple description video coding. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, volume 1, pages 1020–1023, Vancouver, BC, Canada, September 10-13 2000.

[85] R. Puri and K. Ramchandran. Multiple description source coding through forward error correction codes. In *Proceedings of the Thirty-Third Asilomar Conference on Signals, System, and Computers*, volume 1, pages 342–346, Pacific Grove, CA, October 24-27 1999.

[86] R. Puri, K. Ramchandran, K. W. Lee, and V. Bharghavan. Application of FEC based multiple description coding to Internet video streaming and multicast. In *Proceedings of the International Packet Video Workshop*, Sardinia, Italy, May 2000.

[87] A. Reibman. Optimizing multiple description video coders in a packet loss environment. In *Proceedings of the International Packet Video Workshop*, Pittsburgh, PA, April 2002.

[88] R. Singh and A. Ortega. Erasure recovery in predictive coding environments using multiple description coding. In *Proceedings of the International Workshop on Multimedia Signal Processing*, Denmark, September 1999.

[89] J. Apostolopoulos, T. Wong, W.-T. Tan, and S. Wee. On multiple description streaming with content delivery networks. In *Proceedings of the IEEE INFOCOM*, volume 3, pages 1736–1745, New York, NY, June 23-27 2002.

[90] J. Apostolopoulos, W.-T. Tan, S. Wee, and G. W. Wornell. Modeling path diversity for multiple description video communication. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, volume 2, pages 189–192, Rochester, New York, September 22-25 2002.

[91] J. Apostolopoulos, W.-T. Tan, S. Wee, and G. W. Wornell. Modeling path diversity for multiple description video communication. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 3, pages 2161–2164, Orlando, Florida, May 13-17 2002.

[92] J. G. Apostolopoulos and S. J. Wee. Unbalanced multiple description video communication using path diversity. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, volume 1, pages 966–969, Thessaloniki, Greece, October 7-10 2001.

[93] V. N. Padmanabhan, H. J. Wang, P. A. Chou, and K. Sripanidkulchai. Distributing streaming media content using cooperative networking. Microsoft Technical Report MSR-TR-2002-37, April 2002. [Online]. Available: http://research.microsoft.com/∼pachou/publications.htm.

[94] M. Orchard, Y. Wang, V. Vaishampayan, and A. Reibman. Redundancy rate-distortion analysis of multiple description coding using pairwise correlating transforms. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, volume 1, pages 608–611, Santa Barbara, CA, October 26-29 1997.

[95] A. R. Reibman, Y. Wang, X. Qiu, Z. Jiang, and K. Chawla. Transmission of multiple description and layered video over an egprs wireless network. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, volume 2, pages 136–139, Vancouver, BC, Canada, September 10-13 2000.

[96] R. Singh, A. Ortega, L. Perret, and W. Jiang. Comparison of multiple description coding and layered coding based on network simulations. In *Proceedings of the SPIE International Conference on Image and Video Communications and Processing (IVCP)*, volume 3974, pages 929–939, San Jose, CA, January 2000.

[97] P. Sagetong and A. Ortega. Optimal bit allocation for channel-adaptive multiple description coding. In *Proceedings of the SPIE International Conference on Image and Video Communications and Processing (IVCP)*, volume 3974, pages 53–63, San Jose, CA, January 2000.

[98] A. R. Reibman, H. Jafarkhani, M. T. Orchard, and Y. Wang. Performance of multiple description coders on a real channel. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 5, pages 2415–2418, Phoenix, AZ, May 15-19 1999.

[99] H. Wang and A. Ortega. Robust video communication by combining scalability and multiple description coding techniques. In *Proceedings of the SPIE International Conference on Image and Video Communications and Processing (IVCP)*, volume 5022, pages 111–124, Santa Clara, CA, January 20-24 2003.

[100] Y. Wang, A. R. Reibman, M. T. Orchard, and H. Jafarkhani. An improvement to multiple description transform coding. *IEEE Transactions on Signal Processing*, 50:2843–2854, November 2002.

[101] P. A. Chou, H. J. Wang, and V. N. Padmanabhan. Layered multiple description coding. In *Proceedings of the International Packet Video Workshop*, Nantes, France, April 28-29 2003.

[102] J. D. Slepian and J. K. Wolf. Noiseless coding of correlated information sources. *IEEE Transactions on Information Theory*, 19(4):471–480, July 1973.

[103] A. D. Wyner. Recent results in the shannon theory. *IEEE Transactions on Information Theory*, 20(1):2–10, January 1974.

[104] S. S. Pradhan and K. Ramchandran. Distributed source coding using syndromes (DISCUS): design and construction. *IEEE Transactions on Information Theory*, 49(3):626–643, March 2003.

[105] S. S. Pradhan and K. Ramchandran. Distributed source coding using syndromes (DISCUS): design and construction. In *Proceedings of the IEEE Data Compression Conference (DCC)*, pages 158–167, Snowbird, Utah, March 29-31 1999.

[106] G. D. Forney Jr. Coset codes. I. Introduction and geometrical classification. *IEEE Transactions on Information Theory*, 34(5):1123–1151, September 1988.

[107] G. D. Forney Jr. Coset codes. II. Binary lattices and related codes. *IEEE Transactions on Information Theory*, 34(5):1152–1187, September 1988.

[108] S. S. Pradhan and K. Ramchandran. Generalized coset codes for symmetric distributed source coding. *IEEE Transactions on Information Theory*, February 2003. submitted for publication. [Online]. Available: http://www.eecs.umich.edu/∼pradhanv/paper/ittrans03_3.ps.

[109] S. S. Pradhan and K. Ramchandran. Group-theoretic construction and analysis of generalized coset codes for symmetric/asymmetric distributed source coding. In *Proceedings of the Conference on Information Sciences and Systems (CISS)*, Princeton, NJ, March 2000.

[110] D. Schonberg, K. Ramchandran, and S. S. Pradhan. Distributed code constructions for the entire slepian-wolf rate region for arbitrarily correlated sources. In *Proceedings of the IEEE Data Compression Conference (DCC)*, pages 292–301, Snowbird, Utah, March 23-25 2004.

[111] S. S. Pradhan, J. Kusuma, and K. Ramchandran. Distributed compression in a dense microsensor network. *IEEE Signal Processing Magazine*, 19(2):51–60, March 2002.

[112] S. S. Pradhan and K. Ramchandran. Distributed source coding: symmetric rates and applications to sensor networks. In *Proceedings of the IEEE Data Compression Conference (DCC)*, pages 363–372, Snowbird, Utah, March 28-30 2000.

[113] X. Wang and M. T. Orchard. Design of trellis codes for source coding with side information at the decoder. In *Proceedings of the IEEE Data Compression Conference (DCC)*, pages 361–370, Snowbird, Utah, March 27-29 2001.

[114] A. D. Liveris, Z. Xiong, and C. N. Georghiades. Compression of binary sources with side information at the decoder using LDPC codes. *IEEE Communications Letters*, 6(10):440–442, October 2002.

[115] C.-F. Lan, A. D. Liveris, K. Narayanan, Z. Xiong, and C. N. Georghiades. Slepian-Wolf coding of multiple M-ary sources using LDPC codes. In *Proceedings of the IEEE Data Compression Conference (DCC)*, pages 549–549, Snowbird, UT, March 23-25 2004.

[116] A. D. Liveris, Z. Xiong, and C. N. Georghiades. Joint source-channel coding of binary sources with side information at the decoder using IRA codes. In *Proceedings of the IEEE Workshop on Multimedia Signal Processing*, pages 53–56, St. Thomas, US Virginia Islands, December 9-11 2002.

[117] A. D. Liveris, Z. Xiong, and C. N. Georghiades. A distributed source coding technique for correlated images using Turbo codes. *IEEE Communications Letters*, 6(9):379–381, September 2002.

[118] V. Stankovic, A. D. Liveris, Z. Xiong, and C. N. Georghiades. Design of Slepian-Wolf codes by channel code partitioning. In *Proceedings of the IEEE Data Compression Conference (DCC)*, pages 302–311, Snowbird, UT, March 23-25 2004.

[119] A. D. Liveris, Z. Xiong, and C. N. Georghiades. Nested convolutional/turbo codes for the binary Wyner-Ziv problem. In *Proceedings of the IEEE International Conference on Image Processing (ICIP): Special Session on Distributed Source Coding*, volume 1, pages 601–604, Barcelona, Spain, September 14-17 2003.

[120] A. D. Liveris, Z. Xiong, and C. N. Georghiades. Distributed compression of binary sources using conventional parallel and serial concatenated convolutional codes. In *Proceedings of the IEEE Data Compression Conference (DCC)*, pages 193–202, Snowbird, UT, March 25-27 2003.

[121] J. García-Frías and Y. Zhao. Compression of binary memoryless sources using punctured turbo codes. *IEEE Communications Letters*, 6(9):394–396, September 2002.

[122] J. García-Frías and Y. Zhao. Compression of correlated binary sources using turbo codes. *IEEE Communications Letters*, 5(10):417–419, October 2001.

[123] I. Deslauriers and J. Bajcsy. Serial turbo coding for data compression and the Slepian-Wolf problem. In *Proceedings of the IEEE Information Theory Workshop*, pages 296–299, Paris, France, March 31-April 4 2003.

[124] J. Bajcsy and P. Mitran. Coding for the Slepian-Wolf problem with turbo codes. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, volume 2, pages 1400–1404, San Antonio, TX, November 25-29 2001.

[125] J. Chou, S. S. Pradhan, and K. Ramchandran. Turbo and trellis-based constructions for source coding with side information. In *Proceedings of the IEEE Data Compression Conference (DCC)*, pages 33–42, Snowbird, Utah, March 25-27 2003.

[126] A. Aaron and B. Girod. Compression with side information using turbo codes. In *Proceedings of the IEEE Data Compression Conference (DCC)*, pages 252–261, Snowbird, UT, April 2-4 2002.

[127] A. D. Wyner and J. Ziv. The rate-distortion function for source coding with side information at the decoder. *IEEE Transactions on Information Theory*, 22(1):1–10, January 1976.

[128] A. D. Wyner. On source coding with side information at the decoder. *IEEE Transactions on Information Theory*, 21(3):294–300, May 1975.

[129] A. D. Wyner. The rate-distortion function for source coding with side information at the decoder - II: General sources. *IEEE Transactions on Information Theory*, 38(1):60–80, July 1978.

[130] S. S. Pradhan, J. Chou, and K. Ramchandran. Duality between source coding and channel coding and its extension to the side information case. *IEEE Transactions on Information Theory*, 49(5):1181–1203, May 2003.

[131] S. S. Pradhan and K. Ramchandran. Geometric proof of rate-distortion function of gaussian sources with side information at the decoder. In *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, page 351, Sorrento, Italy, June 25-30 2000.

[132] R. Zamir. The rate loss in the Wyner-Ziv problem. *IEEE Transactions on Information Theory*, 42(6):2073–2084, November 1996.

[133] R. Zamir, S. Shamai, and U. Erez. Nested linear/lattice codes for structured multiterminal binning. *IEEE Transactions on Information Theory*, 48(6):1250–1276, June 2002.

[134] R. Zamir and S. Shamai. Nested linear/lattice codes for Wyner-Ziv encoding. In *Proceedings of the IEEE Information Theory Workshop*, pages 92–93, Killarney, Co. Kerry, Ireland, June 22-26 1998.

[135] S. D. Servetto. Lattice quantization with side information: codes, asymptotics, and applications in sensor networks. *IEEE Transactions on Information Theory*, March 2004. to be published. [Online]. Available: http://cn.ece.cornell.edu/publications/papers/20020308/.

[136] S. D. Servetto. Lattice quantization with side information. In *Proceedings of the IEEE Data Compression Conference (DCC)*, pages 510–519, Snowbird, UT, March 28-30 2000.

[137] D. Muresan and M. Effros. Quantization as histogram segmentation: Globally optimal scalar quantizer design in network systems. In *Proceedings of the IEEE Data Compression Conference (DCC)*, pages 302–311, Snowbird, UT, April 2-4 2002.

[138] M. Fleming and M. Effros. Network vector quantization. In *Proceedings of the IEEE Data Compression Conference (DCC)*, pages 13–22, Snowbird, UT, March 27-29 2001.

[139] Z. Liu, S. Cheng, A. Liveris, and Z. Xiong. Slepian-wolf coded nested quantization (SWC-NQ) for Wyner-Ziv coding: Performance analysis and code design. In *Proceedings of the IEEE Data Compression Conference (DCC)*, pages 322–331, Snowbird, UT, March 23-25 2004.

[140] Y. Yang, S. Cheng, Z. Xiong, and W. Zhao. Wyner-Ziv coding based on TCQ and LDPC codes. In *Proceedings of the Thrity-Seventh Asilomar Conference on Signals, Systems, and Computers: Special Session on Distributed Methods in Image and Video Coding*, volume 1, pages 825–829, Pacific Grove, CA, November 9-12 2003.

[141] D. Rebollo-Monedero, R. Zhang, and B. Girod. Design of optimal quantizers for distributed source coding. In *Proceedings of the IEEE Data Compression Conference (DCC)*, pages 13–22, Snowbird, UT, March 25-27 2003.

[142] S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, March 1982.

[143] M. Gastpar, P.-L. Dragotti, and M. Vetterli. The distributed, partial, and conditional Karhunen-Loève transforms. In *Proceedings of the IEEE Data Compression Conference (DCC)*, pages 283–292, Snowbird, UT, March 25-27 2003.

[144] D. Rebollo-Monedero, A. Aaron, and B. Girod. Transforms for high-rate distributed source coding. In *Proceedings of the Thrity-Seventh Asilomar Conference on Signals, Systems, and Computers*, volume 1, pages 850–854, Pacific Grove, CA, November 9-12 2003.

[145] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero. Distributed video coding. *Proceedings of the IEEE: Special Issue on Video Coding and Delivery*, 2004. to appear. [Online]. Available: http://www.stanford.edu/∼bgirod/pdfs/DistributedVideoCoding-IEEEProc.pdf.

[146] S. Rane, A. Aaron, and B. Girod. Wyenr-Ziv video coding with hash-based motion compensation at the receiver. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, Singapore, October 24-27 2004.

[147] A. Aaron, S. Rane, E. Setton, and B. Girod. Transform-domain Wyner-Ziv codec for video. In *Proceedings of the SPIE International Conference on Visual Communications and Image Processing (VCIP)*, volume 5308, pages 520–528, San Jose, CA, January 18-22 2004.

[148] A. Aaron, E. Setton, and B. Girod. Towards practical Wyner-Ziv coding of video. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, volume 3, pages 869–872, Barcelona, Spain, September 14-17 2003.

[149] A. Aaron, R. Zhang, and B. Girod. Wyner-Ziv coding of motion video. In *Proceedings of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 240–244, Pacific Grove, CA, November 3-6 2002.

[150] R. Puri and K. Ramchandran. PRISM: A video voding paradigm based on motion-compensated prediction at the decoder. *IEEE Transactions on Image Processing*, submitted for publication. [Online]. Available: http://www-wavelet.eecs.berkeley.edu/∼rpuri/researchlinks/papers/purirvc2003.pdf.gz.

[151] R. Puri and K. Ramchandran. PRISM: A new robust video coding architecture based on distributed compression principles. In *Proceedings of the 40th Allerton Conference on Communication, Control and Computing*, Allerton, IL, October 2002. [Online]. Available: http://www-wavelet.eecs.berkeley.edu/∼rpuri/researchlinks/papers/purirnr2002.pdf.gz.

[152] R. Puri and K. Ramchandran. PRISM: An uplink-friendly multimedia coding paradigm. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 4, pages 856–859, Hong Kong, April 6-10 2003.

[153] R. Puri and K. Ramchandran. PRISM: A "reversed" multimedia coding paradigm. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, volume 1, pages 617–620, Barcelona, Spain, September 14-17 2003.

[154] A. Sehgal, A. Jagmohan, and N. Ahuja. Robust Wyner-Ziv coding of video. *IEEE Transactions on Multimedia*, 6(2):249–258, April 2004.

[155] A. Sehgal, A. Jagmohan, and N. Ahuja. A state-free causal video encoding paradigm. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, volume 1, pages 605–608, Barcelona, Spain, September 14-17 2003.

[156] A. Sehgal and N. Ahuja. Robust predictive coding and the Wyner-Ziv problem. In *Proceedings of the IEEE Data Compression Conference (DCC)*, pages 103–112, Snowbird, UT, March 25-27 2003.

[157] A. Jagmohan, A. Sehgal, and N. Ahuja. Predictive encoding using coset codes. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, volume 2, pages 29–32, Rochester, NY, September 22-25 2002.

[158] A. Sehgal, A. Jagmohan, and N. Ahuja. Scalable predictive coding as the Wyner-Ziv problem. In *Proceedings of the IEEE 8th International Conference on Communication Systems (ICCS)*, volume 1, pages 101–106, Singapore, November 25-28 2002.

[159] S. Cheng and Z. Xiong. Successive refinement for the Wyner-Ziv problem and layered code design. In *Proceedings of the IEEE Data Compression Conference (DCC)*, pages 531–531, Snowbird, UT, March 23-25 2004.

[160] Q. Xu and Z. Xiong. Layered Wyner-Ziv video coding. In *Proceedings of the SPIE International Conference on Visual Communications and Image Processing (VCIP): Special Session on Multimedia Technologies for Embedded Systems*, volume 5308, pages 83–91, San Jose, CA, January 18-22 2004.

[161] X. Zhu, A. Aaron, and B. Girod. Distributed compression for large camera arrays. In *Proceedings of the IEEE Workshop on Statistical Signal Processing (SSP)*, pages 30–33, St Louis, Missouri, September 28-October 1 2003.

[162] S. S. Pradhan and K. Ramchandran. Enhancing analog image transmission systems using digital side information: a new wavelet-based image coding paradigm. In *Proceedings of the IEEE Data Compression Conference (DCC)*, pages 63–72, Snowbird, Utah, March 27-29 2001.

[163] S. Rane, A. Aaron, and B. Girod. Systematic lossy forward error protection for error resilient digital video broadcasting. In *Proceedings of the SPIE International Conference on Visual Communications and Image Processing (VCIP)*, volume 5308, pages 588–595, San Jose, CA, January 18-22 2004.

[164] S. Rane, A. Aaron, and B. Girod. Systematic lossy forward error protection for error-resilient digital video broadcasting - A Wyner-Ziv coding approach. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, Singapore, October 24-27 2004.

[165] A. Aaron, S. Rane, R. Zhang, and B. Girod. Wyner-Ziv coding for video: Applications to compression and error resilience. In *Proceedings of the IEEE Data Compression Conference (DCC)*, pages 93–102, Snowbird, UT, March 25-27 2003.

[166] S. D. Servetto. Sensing Lena—Massively distributed compression of sensor images. In *Proceedings of the IEEE International Conference on Image Processing (ICIP): Special Session on Distributed Source Coding*, volume 1, pages 613–616, Barcelona, Spain, September 14-17 2003.

[167] W. B. Rabiner and A. P. Chandrakasan. Network-driven motion estimation for wireless video terminals. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(4):644–653, August 1997.

[168] K. Lillevold. Improved direct mode for B pictures in TML. ITU-T Video Coding Experts Group (VCEG) Q.15/SG16 Q15-K-44, Portland, Oregon, August 22-25 2000. [Online]. Available: http://ftp3.itu.int/av-arch/video-site/0008_Por/q15k44.doc.

[169] M. Flierl and B. Girod. Generalized B pictures and the draft H.264/AVC video-compression standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):587–597, July 2003.

[170] B.-M. Jeon and Y.-M. Park. Mode decision for B pictures in TML-5. ITU-T Video Coding Experts Group (VCEG) Q.6/SG16 VCEG-L10, Eibsee, Germany, January 9-12 2001. [Online]. Available: ftp3.itu.int/av-arch/video-site/0101_Eib/VCEG-L10.doc.

[171] Y. Liu, C. I. Podilchuk, and E. J. Delp. Evaluation of joint source and channel coding over wireless networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume V, pages 764–767, Hong Kong, April 6-10, 2003.

[172] Klaus Stuhlmuller, Niko Farber, Michael Link, and Bernd Girod. Analysis of video transmission over lossy channels. *IEEE Journal on Selected Area in Communications*, 18:1012–1032, June 2000.

[173] Robert E. Van Dyck and David J. Miller. Transport of wireless video using separate, concatenated, and joint source-channel coding. *Proceedings of IEEE*, 87:1734–1750, October 1999.

[174] Michael Link. Software provided by, Lucent Technologies.

[175] G. Gleming, A. E. Hoiydi, J. De Vriendt, G. Nikolaidis, F. Piolini, and M. Maraki. A flexible network architecture for UMTS. *IEEE Personal Communications*, pages 8–15, April 1998.

[176] B. Kreller, A. S.-B. Park, J. Meggers, G. Forsgren, E. Kovacs, and M. Rosinus. UMTS: a middleware architecture and mobile API approach. *IEEE Personal Communications*, pages 32–38, April 1998.

[177] K. Nichols, S. Blake, F. Baker, and D. Black. Definition of the differentiated services field (DS field) in the Ipv4 and Ipv6 headers. IETF RFC 2474, December 1998.

[178] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An architecture for differentiated services. IETF RFC 2475, December 1998.

[179] Philip A. Chou. Joint Source/Channel Coding: a position paper for the NSF workshop on source/channel coding. [Online]. Available: http://research.microsoft.com/∼pachou/publications.htm.

[180] R. E. Blahut. *Theory and Practice of Error Control Codes*. Addison Wesley, Reading, MA, 1983.

[181] Klaus Stuhlmuller, Michael Link, and Bernd Girod. Robust Internet video transmission based on scalable coding and unequal error protection. *Signal Processing: Image Communications*, 15(1-2):77–94, 1999.

[182] Michael Gallant and Faouzi Kossentini. Rate-distortion optimized layered coding with unequal error protection for robust Internet video. *IEEE Transactions on Circuits and Systems for Video Technology*, 11:357–372, March 2001.

[183] Pascal Frossard and Olivier Verscheure. AMISP: a complete content-based MPEG-2 error-resilient scheme. *IEEE Transactions on Circuits and Systems for Video Technology*, 11:989–998, September 2001.

[184] Pascal Frossard and Olivier Verscheure. Joint source/FEC rate selection for quality-optimal MPEG-2 video delivery. *IEEE Transactions on Image Processing*, 10:1815–1825, December 2001.

[185] Trista Pei chun Chen and Tsuhan Chen. Adaptive joint source-channel coding using rate shaping. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages 1985–1988, Orlando, Florida, May 13-17 2002.

[186] Wee Sun Lee, Mark R. Pickering, Michael R. Frater, and John F. Arnold. A robust codec for transmission of very low bit-rate video over channels with bursty errors. *IEEE Transactions on Circuits and Systems for Video Technology*, 10:1403–1412, December 2000.

[187] Seong-Won Yuk, Min-Gyu Kang, Byung-Cheol Shin, and Dong-Ho Cho. An adaptive redundancy control method for erasure-code-based real-time data transmission over the Internet. *IEEE Transactions on Multimedia*, 3:366–374, September 2001.

[188] Philip A. Chou, Alexander E. Mohr, Albert Wang, and Sanjeev Mehrotra. Error control for receiver-driven layered multicast of audio and video. *IEEE Transactions on Multimedia*, 3:108–122, March 2001.

[189] Wai tian Tan and Avideh Zakhor. Video multicast using layered FEC and scalable compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 11:373–386, March 2001.

[190] Y. Liu, P. Salama, and E. J. Delp. Error resilience of video transmission by rate-distortion optimization and adaptive packetization. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, volume 2, pages 613–616, Lausanne, Switzerland, August 26–29, 2002.

[191] Philip A. Chou and Zhourong Miao. Rate-distortion optimized streaming of packetized media. *IEEE Transactions on Multimedia*, submitted for publication. [Online]. Available: http://research.microsoft.com/∼pachou/publications.htm.

[192] Y. Takishima, M. Wada, and H. Murakami. Reversible variable length codes. *IEEE Transactions on Communications*, 43:148–152, February 1995.

[193] D. W. Redmill and N. G. Kingsbury. The EREC: an error-resilient technique for coding varible-length blocks of data. *IEEE Transactions on Image Processing*, 5:565–574, April 1996.

[194] R. Llados-Bernaus and R. L. Stevenson. Fixed-length entropy coding for robust video compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 8:745–755, October 1998.

[195] G. Cote, M. Gallant, and F. Kossentini. Semi-fixed length motion vector coding for H.263-based low data rate video compression. *IEEE Transactions on Image Processing*, 8:1451–1455, October 1999.

[196] Hongzhi Li and Chang Wen Chen. Robust image transmission with bi-directional synchronization and hierarchical error correction. *IEEE Transactions on Circuits and Systems for Video Technology*, 11:1183–1187, November 2001.

[197] Judy Y. Liao and John Villasenor. Adaptive Intra block update for robust transmission of H.263. *IEEE Transactions on Circuits and Systems for Video Technology*, 10:30–35, February 2000.

[198] Jong-Tzy Wang and Pao-Chi Chang. Error-propagation prevention technique for real-time video transmission over ATM networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 9:513–523, April 1999.

[199] Pao-Chi Chang and Tien-Hsu Lee. Precise and fast error tracking for error-resilient transmission of H.263 video. *IEEE Transactions on Circuits and Systems for Video Technology*, 10:600–607, June 2000.

[200] Rui Zhang, Shankar L. Regunathan, and Kenneth Rose. Video coding with optimal Inter/Intra-mode switching for packet loss resilience. *IEEE Journal on Selected Area in Communications*, 18:966–976, June 2000.

[201] Stephan Wenger and Michael Horowitz. Scattered Slices: a new error resilient tool for H.26L. JVT-B027, ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, January 29-February 1 2002.

[202] Jie Song and K. J. Ray Liu. A data embedded video coding scheme for error-prone channels. *IEEE Transactions on Multimedia*, 2:415–423, December 2001.

[203] Min Wu. *Multimedia data hiding*. Ph.D. Thesis, Princeton University, May 2001. [Online]. Available: http://www.ee.princeton.edu/∼minwu/research/phd_thesis.html.

[204] Chang-Su Kim, Rin-Chul Kim, and Sang-UK Lee. Robust transmission of video sequence over noisy channel using parity-check motion vector. *IEEE Transactions on Circuits and Systems for Video Technology*, 9:1063–1074, October 1999.

[205] Chang-Su Kim, Rin-Chul Kim, and Sang-UK Lee. An error detection and recovery algorithm for compressed video signal using source level redundancy. *IEEE Transactions on Image Processing*, 9:209–219, February 2000.

[206] Chang-Su Kim, Rin-Chul Kim, and Sang-UK Lee. Robust transmission of video sequence using double-vector motion compensation. *IEEE Transactions on Circuits and Systems for Video Technology*, 11:1011–1021, September 2001.

[207] Tamer Shanableh and Mohammed Ghanbari. The importance of the bi-directionally predicted pictures in video streaming. *IEEE Transactions on Circuits and Systems for Video Technology*, 11:402–414, March 2001.

[208] Chang-Su Kim and Sang-UK Lee. Multiple description coding of motion fields for robust video transmission. *IEEE Transactions on Circuits and Systems for Video Technology*, 11:999–1010, September 2001.

[209] Dapeng Wu, Yiwei Thomas Hou, and Ya qian Zhang. Scalable video coding and transport over broad-band wireless networks. *Proceedings of IEEE*, 89:6–20, January 2001.

[210] Coding of audio-visual objects, Part-2 Visual, Amendment 4: Streaming video profile. ISO/IEC 14496-2/FPDAM4, July 2000.

[211] Yan Yang and Sheila S. Hemami. Generalized rate-distortion optimization for motion-compensated video coders. *IEEE Transactions on Circuits and Systems for Video Technology*, 10:942–955, September 2000.

[212] Thomas Wiegand, Michael Lightstone, Dcbargha Mukherjee, T. George Campbell, and Sanjit K. Mitra. Generalized rate-distortion optimization for motion-compensated video coders. *IEEE Transactions on Circuits and Systems for Video Technology*, 6:182–190, April 1996.

[213] Lisimachos P. Kondi and Aggelos K. Katsaggelos. An operational rate-distortion optimal signle-pass SNR scalable video coder. *IEEE Transactions on Image Processing*, 10:1613–1620, November 2001.

[214] Rui Zhang. *End-to-end rate distortion analysis and optimization for robust video transmission over lossy networks*. Ph.D. Thesis, University of California at Santa Barbara, December 2001.

[215] Dapeng Wu, Yiwei Thomas Hou, Wenwu Zhu, Ya-Qin Zhang, and Jon M. Peha. Streaming video over the Internet: approaches and directions. *IEEE Transactions on Circuits and Systems for Video Technology*, 11:282–300, March 2001.

[216] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: a transport protocol for real-time applications, January 1996. [Online]. Available: http://www.ietf.org/rfc/rfc1889.txt.

[217] C. Zhu. RTP payload format for H.263 video streams, September 1997. [Online]. Available: http://www.ietf.org/rfc/rfc2190.txt.

[218] C. Bormann, L. Cline, G. Deisher, T. Gardos, C. Maciocco, D. Newell, J. Ott, G. Sullivan, S. Wenger, and C. Zhu. RTP payload format for the 1998 version of ITU-T rec. H.263 video (H.263+), October 1998. [Online]. Available: http://www.ietf.org/rfc/rfc2429.txt.

[219] Xiaolin Wu, Samuel Cheng, and Zixiang Xiong. On packetization of embedded multimedia bitstreams. *IEEE Transactions on Multimedia*, 3:132–140, March 2001.

[220] Z. Wang, Y. L. Yu, and D. Zhang. Best neighborhood matching - An information loss restoration technique for block based image coding systems. *IEEE Transactions on Image Processing*, 7(7):1056–1061, July 1998.

APPENDICES

APPENDICES

Appendix A: Rate Distortion Functions for Cascaded MSE Optimum Forward Channels

Proposition A.1 *The cascaded Gaussian MSE optimum forward channels are still optimal in the rate distortion sense. Furthermore, the parameter of the equivalent optimum forward channel is the sum of the parameters featuring each of the cascaded channels.*

Proof: We complete the proof by showing that the two cascaded optimum forward channels, as shown in Fig. 3.3, is equivalent to one optimum forward channel that yields the same MSE rate distortion functions for an arbitrary 2D Gaussian stationary signal.

From Fig. 3.3, we have

$$S''(\Lambda) = \hat{G}(\Lambda)S'(\Lambda) + \hat{N}(\Lambda), \tag{A.1}$$

where the transform function of the second optimum forward channel is represented by

$$\hat{G}(\Lambda) = \max\left\{0, 1 - \frac{\hat{\theta}}{\Phi_{s's'}(\Lambda)}\right\} = \max\left\{0, 1 - \frac{\hat{\theta}}{\max\{0, \Phi_{ss}(\Lambda) - \theta\}}\right\}, \tag{A.2}$$

and the PSD of the additive, independent Gaussian noise for the second channel is

$$\begin{aligned}\Phi_{\hat{n}\hat{n}}(\Lambda) &= \max\left\{0, \hat{\theta}\left(1 - \frac{\hat{\theta}}{\Phi_{s's'}(\Lambda)}\right)\right\} \\ &= \max\left\{0, \hat{\theta}\left(1 - \frac{\hat{\theta}}{\max\{0, \Phi_{ss}(\Lambda) - \theta\}}\right)\right\}. \end{aligned} \tag{A.3}$$

Since

$$S'(\Lambda) = G(\Lambda)S(\Lambda) + N(\Lambda) = \max\left\{0, 1 - \frac{\theta}{\Phi_{ss}(\Lambda)}\right\} S(\Lambda) + N(\Lambda), \tag{A.4}$$

combining (A.1), we have

$$S''(\Lambda) = G(\Lambda)\hat{G}(\Lambda)S(\Lambda) + \hat{G}(\Lambda)N(\Lambda) + \hat{N}(\Lambda). \tag{A.5}$$

Since $\{s\}$, $\{n\}$, and $\{\hat{n}\}$ are all stationary Gaussian processes, and independent of each other, $\{s''\}$, as a linear combination of them, is also a stationary Gaussian process, and jointly stationary Gaussian with $\{s\}$.

Let $\{\tilde{s}''\}$ denote the difference signal between the input to the first channel and the output from the second channel in Fig. 3.3, we have

$$\begin{aligned}\tilde{S}''(\Lambda) &= S(\Lambda) - S''(\Lambda) \\ &= (1 - G(\Lambda)\hat{G}(\Lambda))S(\Lambda) - \hat{G}(\Lambda)N(\Lambda) - \hat{N}(\Lambda). \end{aligned} \tag{A.6}$$

For the spatial frequencies $\Lambda : \Phi_{ss}(\Lambda) \leq \theta$, it is easy to show that $\Phi_{s's'}(\Lambda)$, $\Phi_{s''s''}(\Lambda)$, and the cross spectral density between $\{s\}$ and $\{s''\}$, denoted as $\Phi_{ss''}(\Lambda)$, are all zeros, and $\Phi_{\tilde{s}''\tilde{s}''}(\Lambda) = \Phi_{ss}(\Lambda)$. For $\Lambda : \Phi_{ss}(\Lambda) > \theta$, we have

$$\begin{aligned}
\Phi_{\tilde{s}''\tilde{s}''}(\Lambda) &= |1 - G(\Lambda)\hat{G}(\Lambda)|^2\Phi_{ss}(\Lambda) + |\hat{G}(\Lambda)|^2\Phi_{nn}(\Lambda) + \Phi_{\hat{n}\hat{n}}(\Lambda) \\
&= \left|1 - \frac{\Phi_{ss}(\Lambda) - \theta}{\Phi_{ss}(\Lambda)} \max\left\{0, \frac{\Phi_{ss}(\Lambda) - (\theta + \hat{\theta})}{\Phi_{ss}(\Lambda) - \theta}\right\}\right|^2 \Phi_{ss}(\Lambda) \\
&\quad + \left|\max\left\{0, \frac{\Phi_{ss}(\Lambda) - (\theta + \hat{\theta})}{\Phi_{ss}(\Lambda) - \theta}\right\}\right|^2 \theta\frac{\Phi_{ss}(\Lambda) - \theta}{\Phi_{ss}(\Lambda)} \\
&\quad + \max\left\{0, \hat{\theta}\frac{\Phi_{ss}(\Lambda) - (\theta + \hat{\theta})}{\Phi_{ss}(\Lambda) - \theta}\right\} \\
&= \min\{\theta + \hat{\theta}, \Phi_{ss}(\Lambda)\}. \qquad \text{(A.7)}
\end{aligned}$$

Since both θ and $\hat{\theta}$ are nonnegative, $\Phi_{ss}(\Lambda) \leq \theta$ implies that $\Phi_{ss}(\Lambda) \leq \theta + \hat{\theta}$. Hence, $\Phi_{\tilde{s}''\tilde{s}''}(\Lambda)$ has a universal form, regardless of the specific spatial frequencies, as given in (A.7). The MSE distortion yielded by the cascaded channels is then obtained as the inverse-Fourier transform of the PSD $\Phi_{\tilde{s}''\tilde{s}''}(\Lambda)$, which is

$$D^{II,\theta,\tilde{\theta}} = \frac{1}{4\pi^2}\iint_{\Lambda} \Phi_{\tilde{s}''\tilde{s}''}(\Lambda)d\Lambda = \frac{1}{4\pi^2}\iint_{\Lambda} \min\{\tilde{\theta}, \Phi_{ss}(\Lambda)\}d\Lambda, \qquad \text{(A.8)}$$

where $\tilde{\theta} = \theta + \hat{\theta}$.

Also by (A.5), for $\Lambda : \Phi_{ss}(\Lambda) > \theta$, the cross spectral density $\Phi_{ss''}(\Lambda)$ is

$$\begin{aligned}
\Phi_{ss''}(\Lambda) &= G(\Lambda)\hat{G}(\Lambda)\Phi_{ss}(\Lambda) \\
&= \frac{\Phi_{ss}(\Lambda) - \theta}{\Phi_{ss}(\Lambda)} \max\left\{0, \frac{\Phi_{ss}(\Lambda) - (\theta + \hat{\theta})}{\Phi_{ss}(\Lambda) - \theta}\right\} \Phi_{ss}(\Lambda) \\
&= \max\{0, \Phi_{ss}(\Lambda) - (\theta + \hat{\theta})\}, \qquad \text{(A.9)}
\end{aligned}$$

and the PSD of $\{s''\}$ is

$$\begin{aligned}
\Phi_{s''s''}(\Lambda) &= |G(\Lambda)\hat{G}(\Lambda)|^2\Phi_{ss}(\Lambda) + |\hat{G}(\Lambda)|^2\Phi_{nn}(\Lambda) + \Phi_{\hat{n}\hat{n}}(\Lambda) \\
&= \left|\frac{\Phi_{ss}(\Lambda)-\theta}{\Phi_{ss}(\Lambda)}\max\left\{0, \frac{\Phi_{ss}(\Lambda)-(\theta+\hat{\theta})}{\Phi_{ss}(\Lambda)-\theta}\right\}\right|^2 \Phi_{ss}(\Lambda) \\
&\quad + \left|\max\left\{0, \frac{\Phi_{ss}(\Lambda)-(\theta+\hat{\theta})}{\Phi_{ss}(\Lambda)-\theta}\right\}\right|^2 \theta\frac{\Phi_{ss}(\Lambda)-\theta}{\Phi_{ss}(\Lambda)} \\
&\quad + \max\left\{0, \hat{\theta}\frac{\Phi_{ss}(\Lambda)-(\theta+\hat{\theta})}{\Phi_{ss}(\Lambda)-\theta}\right\} \\
&= \max\{0, \Phi_{ss}(\Lambda)-(\theta+\hat{\theta})\}. \qquad \text{(A.10)}
\end{aligned}$$

Similar to the discussion with $\Phi_{\tilde{s}''\tilde{s}''}(\Lambda)$ in (A.7), (A.9) and (A.10) are also the universal forms for $\Phi_{ss''}(\Lambda)$ and $\Phi_{s''s''}(\Lambda)$. Note that the PSD of the output from the cascaded channels, $\Phi_{s''s''}(\Lambda)$, and the cross spectral density between the input and the output of the channels, $\Phi_{ss''}(\Lambda)$, are identical.

Since $\{s\}$ and $\{s''\}$ are jointly stationary Gaussian, the mutual information rate between $\{s\}$ and $\{s''\}$, $I^{\theta,\hat{\theta}}(S;S'')$, is obtained using (A.9), (A.10), and (3.5) of Lemma 1 as follows

$$\begin{aligned}
I^{\theta,\hat{\theta}}(S;S'') &= -\frac{1}{8\pi^2}\iint_\Lambda \log\left(1-\frac{|\Phi_{ss''}(\Lambda)|^2}{\Phi_{ss}(\Lambda)\Phi_{s''s''}(\Lambda)}\right)d\Lambda \\
&= -\frac{1}{8\pi^2}\iint_\Lambda \log\left(1-\frac{|\max\{0,\Phi_{ss}(\Lambda)-(\theta+\hat{\theta})\}|^2}{\Phi_{ss}(\Lambda)\max\{0,\Phi_{ss}(\Lambda)-(\theta+\hat{\theta})\}}\right)d\Lambda \\
&= \frac{1}{8\pi^2}\iint_\Lambda \max\left\{0, \log\left(\frac{\Phi_{ss}(\Lambda)}{\theta+\hat{\theta}}\right)\right\}d\Lambda. \qquad \text{(A.11)}
\end{aligned}$$

This implies the parametric data rate yielded by the cascaded channels is

$$R^{II,\theta,\tilde{\theta}} = \frac{1}{8\pi^2}\iint_\Lambda \max\left\{0, \log\left(\frac{\Phi_{ss}(\Lambda)}{\tilde{\theta}}\right)\right\}d\Lambda, \qquad \text{(A.12)}$$

where $\tilde{\theta} = \theta + \hat{\theta}$.

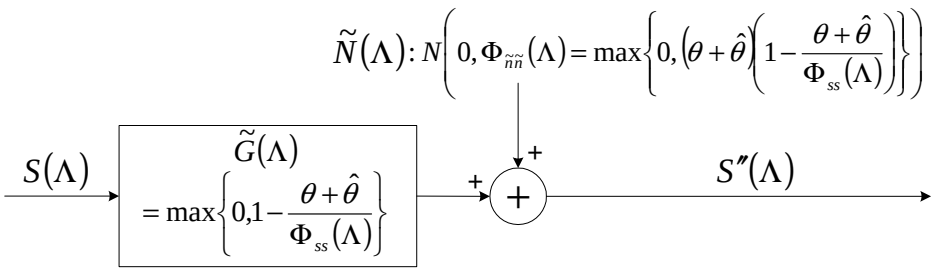


Fig. A.1. The equivalent MSE optimum forward channel for the two cascaded channels

Combining (A.8) and (A.12), we conclude that the two cascaded optimum forward channels yield the Gaussian MSE rate distortion functions of the same form as that yielded by a single optimum forward channel except with parameter $\theta + \hat{\theta}$, the sum of the two parameters featuring the two channels. Equivalently, the two cascaded channels in Fig. 3.3 can be represented by one optimum forward channel as shown in Fig. A.1, where the transform function is represented by

$$\tilde{G}(\Lambda) = \max\left\{0, 1 - \frac{\theta + \hat{\theta}}{\Phi_{ss}(\Lambda)}\right\}, \tag{A.13}$$

and the PSD of the additive, independent Gaussian noise is

$$\Phi_{\tilde{n}\tilde{n}}(\Lambda) = \max\left\{0, (\theta + \hat{\theta})\left(1 - \frac{\theta + \hat{\theta}}{\Phi_{ss}(\Lambda)}\right)\right\}. \tag{A.14}$$

■

Appendix B: The Development of the Alternative Block Diagram for Leaky Prediction Layered Video Coding (LPLC)

From Fig. 3.7, we have

$$S'_b(\Omega) = E'_b(\Omega) + H(\Omega)S'_b(\Omega),$$

implying that

$$E'_b(\Omega) = [1 - H(\Omega)]S'_b(\Omega), \tag{B.1}$$

and

$$S'_b(\Omega) = \frac{E'_b(\Omega)}{1 - H(\Omega)}. \tag{B.2}$$

Also, we have

$$\begin{aligned} S'_e(\Omega) &= E'_e(\Omega) + \hat{S}_e(\Omega) = E'_b(\Omega) + \Psi'(\Omega) + \hat{S}_e(\Omega) \\ &= E'_b(\Omega) + \Psi'(\Omega) + H(\Omega)\left[(1-\alpha)S'_b(\Omega) + \alpha S'_e(\Omega)\right], \end{aligned}$$

which implies, by using (B.1), that

$$\begin{aligned} S'_e(\Omega) &= \frac{E'_b(\Omega) + H(\Omega)(1-\alpha)S'_b(\Omega) + \Psi'(\Omega)}{1 - \alpha H(\Omega)} \\ &= S'_b(\Omega) + \frac{1}{1 - \alpha H(\Omega)}\Psi'(\Omega), \end{aligned} \tag{B.3}$$

and

$$\hat{S}_e(\Omega) = S'_e(\Omega) - E'_b(\Omega) - \Psi'(\Omega). \tag{B.4}$$

Combining (B.4) and (B.3), we have

$$\begin{aligned}\Psi(\Omega) &= E_e(\Omega) - E_b'(\Omega) = S(\Omega) - \hat{S}_e(\Omega) - E_b'(\Omega) \\ &= S(\Omega) - S_e'(\Omega) + \Psi'(\Omega) \\ &= [S(\Omega) - S_b'(\Omega)] - \frac{\alpha H(\Omega)}{1 - \alpha H(\Omega)}\Psi'(\Omega).\end{aligned}$$

Since

$$S(\Omega) - S_b'(\Omega) = E_b(\Omega) - E_b'(\Omega) \triangleq \tilde{E}_b(\Omega), \tag{B.5}$$

then

$$\Psi(\Omega) = \tilde{E}_b(\Omega) - \frac{\alpha H(\Omega)}{1 - \alpha H(\Omega)}[G_e(\Lambda)\Psi(\Omega) + N_e(\Omega)].$$

Hence

$$\begin{aligned}\Psi(\Omega) &= \frac{1 - \alpha H(\Omega)}{1 - \alpha H(\Omega) + G_e(\Lambda)(\alpha H(\Omega))}\tilde{E}_b(\Omega) \\ &\quad - \frac{\alpha H(\Omega)}{1 - \alpha H(\Omega) + G_e(\Lambda)(\alpha H(\Omega))}N_e(\Omega).\end{aligned} \tag{B.6}$$

We observe that the Fourier transform of the mismatch signal $\{\psi\}$ in (B.6), as a function of $\{\tilde{e}_b \triangleq e_b - e_b'\}$ and $\{n_e\}$, has exactly the same form as that of the PEF in the base layer MCP loop, $\{e_b\}$, as a function of $\{s\}$ and $\{n_b\}$ [64]. Analogously, $\{\tilde{e}_b\}$ serves as the input signal to the MCP loop as opposed to $\{s\}$, $\{n_e\}$ serves as the additive, independent Gaussian noise in the optimum forward channel as opposed to $\{n_b\}$, and $(\alpha H(\Omega))$ is the MCP 3D filter combining spatial filtering and motion compensation as opposed to $H(\Omega)$. Therefore, we obtain the alternative diagram for LPLC as shown in Fig. 3.8.

Appendix C: Further Analysis of *Scenario II for LPLC*

If $F(\Lambda) = P(\Lambda)$, we have $\Phi_{\psi\psi}^{I,\breve{\theta}_0}(\Lambda)$ in (3.37) for $\Lambda : \Phi_{\tilde{e}_b\tilde{e}_b}(\Lambda) > \breve{\theta}_0$ that

$$\begin{aligned}\Phi_{\psi\psi}^{I,\breve{\theta}_0}(\Lambda) &= \Phi_{\tilde{e}_b\tilde{e}_b}(\Lambda)[1-\alpha(2-\alpha)|P(\Lambda)|^2] + \breve{\theta}_0\alpha|P(\Lambda)|^2 \\ &< \Phi_{\tilde{e}_b\tilde{e}_b}(\Lambda)[1-2\alpha(1-\alpha)|P(\Lambda)|^2] \le \Phi_{\tilde{e}_b\tilde{e}_b}(\Lambda). \end{aligned} \tag{C.1}$$

Hence, when $\tilde{\theta} = \breve{\theta}_0 + \hat{\theta} > \theta$ and $\breve{\theta}_0 \le \theta$, combining (3.35), (C.1), and (3.27), we have

$$\Phi_{\psi\psi}^{\breve{\theta}_0}(\Lambda) \le \Phi_{\tilde{e}_b\tilde{e}_b}(\Lambda) \le \theta < \tilde{\theta}. \tag{C.2}$$

Thus the data rate function in (3.49) reduces to the same form as (3.30), but the distortion function reduces to

$$\begin{aligned} D_e^{II,\theta,\tilde{\theta}} &= \frac{1}{4\pi^2}\iint_\Lambda \min\left\{\breve{\theta}_0, \Phi_{\psi\psi}^{\breve{\theta}_0}(\Lambda)\right\} \\ &\quad + \frac{1}{1-\alpha^2|P(\Lambda)|^2}\max\left\{0, \Phi_{\psi\psi}^{\breve{\theta}_0}(\Lambda) - \breve{\theta}_0\right\} d\Lambda, \\ &\quad \text{for } \breve{\theta}_0 \le \theta,\ \tilde{\theta} > \theta, \text{ and } F(\Lambda) = P(\Lambda). \end{aligned} \tag{C.3}$$

Note that when $\alpha = 0$,

$$\begin{aligned} &\min\left\{\breve{\theta}_0, \Phi_{\psi\psi}^{\breve{\theta}_0}(\Lambda)\right\} + \frac{1}{1-\alpha^2|P(\Lambda)|^2}\max\left\{0, \Phi_{\psi\psi}^{\breve{\theta}_0}(\Lambda) - \breve{\theta}_0\right\} \\ = \ &\Phi_{\psi\psi}^{\breve{\theta}_0}(\Lambda) = \Phi_{\tilde{e}_b\tilde{e}_b}(\Lambda), \end{aligned}$$

and the distortion given by (C.3) reduces to that given by (3.29). When $\alpha = 1$, for $\Lambda : \Phi_{\tilde{e}_b\tilde{e}_b}(\Lambda) > \breve{\theta}_0$, we have $\Phi_{\psi\psi}^{\breve{\theta}_0}(\Lambda) \ge \breve{\theta}_0$, and

$$\begin{aligned} &\frac{1}{1-|P(\Lambda)|^2}\max\left\{0, \Phi_{\psi\psi}^{\breve{\theta}_0}(\Lambda) - \breve{\theta}_0\right\} \\ = \ &\frac{1}{1-|P(\Lambda)|^2}\max\left\{0, \Phi_{\tilde{e}_b\tilde{e}_b}(\Lambda)(1-|P(\Lambda)|^2) + \breve{\theta}_0|P(\Lambda)|^2 - \breve{\theta}_0\right\} \\ = \ &\Phi_{\tilde{e}_b\tilde{e}_b}(\Lambda) - \breve{\theta}_0. \end{aligned}$$

Thus it is easy to show that the distortion given by (C.3) also reduces to that given by (3.29) for $\alpha = 1$.

Appendix D: A Discussion of Embedded Quantization Noise

In this section, we discuss the variance of the signal $\{\Delta q_{e,\mathrm{dec}}\}$ in (3.86), $\sigma^2_{\Delta q_{e,\mathrm{dec}}}$, and the cross-correlation between $\{\Delta q_{e,\mathrm{dec}}\}$ and $\{q_{e,\min}\}$, $E\{q_{e,\min}\Delta q_{e,\mathrm{dec}}\}$.

Let matrix U denote the orthogonal, separable 2D transform applied to the mismatch signal $\{\psi\}$ carried by the enhancement layer in LPLC. U has a dimension of $M \times M$.[1]

We make the following three assumptions: First, we assume uniform embedded quantization operations are used in encoding the transform coefficients of $\{\psi\}$. Let δ_γ, $\gamma = 0, 1, 2, \cdots$, denote the selected series of embedded quantization steps, where

$$\delta_\gamma = 2^\gamma \delta_0, \quad \text{for } \gamma = 1, 2, 3 \cdots . \tag{D.1}$$

The enhancement layer data rates in accordance to the use of above embedded quantization steps are $R_e^{(\gamma)}$, for $\gamma = 0, 1, 2, \cdots$.

Secondly, we assume drift occurs as a result of data rate truncation to the bitstream of the enhancement layer, and the possible truncation point only occurs at one of the discrete data rates $R_e^{(\gamma)}$, $\gamma = 0, 1, 2, \cdots$.

Thirdly, we assume that the reconstruction level of an arbitrary quantizer is placed in the middle of the corresponding quantization interval.

[1] We assume U has a square size. This conforms with the transform used in most real video coding systems, such as the 8×8 block based DCT in MPEG-2 and the 4×4 block based DCT-like integer transform in H.26L/H.264.

Let $Q_{e,\min}$ and $Q_{e,\text{dec}}$ denote the quantization noise matrices when the enhancement layer is decoded at data rate $(R_{e,\min} - R_b)$ and $(R^{II}_{e,\text{dec}} - R_b)$ respectively, and let

$$\Delta Q_{e,\text{dec}} \triangleq Q_{e,\text{dec}} - Q_{e,\min}. \tag{D.2}$$

Then we have

$$\begin{aligned} q_{e,\min}(x,y,t) &= (U^T Q_{e,\min} U)(x,y,t), & \text{(D.3)} \\ q_{e,\text{dec}}(x,y,t) &= (U^T Q_{e,\text{dec}} U)(x,y,t), & \text{(D.4)} \\ \Delta q_{e,\text{dec}}(x,y,t) &= (U^T \Delta Q_{e,\text{dec}} U)(x,y,t). & \text{(D.5)} \end{aligned}$$

The cross-correlation between $\{\Delta q_{e,\text{dec}}\}$ and $\{q_{e,\min}\}$ is

$$\begin{aligned} & E\left\{q_{e,\min}\Delta q_{e,\text{dec}}\right\}(\tau_x,\tau_y,t) \\ = \; & E\left\{q_{e,\min}(x,y,t)\Delta q_{e,\text{dec}}(x+\tau_x,y+\tau_y,t)\right\} \\ = \; & E\left\{\left(U^T Q_{e,\min} U\right)(x,y,t)\left(U^T \Delta Q_{e,\text{dec}} U\right)(x+\tau_x,y+\tau_y,t)\right\}, \end{aligned} \tag{D.6}$$

where (τ_x, τ_y) denote an arbitrary pair of integer distances satisfying $-(M-1) \leq \tau_x \leq (M-1)$ and $-(M-1) \leq \tau_y \leq (M-1)$.

Since both $Q_{e,\min}$ and $\Delta Q_{e,\text{dec}}$ have a dimension of $M \times M$, we write

$$Q_{e,\min} = \begin{bmatrix} Q_{e,\min}(\mu_1,v_1,t) & Q_{e,\min}(\mu_1,v_2,t) & \dots & Q_{e,\min}(\mu_1,v_M,t) \\ Q_{e,\min}(\mu_2,v_1,t) & Q_{e,\min}(\mu_2,v_2,t) & \dots & Q_{e,\min}(\mu_2,v_M,t) \\ \vdots & \vdots & \ddots & \vdots \\ Q_{e,\min}(\mu_M,v_1,t) & Q_{e,\min}(\mu_M,v_2,t) & \dots & Q_{e,\min}(\mu_M,v_M,t) \end{bmatrix}, \tag{D.7}$$

and

$$\Delta Q_{e,\mathrm{dec}} = \begin{bmatrix} \Delta Q_{e,\mathrm{dec}}(\mu_1, v_1, t) & \Delta Q_{e,\mathrm{dec}}(\mu_1, v_2, t) & \dots & \Delta Q_{e,\mathrm{dec}}(\mu_1, v_M, t) \\ \Delta Q_{e,\mathrm{dec}}(\mu_2, v_1, t) & \Delta Q_{e,\mathrm{dec}}(\mu_2, v_2, t) & \dots & \Delta Q_{e,\mathrm{dec}}(\mu_2, v_M, t) \\ \vdots & \vdots & \ddots & \vdots \\ \Delta Q_{e,\mathrm{dec}}(\mu_M, v_1, t) & \Delta Q_{e,\mathrm{dec}}(\mu_M, v_2, t) & \dots & \Delta Q_{e,\mathrm{dec}}(\mu_M, v_M, t) \end{bmatrix}. \tag{D.8}$$

Next we prove

$$E\left\{Q_{e,\min}(\mu^{(1)}, v^{(1)}, t)\Delta Q_{e,\mathrm{dec}}(\mu^{(2)}, v^{(2)}, t)\right\} \approx 0, \tag{D.9}$$

i.e., $\{q_{e,\min}\}$ and $\{\Delta q_{e,\mathrm{dec}}\}$ are approximately uncorrelated in the transform domain, where $(\mu^{(1)}, v^{(1)})$ and $(\mu^{(2)}, v^{(2)})$ denote two arbitrary locations in the 2D $M \times M$ grid. If (D.9) is true, we have

$$E\left\{q_{e,\min}\Delta q_{e,\mathrm{dec}}\right\}(\tau_x, \tau_y, t) \approx 0, \tag{D.10}$$

implying that $\{q_{e,\min}\}$ and $\{\Delta q_{e,\mathrm{dec}}\}$ are also approximately uncorrelated in the spatial domain. This is true because by (D.6), $\left(U^T Q_{e,\min} U\right)(x, y, t)$ is a linear combination of the entries of matrix $Q_{e,\min}$ in (D.7), and $\left(U^T \Delta Q_{e,\mathrm{dec}} U\right)(x + \tau_x, y + \tau_y, t)$ is a linear combination of the entries of matrix $\Delta Q_{e,\mathrm{dec}}$ in (D.8).

For a quantization noise as a result of the use of a specific quantization step δ_γ, we assume the spatial samples located in the 2D $M \times M$ grid are identically distributed random variables, each with a zero-mean, and any two samples from different locations are uncorrelated.

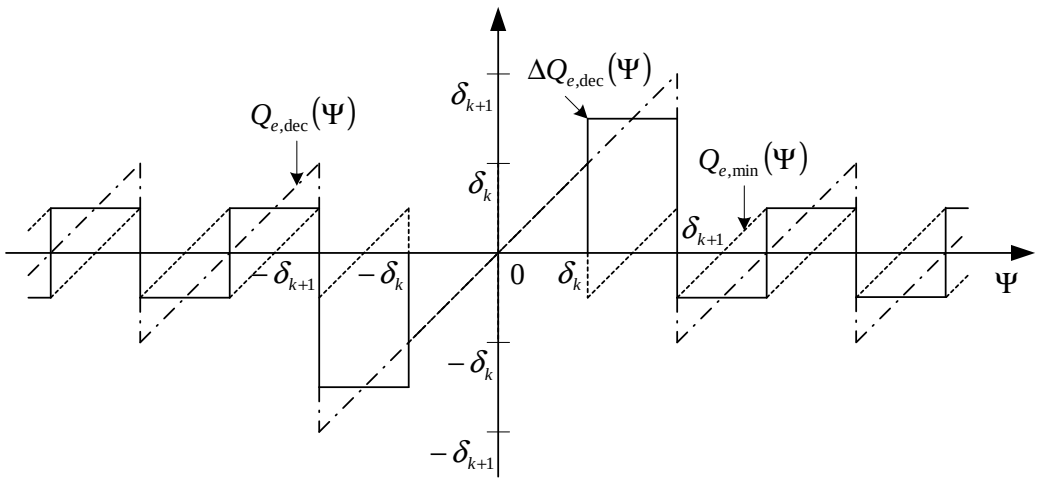


Fig. D.1. Quantization noise introduced by the use of uniform embedded quantization steps

Hence, when at least one coordinate does not agree for $(\mu^{(1)}, v^{(1)})$ and $(\mu^{(2)}, v^{(2)})$, we have

$$E\left\{Q_{e,\min}(\mu^{(1)}, v^{(1)}, t)\Delta Q_{e,\mathrm{dec}}(\mu^{(2)}, v^{(2)}, t)\right\} = 0, \quad \text{for } \mu^{(1)} \neq \mu^{(2)} \text{ or } v^{(1)} \neq v^{(2)}. \tag{D.11}$$

Next we discuss the case when both coordinates for $(\mu^{(1)}, v^{(1)})$ and $(\mu^{(2)}, v^{(2)})$ agree to each other. First, suppose $Q_{e,\min}$ is the quantization noise introduced by the use of the quantization step δ_k, and $Q_{e,\mathrm{dec}}$ as a result of δ_{k+1}. We have

$$\delta_{k+1} = 2\delta_k.$$

As shown in Fig. D.1, Ψ is a random variable representing an arbitrary transform coefficient of the mismatch signal $\{\psi\}$. Let $f_\Psi(\varepsilon)$ denote the probability density function (p.d.f.) of Ψ. We have

$$\begin{aligned} & E\left\{Q_{e,\min}(\mu,v,t)\Delta Q_{e,\mathrm{dec}}(\mu,v,t)\right\} \\ = \; & E\left\{Q_{e,\min}(\Psi)\Delta Q_{e,\mathrm{dec}}(\Psi)\right\} \\ = \; & \int_{-\infty}^{\infty} Q_{e,\min}(\varepsilon)\Delta Q_{e,\mathrm{dec}}(\varepsilon) f_\Psi(\epsilon) d\varepsilon. \end{aligned} \tag{D.12}$$

We assume $f_\Psi(\varepsilon)$ is asymptotically zero outside a finite region and partition the non-zero support of $f_\Psi(\varepsilon)$ into N intervals, each with a length of δ_k. All the intervals, $[n\delta_k, (n+1)\delta_k),\ n = 0, 1, \cdots, N-1$, coordinate with the quantization intervals specified by δ_k, as shown in Fig. D.1. Then,

$$\begin{aligned} & E\left\{Q_{e,\min}(\Psi)\Delta Q_{e,\mathrm{dec}}(\Psi)\right\} \\ = \; & \sum_{n=0}^{N-1} \int_{n\delta_k}^{(n+1)\delta_k} Q_{e,\min}(\varepsilon)\Delta Q_{e,\mathrm{dec}}(\varepsilon) f_\Psi(\epsilon) d\varepsilon. \end{aligned} \tag{D.13}$$

We assume $R_e^{(k)}$ is sufficiently large, and hence δ_k is sufficiently small, so that $f_\Psi(\varepsilon)$ is approximately constant over each interval. Thus,

$$\begin{aligned} & E\left\{Q_{e,\min}(\Psi)\Delta Q_{e,\mathrm{dec}}(\Psi)\right\} \\ \approx \; & \sum_{n=0}^{N-1} f_\Psi(n) \int_{n\delta_k}^{(n+1)\delta_k} Q_{e,\min}(\varepsilon)\Delta Q_{e,\mathrm{dec}}(\varepsilon) d\varepsilon. \end{aligned} \tag{D.14}$$

From Fig. D.1, we have

$$\int_{n\delta_k}^{(n+1)\delta_k} Q_{e,\min}(\varepsilon)\Delta Q_{e,\mathrm{dec}}(\varepsilon) d\varepsilon = 0, \quad \text{for } n = 0, 1, \cdots, N-1. \tag{D.15}$$

Therefore,

$$E\left\{Q_{e,\min}(\mu,v,t)\Delta Q_{e,\mathrm{dec}}(\mu,v,t)\right\} \approx 0. \tag{D.16}$$

Above analysis can be easily generalized to obtain the result in (D.16) for the case where $q_{e,\min}$ and $q_{e,\mathrm{dec}}$ are a result of any two arbitrary quantization steps δ_k and δ_l, and $\delta_k < \delta_l$.

Combining (D.6), (D.9), (D.11), and (D.16), we have

$$E\left\{q_{e,\min}\Delta q_{e,\mathrm{dec}}\right\} \approx 0,$$

i.e., $\{q_{e,\min}\}$ and $\{\Delta q_{e,\mathrm{dec}}\}$ are approximately uncorrelated with each other.

Moreover, we approximate the variance of $\{\Delta q_{e,\mathrm{dec}}\}$ as

$$\begin{aligned}
\sigma^2_{\Delta q_{e,\mathrm{dec}}} &= E\left\{(q_{e,\mathrm{dec}} - q_{e,\min})^2\right\} \\
&= \sigma^{2(II)}_{q_{e,\mathrm{dec}}} - \sigma^2_{q_{e,\min}} - 2E\left\{q_{e,\min}\Delta q_{e,\mathrm{dec}}\right\} \\
&\approx \sigma^{2(II)}_{q_{e,\mathrm{dec}}} - \sigma^2_{q_{e,\min}}. \qquad \text{(D.17)}
\end{aligned}$$

VITA

VITA

Yuxin Liu was born in Beijing, P. R. China. She earned her BS degree (with highest honor) in 1995 and her MS in 2000, both in electrical engineering from Tsinghua University, Beijing, China. She was supervised by Professor Yanda Li for her graduate study at Tsinghua on wavelet image coding and image data hiding.

Since Fall 2000, she has been pursuing her Ph.D. degree in the School of Electrical and Computer Engineering at Purdue University, West Lafayette, Indiana. She had been a teaching assistant from August 2000 through May 2001, supervising the Computer Design and Prototyping Lab for undergraduate studies.

Since Fall 2001, she has been a research assistant in the Video and Image Processing Laboratory (VIPER) under the supervision of Professor Edward J. Delp. She has been co-advised by Professor Paul Salama associated with the Department of Electrical and Computer Engineering at Indiana University - Purdue University - Indianapolis (IUPUI), Indianapolis, Indiana. Her research work has been funded by a grant from the Indiana Twenty-First Century Research and Technology Fund. Her current research interests include video compression (MPEG-4/H.26L/H.264), multimedia communications, wireless networking, digital watermarking, and wavelet signal processing.

During the summers of 2001 and 2002, she worked at Bell Laboratories, Lucent Technologies, Murray Hill, New Jersey, as a summer intern. She worked with Dr. Christine I. Podilchuk on reliable video streaming over wireless networks and pre-/post processing of compressed videos.

Yuxin Liu is a member of the Eta Kappa Nu (HKN) national electrical engineering honor society and a student member of the IEEE professional society.

Yuxin Liu's publications for her research work at Purdue include:

Journal papers:

1. Yuxin Liu, Paul Salama, Zhen Li, and Edward J. Delp, "An Enhancement of Leaky Prediction Layered Video Coding," submitted to the *IEEE Transactions on Circuits and Systems for Video Technology* (has finished the first revision).

2. Gregory W. Cook, Josep Prades-Nebot, Yuxin Liu, and Edward J. Delp, "Rate-Distortion Analysis of Motion Compensated Rate Scalable Video," submitted to the *IEEE Transactions on Image Processing* (has finished the first revision).

3. Yuxin Liu, Bin Ni, Xiaojun Feng, and Edward Delp, "LOT-Based Adaptive Image Watermarking," submitted to the *Journal of Electronic Imaging*, February 2004.

4. Yuxin Liu, Josep Prades-Nebot, Gregory W. Cook, Paul Salama, and Edward J. Delp, "Rate-Distortion Analysis of Leaky Prediction Layered Video Coding," submitted to the *IEEE Transactions on Image Processing*, July 2004.

Conference papers:

1. Yuxin Liu, Josep Prades-Nebot, Paul Salama, and Edward J. Delp, "Low Complexity Video Encoding Using B-Frame Direct Modes," submitted to the *SPIE International Conference on Image and Video Communications and Processing (IVCP)*, January 16-20, 2005, San Jose, CA.

2. Yuxin Liu, Josep Prades-Nebot, Gregory W. Cook, Paul Salama, and Edward J. Delp, "Rate Distortion Performance of Leaky Prediction Layered Video Coding: Theoretic Analysis and Results," submitted to the *SPIE International Conference on Image and Video Communications and Processing (IVCP)*, January 16-20, 2005, San Jose, CA.

3. Limin Liu, Yuxin Liu, and Edward J. Delp, "Network-Driven Wyner-Ziv Video Coding," submitted to the *SPIE International Conference on Image and Video Communications and Processing (IVCP)*, January 16-20, 2005, San Jose, CA.

4. Eugene T. Lin, Yuxin Liu, and Edward J. Delp, "Detection of Mass Tumors in Mammograms Using SVD Subspace Analysis," submitted to the *SPIE International Conference on Computational Imaging*, January 16-20, 2005, San Jose, CA.

5. Yuxin Liu, Josep Prades-Nebot, Paul Salama, and Edward J. Delp, "Performance Analysis of Leaky Prediction Layered Video Coding Using Quantization Noise Modeling," to appear in the *IEEE International Conference on Image Processing (ICIP)*, October 24-27, 2004, Singapore.

6. Yuxin Liu, Paul Salama, Gregory W. Cook, and Edward J. Delp, "Rate-Distortion Analysis of Layered Video Coding by Leaky Prediction," *Proceedings of the SPIE International Conference on Video Communications and Image Processing (VCIP)*, January 18-22, 2004, San Jose, CA, Vol. 5308, pp. 543-554.

7. Yuxin Liu, Bin Ni, Xiaojun Feng, and Edward J. Delp, "LOT-Based Adaptive Image Watermarking," *Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents VI*, January 18-22, 2004, San Jose, CA.

8. Yuxin Liu, Zhen Li, Paul Salama, and Edward J. Delp, "A Discussion of Leaky Prediction Based Scalable Coding," *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, July 6-9, 2003, Baltimore, MD, Vol. 2, pp. 565-568.

9. Yuxin Liu, Christine I. Podilchuk, and Edward J. Delp, "Evaluation of Joint Source and Channel Coding Over Wireless Networks," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, April 6-10, 2003, Hong Kong, Vol. 5, pp. 764-767.

10. Yuxin Liu, Paul Salama, and Edward J. Delp, "Multiple Description Scalable Coding for Error Resilient Video Transmission Over Packet Networks," *Proceedings of the SPIE International Conference on Image and Video Commu-*

nications and Processing (IVCP), January 20-24, 2003, Santa Clara, CA, Vol. 5022, pp. 157-168.

11. Yuxin Liu, Paul Salama, and Edward J. Delp, "Error Resilience of Video Transmission by Rate-distortion Optimization and Adaptive Packetization," *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, August 26-29, 2002, Lausanne, Switzerland, Vol. 2, pp. 613-616.