BACKWARD CHANNEL AWARE DISTRIBUTED VIDEO CODING

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Limin Liu

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

December 2007

Purdue University

West Lafayette, Indiana

To my parents, Binbin Lin and Guojun Liu;

To my grandfather, Changlian Liu;

To my husband, Zhen Li;

And in memory of my grandparents, Yuzhu Ruan and Decong Lin.

ACKNOWLEDGMENTS

I am very grateful to my advisor, Professor Edward J. Delp, for his invaluable guidance and support, for his confidence in me, and for the precious opportunities he had given to me. I wish to express my sincere thanks to his inspiring instructions and broad range of expertise, which have led me to the interesting and charming world of video coding. It has been a great honor to be a part of the Video and Image Processing (VIPER) lab.

I also thank my Doctoral Committee: Professor Zygmunt Pizlo, Professor Mark Smith, and Professor Michael D. Zoltowski, for their advice, encouragement, and insights despite their extremely busy schedules.

I would like to thank the Indiana Twenty-First Century Research and Technology Fund for supporting the research.

I am very fortunate to work with many incredibly nice and brilliant colleagues in the VIPER lab. I appreciate the support and friendship from them. Working with them is one of the joyful highlights during my graduate school life: Dr. Gregory Cook, Dr. Hyung Cook Kim, Dr. Eugene Lin, Dr. Yuxin Liu, Dr. Paul Salama, Dr. Yajie Sun, Dr. Cuneyt Taskiran, Dr. Hwayoung Um, Golnaz Abdollahian, Marc Bosch, Ying Chen, Oriol Guitart, Michael Igarta, Deen King-Smith, Liang Liang, Ashok Mariappan, Anthony Martone, Aravind Mikkilineni, Nitin Khanna, Ka Ki Ng, Carlos Wang, and Fengqing Zhu. I would also like to thank our visiting researchers from abroad for their perspectives: Professor Reginald Lagendijk, Professor Fernando Pereira, Professor Luis Torres, Professor Josep Prades-Nebot, Pablo Sabria, and Rafael Villoria.

I would like to thank Mr. Mike Deiss and Dr. Haoping Yu for offering me summer internship at Thomson Corporate Research in 2005. I am particularly grateful for

the opportunity to design and develop the advanced 4:4:4 scheme for H.264, which was adopted by the Joint Video Team (JVT) and became a new profile in H.264.

I would like to thank Dr. Margaret (Meg) Withgott and Dr. Yuxin (Zoe) Liu for the summer internship at Sun Microsystems Laboratories in 2006. Their guidance and encouragement has been very beneficial. I also thank Dr. Gadiel Seroussi of Mathematical Sciences Research Institute for the discussion on video compression and general coding problems.

I would like to thank Dr. Vadim Sheinin, Dr. Ligang (Larry) Lu, Dr. Dake He, Dr. Ashish Jagmohan, and Dr. Jun Chen for the opportunity to work at IBM T. J. Watson Research Lab as a summer intern in 2007. I enjoyed the numerous and passionate discussions with them. And I miss all my summer intern friends from IBM Research Lab.

I would like to thank all my friends from Tsinghua University. Their friendship along the journey made my undergraduate years a very cherishable memory.

I have dedicated this document to my mother and father, and in memory of my grandparents for their constant support throughout these years. They taught me to think positively and make persistent efforts. I would also thank my parents-in-law, brothers-in-law and sisters-in-law. Their love allows me to experience the joy of a big family.

Finally, I would like to express my sincere appreciation to my husband, Zhen Li, for his patience, understanding, encouragement and love. I could never have gone this far without every bit of his help.

TABLE OF CONTENTS

Page

Page

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

AVC	Advanced Video Coding
BCAME	Backward Channel Aware Motion Estimation
BCAWZ	Backward Channel Aware Wyner-Ziv Video Coding
CABAC	Context-Adaptive Binary Arithmetic Coding
CAVLC	Context-Adaptive Variable-Length Coding
CCITT	International Telegraph and Telephone Consultative Committee
DCT	Discrete Cosine Transform
DISCUS	DIstributed Source Coding Using Syndromes
DVC	Distributed Video Coding
DWT	Discrete Wavelet Transform
EBCOT	Embedded Block Coding with Optimized Truncation
FIR	Finite Impulse Response
GOP	Groups Of Pictures
ISDN	Integrated Services Digital Network
ISO	International Organization for Standardization
ITU	International Telecommunication Union
JVT	Joint Video Team
KLT	Karhunen Lòeve Transform
LDPC	Low Density Parity Check
Mbps	Mbits/sec
MC	Motion Compensation
MCP	Motion-Compensated Prediction
ME	Motion Estimation
MPEG	Moving Picture Experts Group

NDME	Network-Driven Motion Estimation
NSQ	Nested Scalar Quantization
OBMC	Overlapped Block Motion Compensation
PCCC	Parallel Concatenated Convolutional Code
PRISM	Power-efficient, Robust, hIgh-compression, Syndrome-based Multimedia coding
PSNR	Peak Signal-to-Noise Ratio
QP	Quantization Parameter
RSC	Recursive Systematic Convolutional
RVLC	Reversible Variable Length Code
SLEP	Systematic Lossy Error Protection
SWC	Slepian-Wolf coding
TCSQ	Trellis-Coded Scalar Quantization
VCEG	Video Coding Experts Group
VLC	Variable-Length Code
WZVC	Wyner-Ziv Video Coding

ABSTRACT

Liu, Limin Ph.D., Purdue University, December, 2007. Backward Channel Aware Distributed Video Coding. Major Professor: Edward J. Delp.

Digital image and video coding has witnessed rapid development in the past decades. Conventional hybrid motion-compensated-prediction (MCP) based video coding exploits both spatial and temporal redundancy at the encoder. Hence the encoder requires much more computational resources than the decoder. This poses a challenge for applications such as video surveillance systems and wireless sensor networks. Only limited memory and power are available at the encoder for these applications, while the decoder has access to more powerful computational resources. The Slepian-Wolf theorem and Wyner-Ziv theorem have proved that a distributed video coding scheme is achievable where sources are encoded separately and decoded jointly. The basic goal of our research is to analyze the performance of the low complexity video encoding theoretically, and to design new practical techniques to achieve a high video coding efficiency while maintaining low encoding complexity.

In this thesis, we propose a new backward channel aware Wyner-Ziv approach. The basic idea is to use backward channel aware motion estimation to code the key frames in Wyner-Ziv video coding, where motion estimation is done at the decoder and motion vectors are sent back to the encoder. We refer to these backward predictive coded frames as BP frames. A mode decision scheme through the feedback channel is studied. Compared to Wyner-Ziv video coding with INTRA coded key frames, our approach can significantly improve the coding efficiency. We further consider the scenario when there are transmission errors and delays over the backward channel. A hybrid scheme with selective coding is proposed to address the problem. Our results show that the coding performance can be improved by sending more motion vectors

to the encoder. However, there is a tradeoff between complexity and rate-distortion performance in backward channel aware Wyner-Ziv video coding. We present a model to quantitatively analyze the complexity and rate-distortion tradeoff for BP frames. Three estimators, the minimum estimator, the median estimator and the average estimator, are proposed and the complexity-rate-distortion analysis is presented.

1. INTRODUCTION

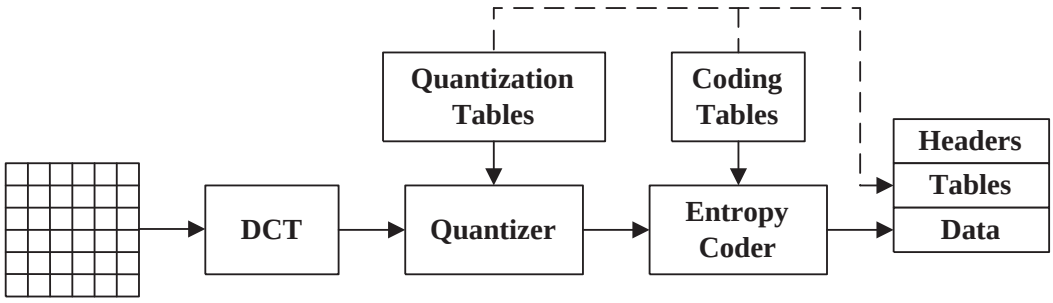
Digital images and videos are everywhere today with a wide range of applications, such as high definition television, video delivery by mobile telephones and handheld devices. Multimedia information is digitally represented so that it can be stored and transmitted conveniently and accurately. However, digital image and video data generally require huge storage and transmission bandwidth. Even with the rapid increase in processor speeds, disk storage capacity and broadband networks, an efficient representation of the image and video signal is needed. Video compression algorithms are used to reduce the data rate of the video signal while maintaining video quality. A typical video coding system consists of an encoder and a decoder, which is referred to as a *codec* [1]. To ensure the inter-operability between different platforms and applications, image and video compression standards have been developed over the years

In this chapter, we first provide an overview of the image and video coding standards. In particular, we describe the current video coding standard H.264 in detail. We then discuss on-going research within the video coding standard community, and give an overview of the recent advances in video coding and their potential applications.

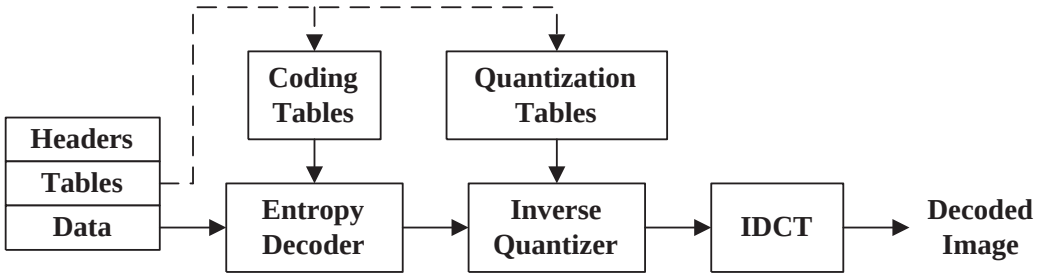
1.1 Overview of Image and Video Coding Standards

1.1.1 Image Coding Standards

JPEG (Joint Photographic Experts Group) [2–6] is a group established by members from both the International Telecommunication Union (ITU) and the International Organization for Standardization (ISO) to work on image coding standards.



(a) encoder



(b) decoder

Fig. 1.1. Block Diagram of a DCT-Based JPEG Coder

The JPEG Standard

JPEG specifies a still image coding process and the file format of the bitstream. An input image is first divided into non-overlapping blocks of size 8×8. Each block is transformed into the frequency domain by the DCT transform coding, followed by the quantization of the DCT coefficients and entropy coding. Fig. 1.1(a) shows a block diagram of a Discrete Cosine Transform (DCT)-based JPEG encoder. The process is repeated for the three color components for color images. The decoding process is the inverse operation of the encoding process in an order that is the reverse of that of the encoder. Fig. 1.1(b) shows the block diagram of the DCT-based JPEG decoder.

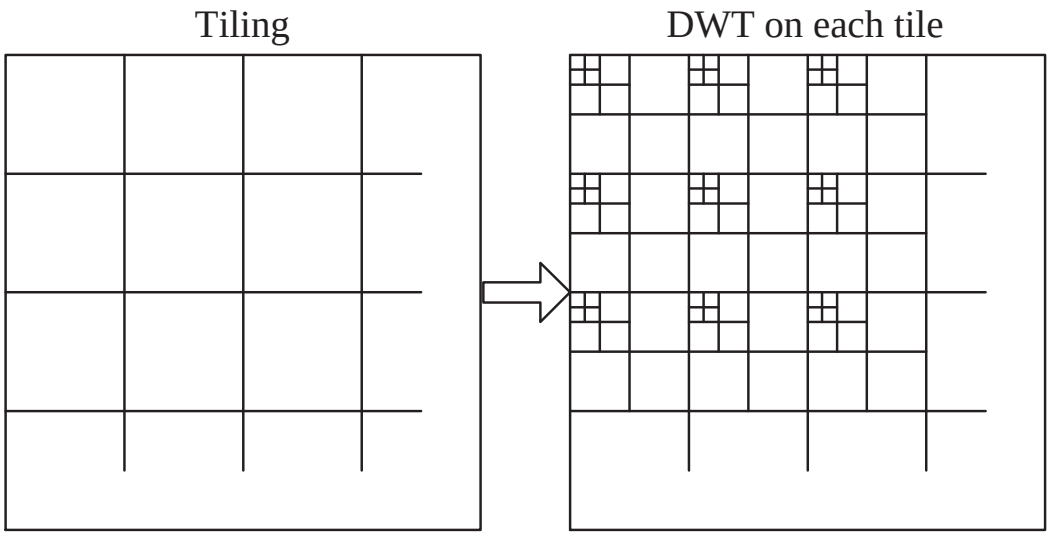


Fig. 1.2. Discrete Wavelet Transform of Image Tile Components

The JPEG 2000 Standard

JPEG 2000 [4, 7, 8] is a wavelet-based image compression standard. It provides superior low data rate performance. JPEG 2000 also provides efficient scalability such as rate scalability and resolution scalability, which allows a decoder to decode and extract information from part of the compressed bit stream.

The main processing blocks of JPEG 2000 include transform, quantization and entropy coding. First the input image is decomposed into components that are handled separately. There are two possible choices: the YCrCb domain and the YUV domain. The input component is then divided into rectangular non-overlapping tiles, which are processed independently as shown in Fig. 1.2. The use of discrete wavelet transform (DWT) instead of DCT as in JPEG is one of the major differences between JPEG and JPEG 2000. The tiles can be transformed into different resolution levels to provide Region-of-Interest (ROI) coding. Before entropy coding, the transform coefficients are quantized within the subband. Arithmetic coding is used in JPEG 2000.

1.1.2 Video Coding Standards

Since the early 1990s, a series of video coding standards have been developed to meet the growing requirements of video applications. Two groups have been actively involving in the standardization activities: the Video Coding Experts Group (VCEG) and the Moving Picture Experts Group (MPEG). VCEG is working under the direction of the ITU Telecommunication Standardization Sector (ITU-T), which is formerly known as International Telegraph and Telephone Consultative Committee (CCITT). This group typically works on the standard with the names "H.26x". MPEG carries out the standardization work under ISO/IEC and labels them as "MPEG-x". In this section, we briefly review the standards in the chronological order and describe the latest standard H.264 in details.

H.261

H.261 was approved in 1991 for video-conferencing systems and video-phone services over the integrated services digital network (ISDN) [9] [10]. The target data rate is at multiples of 64 kbps. The H.261 standard has mainly two modes: the INTRA and INTER modes. The INTRA mode is basically similar to the JPEG compression (Section 1.1.1) where DCT-based block transform is used. For the INTER mode, motion estimation (ME) and motion compensation (MC) were first adopted. The motion search resolution adopted in H. 261 is integer-pixel accuracy. For every 16×16 macroblock, one motion vector is chosen from a search window centered by the original pixel position.

MPEG-1

MPEG-1 was started in 1988 and finally approved in 1993. The main application of MPEG-1 is for storage of video data on various digital storage media such as CD-ROM. MPEG-1 added more features to H.261. The comparison of the H.261 and

Table 1.1
Comparison of the H.261 and MPEG-1 standards

H.261	**MPEG-1**
Sequential access	Random access
One basic frame rate	Flexible frame rate
CIF and QCIF images only	Flexible image size
I and P frames	I, P and B frames
MC over 1 frame	MC over 1 or more frames
1 pixel MV accuracy	1/2 pixel MV accuracy
Variable threshold + uniform quantization	Quantization matrix
Slice structure	Uses groups of pictures (GOP)

MPEG-1 standards are shown in Table 1.1 [9]. A significant improvement is the introduction of bi-directionally prediction in MPEG-1, where both the previous and next reconstructed frames can be used as reference frames.

MPEG-2/H.262

MPEG-2, also known as H.262, was developed as a joint effort between VCEG and MPEG. MPEG-2 aims to serve a variety of applications, such as DVD video and digital television broadcasting. The key features are:

- MPEG-2 accepts not only progressive video, but also interlaced video. It also adds a new macroblock size of 8×16.
- MPEG-2 provides a scalable bitstream. The syntax allows more than one layer of video. Three formats of scalability are available in MPEG-2: spatial scalability, temporal scalability and SNR scalability.
- MPEG-2 features more options in the quantization and coding steps to further improve the video quality.

H.263

H.263 [11], and its extensions, known as H.263+ [12] and H.263++, share many similarities with H. 261 with more coding options. It was originally developed for low data rate but eventually extended to an arbitrary data rate. It is widely used in the video streaming applications. With the new coding tools, H.263 can achieve similar video quality as H.261 with roughly half the data rate or lower. The motion search resolution specified in H.263 is half-pixel accuracy. The quantized DCT coefficients are coded with 3-D Variable Length Coding (last, run, level) instead of a 2-D Variable Length Coding (run, level) and the end-of-block marker. In the advanced prediction mode, there are two important options which result in significant coding gains:

- H.263 allows four motion vectors per macroblock and the one/four vectors decision is indicated in the macroblock header. Despite using more bits on the transmission of motion information, it gives more accuracy prediction and hence results in smaller residual entropy.

- Overlapped block motion compensation (OBMC) is adopted to reduce the blocking artifacts. The blocks are overlapped quadrant-wise with the neighboring blocks. In H. 263 Annex F, each pixel is predicted by a weighted sum of three prediction values obtained from three motion vectors. OBMC provides improved prediction accuracy as well as better subjective quality in video coding, with increased computational complexity.

MPEG-4

The MPEG-4 standard [13] includes the specifications for Systems, Visual, and Audio. It has been used in several areas, including digital television, interactive graphics applications, and multimedia distribution over the networks. MPEG-4 enables object-based video coding by coding the contents independently. A scene is composed of several Video Objects (VOs). The information of shape, texture, shape

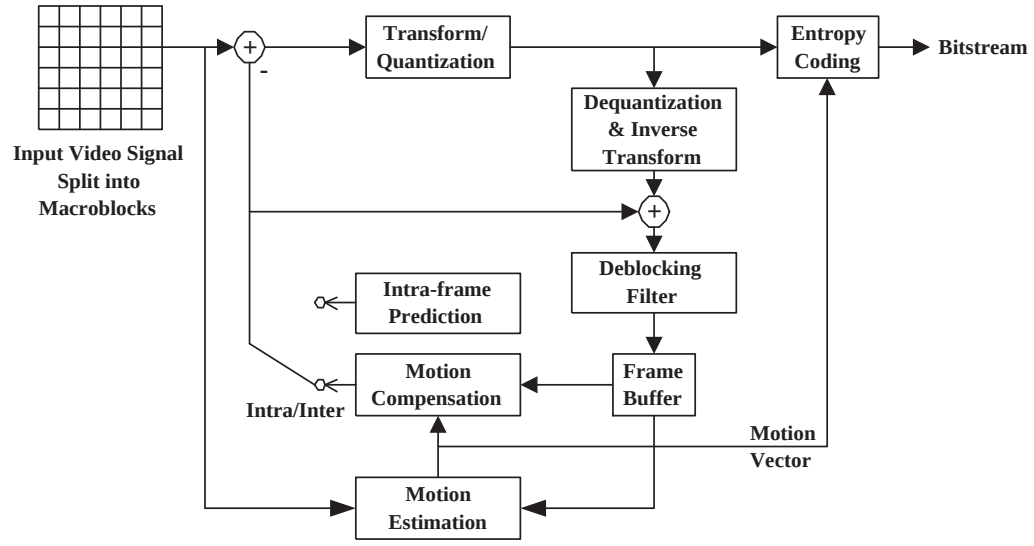
motion, and texture motion is extracted by image analysis and coded by parameter coding. MPEG-4 also supports mesh, face and body animation. Its provides various coding tools for the scalability of contents. The Fine Granular Scalability (FGS) Profile allows the adaptability to bandwidth variation and resilience to packet losses. MPEG-4 also incorporates several error-resilience tools [14,15], which achieves better resynchronization, error isolation and data recovery. NEWPRED mode is a new error resilience tool adopted in MPEG-4. It allows the encoder to update the reference frames adaptively. A feedback message from the decoder identifies the lost or damaged segments and the encoder will avoid using them as further reference.

H.264

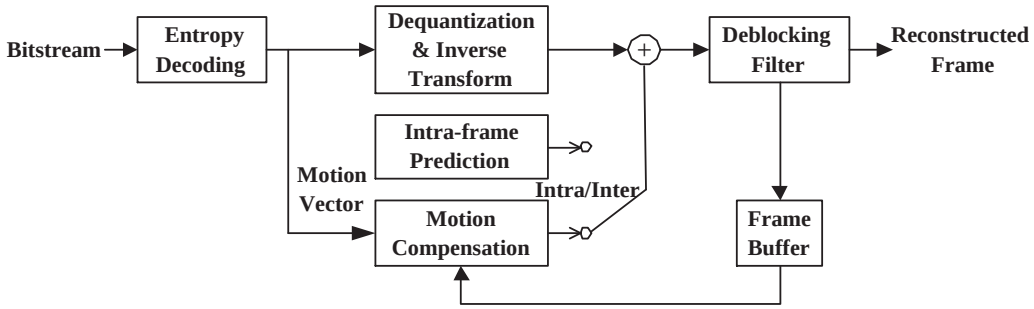
After the development of H.263 and MPEG-4, a longer term video coding project, known as H.26L was set up. It further evolved into H.264, which was approved as a joint coding standard of both ISO and ITU-T in 2004.

A video coding standard generally defines only syntax of the decoder, which provides flexibility for encoder optimization. All these standards (Section 1.1.2) are based on block-based hybrid video coding as shown in Fig. 1.3. More sophisticated coding methods, such as highly accurate motion search and better texture models, lead to the advances in video compression.

In H.264, the input video frame is divided into slices, which is further divided into macroblocks, and each macroblock is processed independently. The basic coding unit of the encoder and decoder is a fixed-size macroblock which consists of 16×16 samples of the luma component and 8×8 samples of the chroma components for the 4:2:0 format. The macroblock can be further divided into small blocks. A sequence of macroblocks is organized to form the slice in the raster scan order, which represents a region in the picture that can be decoded independently. For example, a picture may contain three slices as shown in Fig. 1.4 [1] [16] . Each slice is self-contained, which means each slice could be decoded without the information from the other slices.



(a) Encoder



(b) Decoder

Fig. 1.3. A Hybrid Motion-Compensated-Prediction Based Video Coder (H.264)

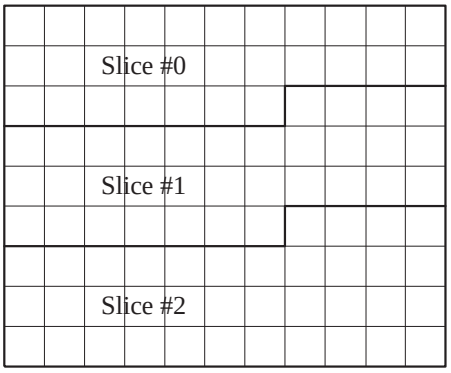


Fig. 1.4. Subdivision of a Picture into Slices

The basic block diagram of the H.264 encoder is shown in Fig. 1.3-(a). First the current block is predicted from the previously coded spatially or temporally neighboring blocks. If INTER coding is selected, a motion vector is obtained to rebuild the current block using motion compensation, where the two-dimensional motion vector represents the displacement between the current block and its best matching reference block. The prediction error is transformed by DCT (or integer transform in H. 264) to reduce the statistical correlation. The transform coefficients are quantized by a predefined quantization table, where a quantization parameter controls the step size of the quantizer. Generally one of the following two entropy coding methods is used for the quantized coefficients: variable-length code (VLC) and arithmetic coding. The in-loop deblocking filter is used to reduce "blocking" artifacts while maintaining the sharpness of the edges across block boundaries. It enables the reduction in the data rate and improves the subjective quality.

There are five slice types, where the first three are similar to the previous standards and the last two are new [1] [16] [17]:

- **I slice**: All macroblocks in the slice only uses INTRA prediction.
- **P slice**: The macroblocks in the slice use not only INTRA prediction, but also INTER prediction (temporal prediction) with only one motion vector per block.
- **B slice**: Two motion vectors are available for every block in the slice. A weighted average of the pixel values form the motion compensated prediction.
- **SP slice and SI slice**: Switching P slice and switching I slice provide exact switching between different video streams, as well as random access, fast forward and reverse. The difference between these two types is that SI slice uses only INTRA prediction and SP slice uses INTER prediction as well.

The main difference between I slice and P/B slice is that temporal prediction is not employed in I slice. Three sizes are available for INTRA prediction: 16×16, 8×8 and 4×4. Fig. 1.5 shows the prediction from different directions, where 16

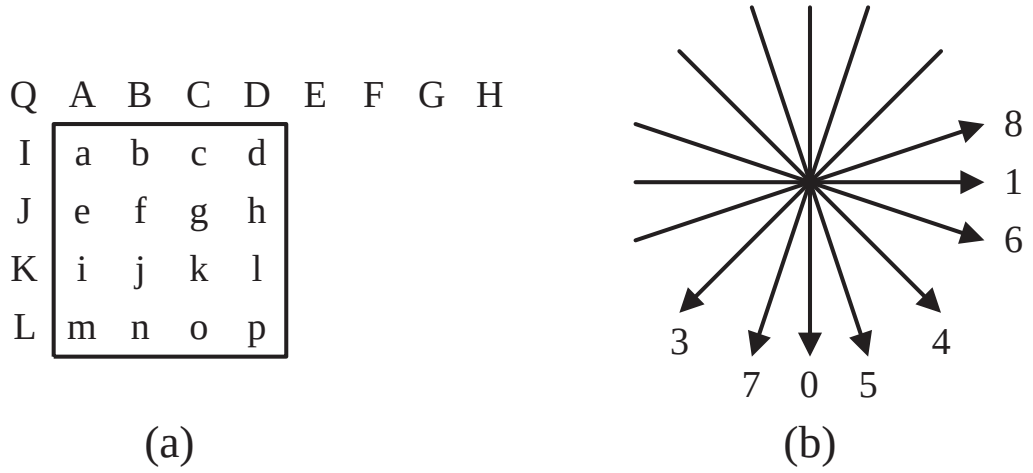


Fig. 1.5. (a) INTRA_4 × 4 Prediction (b) Eight "Prediction Directions"

samples denoted by a through p are predicted by the neighboring samples denoted by A through Q. Eight possible directions of prediction are illustrated excluding the "DC" prediction mode. Fig. 1.6 lists five of the nine INTRA_4 × 4 modes corresponding to the directions in Fig. 1.5. In mode 0 (vertical prediction), the samples above the top row of the 4 × 4 block are copied directly as the predictor, which is illustrated by the arrows. Similarly, mode 1 (horizontal prediction) uses the column to the left of the 4 × 4 block as the predictor. For mode 2 (DC prediction), the average of the adjacent samples are taken as the predictor. The other six modes are the diagonal prediction modes. They are known as diagonal-down-left, diagonal-down-right, vertical-right, horizontal-down, vertical-left, and horizontal-up prediction respectively. In INTRA_16 × 16 mode, there are only four modes available: vertical, horizontal, DC, and plane predictions. For the first three, they are similar to the corresponding modes in INTRA_4×4. The plane prediction uses the linear combinations of the neighboring pixels as the predictor.

As mentioned above, the previous standards use DCT for transform coding. In H.264, a separable integer transform is used. The integer transform is an approxi-

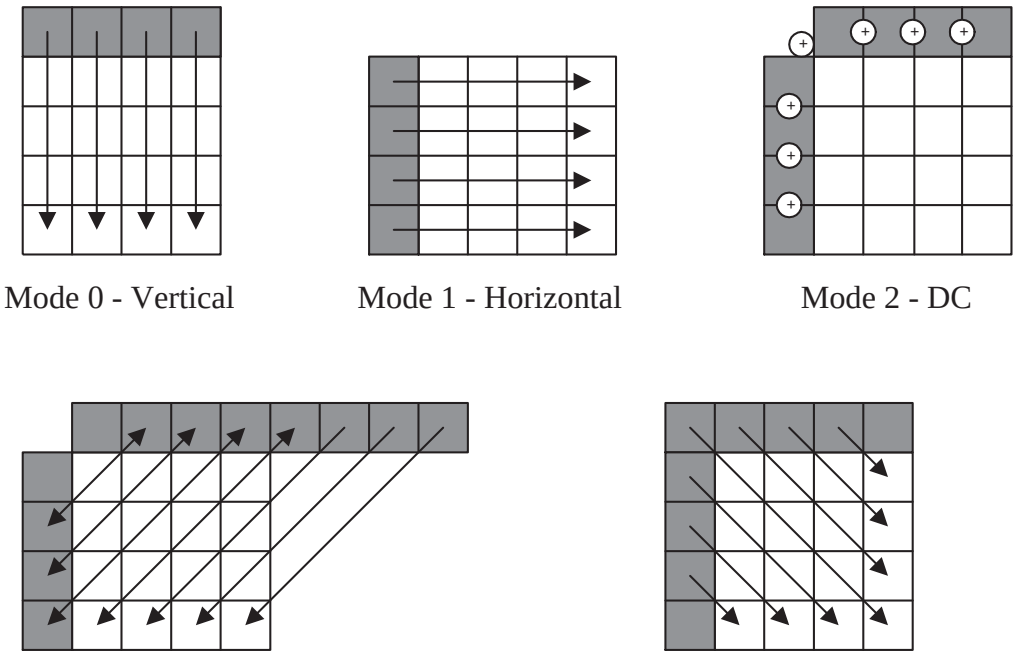


Fig. 1.6. Five of the Nine INTRA_4 × 4 Prediction Modes

mation of DCT transform that uses integer arithmetic. The basic matrix is designed as:

$$T_{4\times4} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \tag{1.1}$$

The matrix is very simple and only a few additions, subtractions, and bit shifts are needed for implementation. Moreover, the mismatches due to floating point numbers between the encoder and the decoder in the DCT transform are avoided.

Two entropy coding methods are supported in H.264 in a context adaptive way: context-adaptive variable-length coding (CAVLC) and context-adaptive binary arithmetic coding (CABAC). CABAC achieves higher coding efficiency by sacrificing complexity. It allows a non-integer number of bits per symbol while CAVLC supports only an integer number of bits per symbol. In general CABAC uses $10\% - 15\%$ less bits than CAVLC. In CAVLC, the codes that occur more frequently are assigned shorter codes and vice versa. In CABAC, context modeling is first used to enable the choice of a suitable model for each syntax element. For example, transform coefficients and motion vectors belong to different models. Similar to a Variable Length Coding (VLC) table, a specified binary tree structure is then used to support the binarization process. Finally a context-conditional probability estimation is employed.

In P slices, temporal prediction is used and motion vectors are estimated between pictures. Compared to the previous standards, H.264 allows more partition sizes that include 16×16, 16×8, 8×16, or 8×8. When 8×8 macroblock partition is chosen, additional information is used to specify whether it is further partitioned to 8×4, 4×8, or 4×4 sub-macroblocks. The possible partitioning results are shown in Fig. 1.7, where the index of the block shows the order of the coding process. For example, as shown in Fig. 1.8, the macroblock is partitioned into four blocks of size 8×16, 8×8, 4×8, and 4×8 respectively. Each block is assigned one motion vector and four motion vectors are transmitted for this P macroblock.

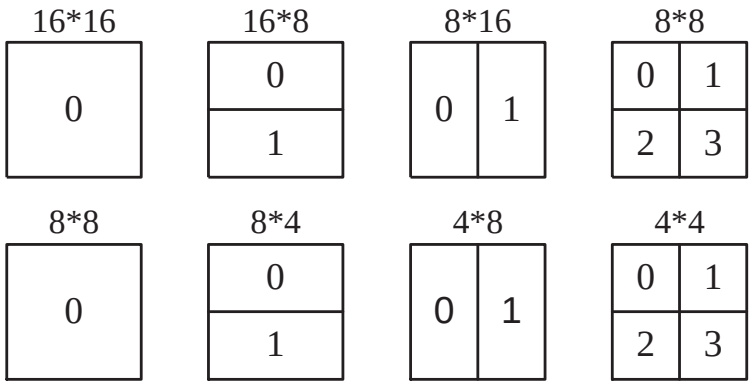


Fig. 1.7. Segmentation of the Macroblock for Motion Compensation

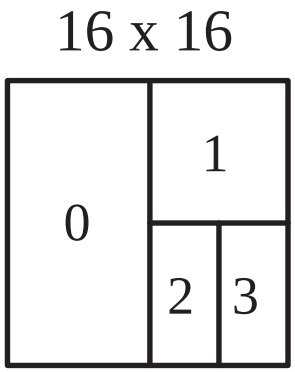


Fig. 1.8. An Example of the Segmentation of One Macroblock

As in the previous standards, H.264 supports fractional accuracy motion vectors. If the motion vector points to integer samples, the prediction value is the corresponding pixel value of the reference picture. If the motion vector points to fractional positions, the prediction values at half-sample positions are obtained by a one-dimensional 6-tap Finite Impulse Response (FIR) filter horizontally and vertically. The prediction values at quarter-sample positions are obtained by the average of the pixel values at the integer and half-sample positions. Fig. 1.9 shows the positions of the pixels, where gray pixels are integer-sample positions. To obtain the half-sample pixels b and h, two intermediate values b_1 and h_1 are derived using a 6-tap filter:

$$b_1 = E - 5F + 20G + 20H - 5I + J \tag{1.2}$$

$$h_1 = A - 5C + 20G + 20M - 5R + T \tag{1.3}$$

Then the values are clipped to $0 - 255$:

$$b = (b_1 + 16) >> 5 \tag{1.4}$$

$$h = (h_1 + 16) >> 5 \tag{1.5}$$

The half-sample pixels m and s are obtained in a similar way. The center half-sample pixel j is derived by:

$$j = (cc - 5dd + 20h_1 + 20m_1 - 5ee + ff + 512) >> 10 \tag{1.6}$$

where the intermediate values cc, dd, m_1, ee, and ff are derived similar to b_1. The samples at quarter-pixel positions are obtained as the average of the neighboring samples either in the vertical or diagonal directions:

$$a = (G + b + 1) >> 1 \tag{1.7}$$

$$e = (b + h + 1) >> 1 \tag{1.8}$$

In H.264, multiple pictures are available in the reference buffer [16] as shown in Fig. 1.10. The number of reference buffers is specified in the header. The weighted prediction using multiple previously decoded pictures significantly outperforms prediction with only one previous decoded picture [18].

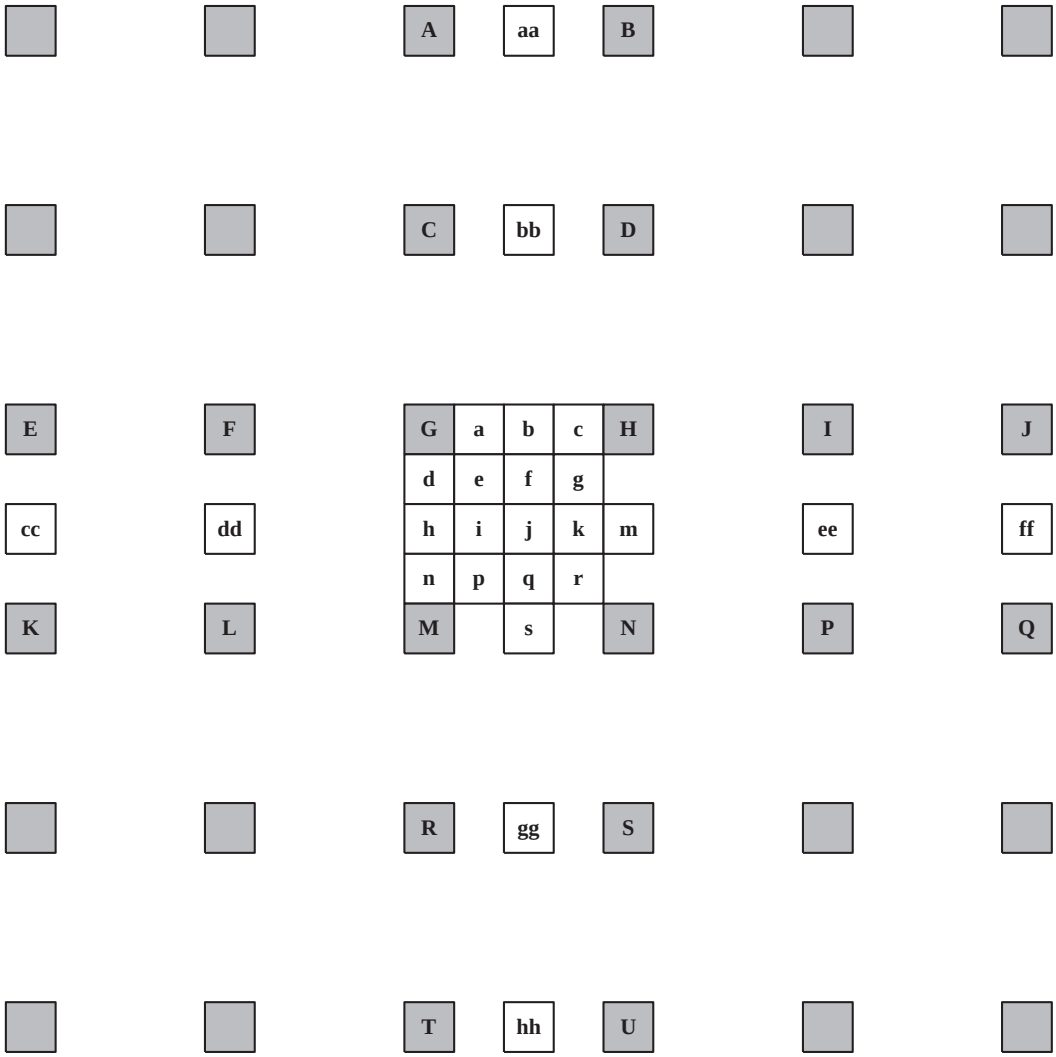


Fig. 1.9. Filtering for Fractional-Sample Accurate Motion Compensation

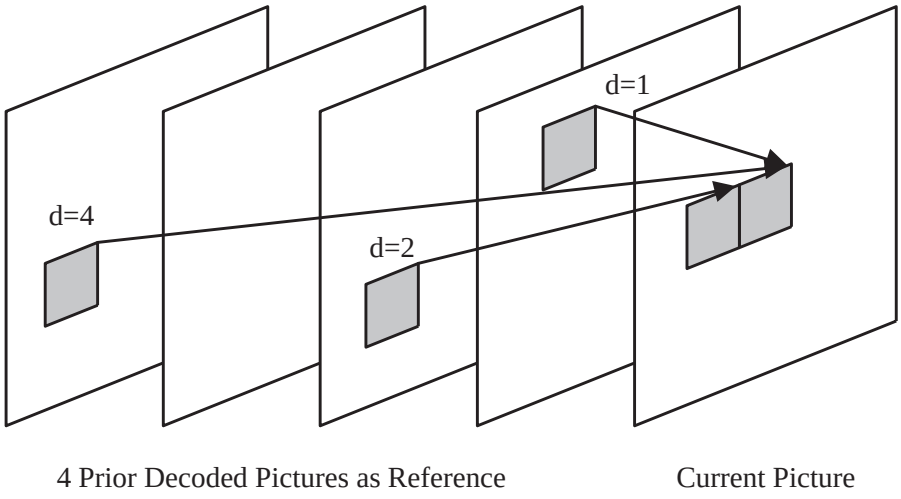


Fig. 1.10. Multiframe Motion Compensation

The usage of B slices significantly improves the coding efficiency. A main feature of a B slice is that there are two motion vectors available for motion compensation. Therefore, B slices organize two distinct lists of reference pictures, List 0 and List 1. The standard provides four modes for the prediction of B slices: List 0, List 1, bi-predictive, and direct modes. For List 0 or List 1 modes, the reference blocks are derived only from List 0 or List 1. In the bi-predictive mode, a weighted combination of the reference blocks from List 0 and List 1 are formed as the predictor. In the direct mode, the motion vector is not derived by motion search but by scaling the available motion vectors of the co-located macroblock in another reference picture. As a new feature, H.264 allows a B slice as the reference for its following pictures.

In summary, H.264 incorporates a collection of the state-of-the-art video coding tools. Here are some important improvements compared to the previous standards [16]:

- improved motion-prediction techniques;
- adoption of a small block-size exact-match transform;
- application of adaptive in-loop deblocking filter;
- employment of the advanced entropy coding methods.

These new features could reduce the data rate by approximately 50% with similar perceptual quality in comparison to H.263 and MPEG-4 [1]. The enhanced performance of H.264 presents a promising future for video applications.

1.2 Recent Advances in Video Compression

With the success of the various video coding standards based on hybrid motion compensated methods, the research community has also been investigating new techniques that can address next-generation video services [5, 19]. These techniques provide higher robustness, interactivity and scalability. For instance, spatial and temporal models for texture analysis and synthesis have been developed to increase the

coding efficiency for video sequences containing textures [20]. Context-adaptive algorithms for intra prediction and motion estimation are proposed in [21] [22]. An algorithm to identify regions of interests (ROIs) in home video is proposed in [23]. A low bit-rate video coding approach is presented in [24] which uses modified adaptive warping and long-term spatial memory. This section describes some recent advances in video coding. Distributed video coding (DVC) is a new approach that reduces the complexity of the encoder and has potential applications in video surveillance and error resilience. Scalable video coding provides flexible adaptation to network conditions. We also give an overview on multi-view and 3D video compression techniques.

1.2.1 Distributed Video Coding

Motion-compensated-prediction (MCP) based video coding systems are highly asymmetrical since the computationally intensive motion prediction is performed in the encoder. Generally, the encoder is approximately $5 - 10$ times more complex than the decoder [25]. It satisfies the needs in applications such as video streaming, broadcast system, and Digital Versatile Disk (DVD), where power constraints are less a concern at the encoder. However, new applications with limited access to power, memory, computational resources at the encoder have difficulties using the conventional video coding systems. For example, in the wireless sensor networks, the nodes are typically not able to be recharged during the mission. Therefore, a simple encoder with low complexity is needed.

Distributed coding is one method to achieve low complexity at the encoder. In distributed coding, source statistics are exploited at the decoder and the encoder can be simplified. The theoretical basis for the problem dates back to two theorems in the 1970s. Slepian and Wolf proved a theorem to address lossless compression [26]. Wyner and Ziv extended the results to the lossy compression case [27]. Therefore, low complexity distributed video encoding approaches are sometimes also referred to

as Wyner-Ziv video coding. The two theorems are described in detail in Section 2.1.1 and Section 2.1.2 respectively.

Since these theoretic results were revisited in the late 1990s, several methods have been developed to achieve the results predicted in these two theorems [28–37]. They are generally based on the channel coding techniques, for example turbo codes [38–41] and low-density-parity-check (LDPC) codes [42–45]. Generation of side information at the decoder is essential to the performance of the system and various ways to improve it are studied in the literature. A hierarchical frame dependency structure was presented in [29] where the interpolation of the side information is done progressively. A weighted vector median filter of the neighboring motion vectors is used to eliminate the motion outliers in [46, 47]. Side information generated by a 3D face model is combined with block-based generated side information to improve the quality of the side information [48]. A refined side estimator that iteratively improves the quality of the side information has been proposed by different research groups [35, 49–51]. Even though the distributed video coding approaches are still far from mature, the research community has identified several potential applications that range from wireless cameras, surveillance systems, distributed streaming and video conferencing to multiview image acquisition and medical image processing [52]. In this section we give an introduction to the basis codec design for distributed video coding as well as some applications.

Distributed Source Coding Using Syndromes (DISCUS)

Distributed Source Coding Using Syndromes (DISCUS) [53] is one of the pioneer works related to distributed source coding. Before describing the design of DISCUS [53], it is beneficial to introduce an example to give a brief idea of the basic concept. Suppose X and Y are correlated binary words. The encoder has only the information

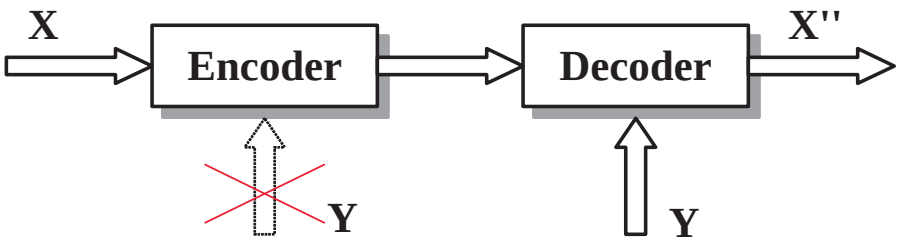


Fig. 1.11. Side Information in DISCUS

of X and the side information Y is available at the decoder as shown in Fig. 1.11. The problem is formulated as:

$$X, Y \in \{0,1\}^n \qquad d_H(X,Y) \leq t \tag{1.9}$$

where $d_H(X,Y)$ represents the Hamming distance between codewords X and Y. The encoder sends the index of the coset which X belongs to. The binary linear code is appropriately chosen with parameter (n,k), where n is the output code length and k is the input message length. The estimate of X is chosen by the index of the coset and side information Y. For example, suppose X and Y are 3-bit binary words with equal probability, i.e., $P(X=x)=\frac{1}{8}$. If we have no information of Y, 3 bit is required to encode X. Now the decoder observes Y and has the prior knowledge of the correlation condition that the Hamming distance between X and Y is not greater than one. Four coset sets are established where the Hamming distance between the two codewords consisted of every coset is 3, i.e., four coset sets are $\{000,111\}, \{001,100\}, \{010,101\}$, and $\{100,011\}$. A 2-bit code is required to send the index of the coset. Assume the index received at the decoder is the third one, i.e., X may be 010 or 101. Since the decoder observes Y is 011, X must be 010 because of the correlation condition. With side information, the length of the codeword is reduced to 2. The results can be generalized to the continuous-valued source X and side information Y.

PRISM

PRISM (Power-efficient, Robust, hIgh-compression, Syndrome-based Multimedia coding) is a distributed video coding architecture proposed in [34,54,55]. It addresses the problem of drift due to prediction mismatch. For video coding, the current macroblock is regarded as X in the distributed source coding and the best predictor for X from the previous frame is regarded as side information Y.

The block diagrams of the PRISM encoder and decoder are shown in Fig. 1.12 and Fig. 1.13 respectively. The input video source is first partitioned into 16×16 or 8×8 blocks and blockwise DCT transformed. The DCT coefficients are arranged using the zigzag scan. Since the first few coefficients carry most of the important information about the frames, only about 20% of the DCT coefficients are encoded using syndrome codes. Simple trellis codes are chosen for small block-lengths. Refinement quantization is used after the syndrome coding. The rest of the DCT coefficients are coded as conventional INTRA frames. Hence, no motion search is performed at the encoder which ensures the simplicity of the encoder. At the decoder, motion search is used to generate side information for the syndrome decoding. The decoder uses Viterbi algorithm for the decoding.

With the encoder and decoder structures as shown in Fig. 1.12 and 1.13, PRISM has the following features:

- low encoding complexity and high decoder complexity: The complexity of the encoder is comparable to the motion JPEG. The complexity is shifted to the decoder due to the motion search operations.

- robustness: Since the frames are self-contained encoded, there is no drift problem existing here. The use of channel codes also improves the inherent robustness. The step size of the base quantizer can be continuously tuned to achieve a specific target data rate.

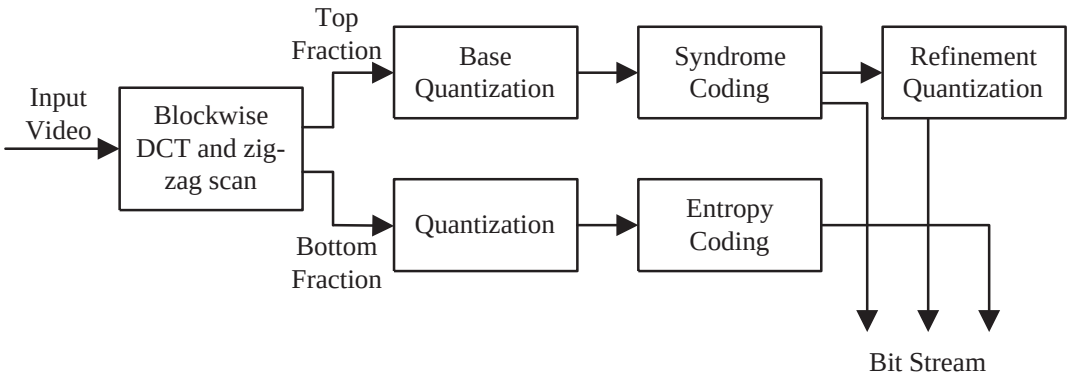


Fig. 1.12. Block Diagram of the PRISM Encoder

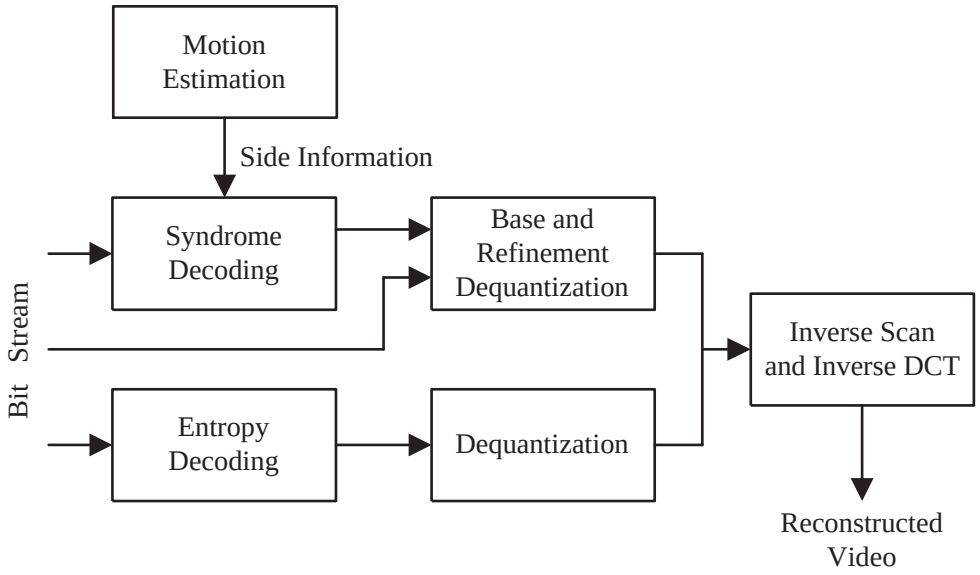


Fig. 1.13. Block Diagram of the PRISM Decoder

Wyner-Ziv Video Coding for Error-resilient Compression

A problem with motion-compensated prediction (MCP) based video coder is the predictive mismatch between the reference frames at the encoder and the decoder when there is an error during transmission. We also refer this scenario as the drift problem. Drift errors will propagate through the subsequent frames until an INTRA frame is sent and lead to significant quality degradation. In [56], an error resilient method is proposed using Wyner-Ziv video coding by periodically transmitting a small amount of additional information to prevent the propagation of errors.

The problem of predictive coding is re-formulated as a variant of the Wyner-Ziv problem in [56, 57]. Denote two successive symbols to be encoded as x_{n-1} and x_n. Assume $\tilde{x}_{n-1}$ is the reconstruction of x_{n-1} at the decoder which is possibly erroneous. The problem is formulated as coding the symbol x_n with the side information $\tilde{x}_{n-1}$.

The additional information sent by the encoder is termed as "coset information." The frame including the coset information is denoted as "peg frames." The encoder sends both the residual frame and the coset index for the "peg frame." The generation of the coset information is described by the following. First, a forward transform is used on the 4×4 block. Then the transform coefficients are quantized. The LDPC (Low- Density-Parity-Check) encoding is done on each bit-plane of the transform-domain coefficients. Forward error correction (FEC) is used to protect the peg information. All the non-peg frames are coded by H.26L. The proposed approach demonstrates the capability of error correcting in the event of channel loss [56, 57].

Systematic Lossy Error Protection

In systematic lossy source-channel coding [58], a digital channel is added as an enhancement layer in addition to the original analog channel. The Wyner-Ziv paradigm is employed as forward error protection for systematic lossy source-channel coding [30,59–61]. The system diagram shown in Fig. 1.14 is referred to as systematic lossy error protection (SLEP). The main video coder is regarded as the systematic

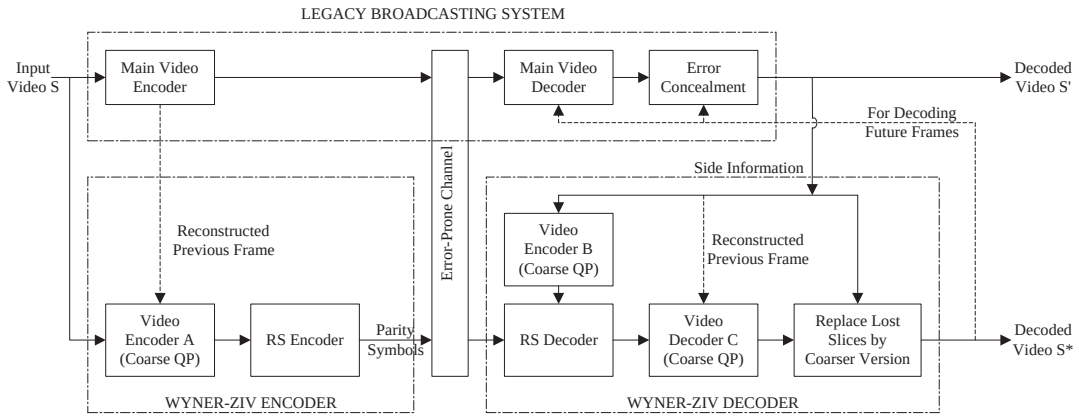


Fig. 1.14. Systematic Lossy Error Protection (SLEP) by Combining Hybrid Video Coding and RS Codes

part of the system. If there is no transmission error, the Wyner-Ziv bitstream is redundant. When the signal transmitted from the main video encoder encounters channel error, the Wyner-Ziv coder serves as the augmentation information. In this situation, the video encoder A codes the input video S by a coarser Quantization Parameter (QP). The bitstream is sent through the Reed-Solomon (RS) coder and only the parity bits of the RS coder is transmitted to the decoder. The mainstream reconstructed signal at the decoder S' is coded by a video encoder B which is identical to the video encoder A. The output serves as the side information for the decoder of the RS bitstream. After the video decoder C, the reconstructed video signal replaces the corrupted slices by the coarser version. Hence the system prevents large errors due to error-prone channel. The proposal outperforms the traditional forward error correction (FEC) when the symbol error rate is high. An extension of this approach is presented in [61] where unequal error protection for the bitstream is used to further improve error resilience performance. The significant data elements, such as motion vectors, are provided with more parity bits than low priority data elements, such as transform coefficients.

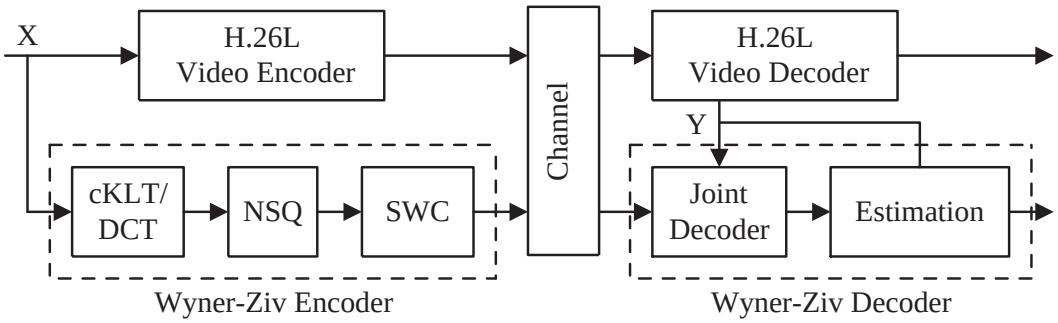


Fig. 1.15. Block Diagram of the Layered Wyner-Ziv Video Codec

Layered Wyner-Ziv Video Coding

A layered Wyner-Ziv video codec was proposed in [62, 63] which also achieves scalability. It can encode the source once and decode the same bitstream at different lower rates. The proposed system meets the requirements of unpredictable variations of the bandwidth. Fig. 1.15 shows the block diagram of the layered Wyner-Ziv video codec. The H.26L video coder is treated as the base layer and the bitstream from the Wyner-Ziv video coder is considered as the enhancement layer. Three components are included in the Wyner-Ziv encoder: DCT transform, nested scalar quantization (NSQ), and Slepian-Wolf coding (SWC). NSQ partitions the DCT coefficients into cosets and output the indices of the cosets. Multi-level LDPC code is employed to design SWC. Every bit plane is associated with a portion of the coefficients, where the most significant bit plane is assigned as the first bit plane. Each extra bit plane is regarded as an additional enhancement layer. The decoder recovers the bit planes sequentially starting with the first plane. Every extra bit plane that the decoder receives improves the decoding quality.

1.2.2 Scalable Video Coding

Scalable video coding provides flexible adaptation to heterogeneous network conditions. The source sequence is encoded once and the bitstream can be decoded partially or completely to achieve different quality. The base layer provides the basic information of the sequence and each enhancement layer can be added to improve the quality incrementally.

Research on scalable video coding has been going on for about twenty years [64]. The rate distortion analysis of scalable video coding are extensively studied in [65–68]. Because of the inherent scalability of the wavelet transform, wavelet based video coding structure have been exploited. A fully rate scalable video codec, known as SAMCoW (Scalable Adaptive Motion Compensated Wavelet) was proposed in [69]. A 3-D wavelet transform with MCTF (Motion Compensated Temporal Filtering) [70] has been developed. On the other hand, a design based on hybrid video coding is standardized. MPEG-2 was the first video coding standard to introduce the concept of scalability [70]. The scalable extension of H.264 [64] is a current standardization effort supporting temporal, spatial, and SNR scalability. Compared to the state-of-the-art single layer H.264, the scalable extension has a small increase in the decoding complexity. Spatial and SNR scalability generally have a negative impact on the coding performance depending on the encoding parameters. In the following, we give an overview of the concept of temporal, spatial, and SNR scalability respectively.

Temporal Scalability

Temporal scalability allows the decoding at several frame rates from a bitstream. As shown in Fig. 1.16 temporal scalable video coding can be generated by a hierarchical structure. The first row index $T_i(i = 0, 1, 2, 3)$ represents the index of the layers, where T_0 is the base layer and $T_i(i = 1, 2, 3)$ are the enhancement layers. The second row index denotes the coding order where the frames of the lower layer are coded before the neighboring frames with higher layers. If only T_0 layer is decoded,

it achieves 7.5 frames per second for the sequence, adding T_1 layer can produce a sequence with 15 frames per second. The frame rate can be further increased to 30 frames per second by decoding T_2 layer. The hierarchical structure shows an improved coding efficiency especially with cascading quantization parameters. The base layer is encoded with high fidelity following lower quality coded enhancement layers. Even though the enhancement layers are generally coded as B frames, they can be coded as P frames to reduce the delay with a cost in the coding efficiency.

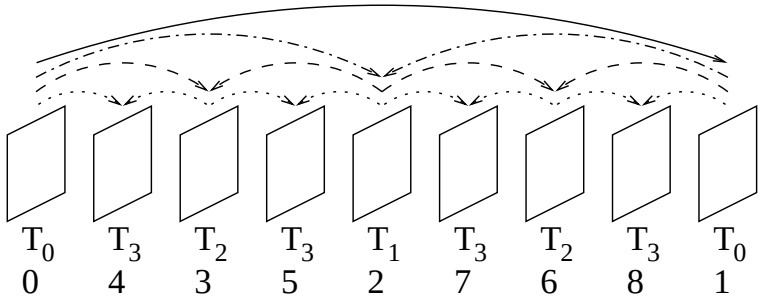


Fig. 1.16. Hierarchical Structure of Temporal Scalability

Spatial Scalability

In spatial scalability, each layer corresponds to a specific resolution. Besides the prediction for single-layer coding, spatial scalable video coding also exploits the inter-layer correlation to achieve higher coding efficiency. The inter-layer prediction can use the information from the lower layers as the reference. This ensures that a set of layers can be decoded independent of all higher layers. The restriction of the prediction results in lower coding performance than single-layer coding at highest resolution.

The inter-layer correlation is exploited in several aspects [71]. Inter-layer intra texture prediction uses the interpolation of the lower layer as the prediction of the current macroblock. The borders of the block from the lower layer are extended before applying the interpolation filter. Two modes are available in inter-layer motion

prediction: the base layer mode and the quarter pel refinement mode. For the base layer mode, no additional motion header is transmitted. The macroblock partitioning, the reference picture indices, and the motion vectors of the lower layer are copied or scaled to be used in the current layer. For the quarter pel refinement mode, a quarter-pixel motion vector refinement is transmitted for a refined motion vector. A flag is sent to signal the use of inter-layer residual prediction, where the residual signal of the lower layer is upsampled as the prediction and only the difference is transmitted.

SNR Scalability

Two concepts are used in the design of SNR scalable coding: coarse-grain scalability (CGS) and fine-grain scalability (FGS) [64, 70, 72, 73]. In CGS, SNR scalability is achieved by using similar inter-layer prediction techniques as described in Section 1.2.2 without the interpolation/upsampling. It can be regarded as a special case of spatial scalability which has the identical frame resolution through the layers. CGS is characterized by its simplicity in design and low decoder complexity. However, it lacks flexibility in the sense that no fine tuning of the SNR points is achieved. The number of the SNR points is fixed to the number of layers.

FGS coding allows the truncating and decoding a bitstream at any point with bit-plane coding. Progressive refinement (PR) slices are used in FGS to achieve fully SNR scalability over a wide range of rate-distortion points. The transform coefficients are encoded successively in PR slices by requantization and a modified entropy coding process.

1.2.3 Multi-View Video Coding

3D video (3DV) is an extension of two-dimensional video to give the viewer the impression of depth. ISO/IEC has specified a language to represent 3D graphic data [74], referred to as virtual reality modeling language (VRML). Later, a language known as BInary Format for Scenes (BIFS) was introduced as an extension of VRML

Fig. 1.17. *Uli* Sequences (Cameras 0, 2, 4)

[74]. Free viewpoint video (FVV) provides the viewers an interactive environment with realistic impressions. The viewers are allowed to choose the view positions and view directions freely. 3DV and FVV have many overlaps in the applications and they can be combined into a single system. The applications span entertainment, education, sightseeing, surveillance, archive and broadcasting [75]. Generally multi-view video sequences are captured simultaneously by multiple cameras. Fig. 1.17 shows an example provided by HHI [76]. The complete test sequences consist of eight video sequences captured by eight cameras with 20cm spacing using 1D/parallel projection. Fig. 1.17 shows the first frames of three sequences taken by Camera 0, 2 and 4.

3DV and FVV representations require the transmission of a huge amount of data. Multi-view video coding (MVC) [77] addresses the problem of jointly compressing multiple video sequences. Besides the spatial and temporal correlations as in a single-view video sequences, multi-view video coding also exploits the inter-view correlations between the adjacent views. As shown in Fig. 1.17, the adjacent sequences recorded by dense camera settings have a high statistical dependency. Even though temporal prediction modes are chosen with a high percentage, inter-view prediction is more suitable for low frame rates and fast motion sequences [76]. After the call for proposals by MPEG, many MVC techniques have been proposed. A multi-view video coding scheme based on H.264 has been presented in [76]. The bitstream is designed in compliance with the H.264 standard and it has shown a significant improvement in

coding efficiency over simulcast anchors. It was chosen as the reference solution in MPEG to build the Joint Multiview Video Model (JMVM) software [78]. An inter-view direct mode is proposed in [79] to save the bits of coding the motion vectors. A view synthesis method is discussed in [80] to produce a virtual synthesis view by the depth map. A novel scalable wavelet based MVC framework is introduced in [81]. Based on the idea of distributed video coding as discussed in Section 1.2.1, a distributed multi-view video coding framework is proposed in [82] to reduce the encoder's computational complexity and the inter-camera communication.

Joint Multiview Video Model (JMVM) [78] is reference software for MVC developed by the Joint Video Team (JVT). JMVM adopted the coding method presented in [76]. The method is based on H.264 with a hierarchical B structure as shown in Fig. 1.18. The horizontal index T_i denotes the temporal index and the vertical index C_i denotes the index of the camera. The N video sequences from the N cameras are rearranged to a single source signal. The spatial, temporal, and inter-view redundancies are removed to generate a standard-compliant compressed bitstream. At the decoder, the single bitstream is decoded and split to N reconstructed sequences. For each separate view, a hierarchical B structure as described in Section 1.2.2 is used. Inter-view prediction is applied to every 2nd view, such as the view taken by the C_1 camera. Each group of picture (GOP) contains N times the length of the GOP for every view. In the example, the length of the GOP for every view is eight and the *Uli* sequences have eight views, which results in a total GOP of $8 \times 8 = 64$ frames. The order of coding is arranged to minimize the memory requirements. However, the decoding of a higher layer frame still needs several references. For example, the decoding of B_3 frame at $(\mathrm{C}_0, \mathrm{T}_1)$ needs four references, namely, the frames at $(\mathrm{C}_0, \mathrm{T}_0)$, $(\mathrm{C}_0, \mathrm{T}_8)$, $(\mathrm{C}_0, \mathrm{T}_4)$, and $(\mathrm{C}_0, \mathrm{T}_2)$. The decoding of B_4 frame at $(\mathrm{C}_1, \mathrm{T}_1)$ needs fourteen references decoded beforehand. From the experimental results, the standard compliant scheme demonstrates a high coding efficiency with a reasonable encoder complexity and memory requirement.

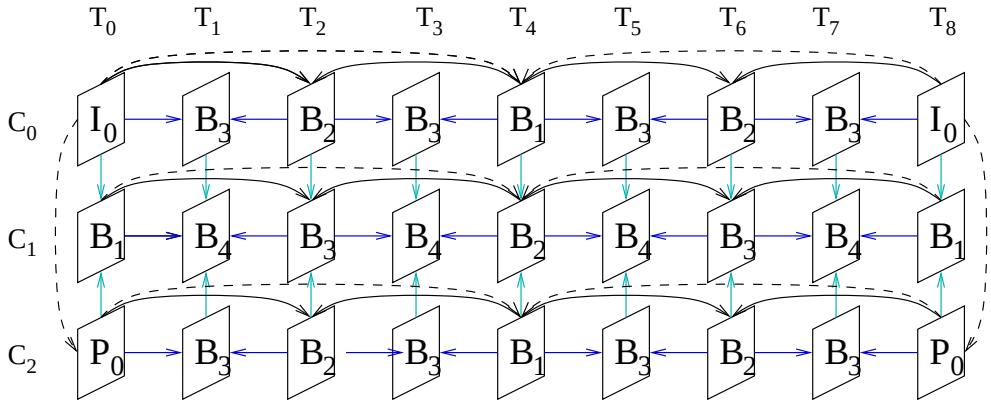


Fig. 1.18. Inter-View/Temporal Prediction Structure

1.3 Overview of The Thesis

1.3.1 Contributions of The Thesis

In this thesis, we study the new approaches for low complexity video coding [37, 49, 83–93]. The main contributions of this thesis are:

- **Wyner-Ziv Video Codec Architecture Testbed**

  We studied the Wyner-Ziv video codec architecture and built a Wyner-Ziv video coding testbed. The video sequences are divided into two different parts using different coding schemes. Part of the sequence is coded by conventional INTRA coding scheme and referred as key frames. The other part of the sequence is coded as Wyner-Ziv frames using channel coding methods. Both turbo codes and low-density-parity-check (LDPC) codes are supported in the system. The sequences can be coded either in pixel domain and transform domain. Only parts of the parity bits from the channel coders are transmitted to the decoder. Hence, the decoding of the Wyner-Ziv frames needs side information at the decoder. The side information can be extracted from the key frames by extrapolation or interpolation. We study various methods for side information generation. In addition, we also analyze the rate-distortion performance of Wyner-Ziv video coding compared with conventional INTRA or INTER coding.

- **Backward Channel Aware Wyner-Ziv Video Coding**

  In Wyner-Ziv video coding, many key frames have to be INTRA coded to keep the complexity at the encoder low and provide side information for the Wyner-Ziv frames. However, the use of INTRA frames limits the coding performance. We propose a new Wyner-Ziv video coding method that uses backward channel aware motion estimation to code the key frames. The main idea is to do motion estimation at the decoder and send the estimated motion vector back to the encoder. In this way, we can keep the complexity low at the encoder with minimum usage of the backward channel. Our experimental results show that the scheme can significantly improve the coding efficiency by 1-2 dB compared with Wyner-Ziv video coding with INTRA-coded key frames. We also propose to use multiple modes of motion decision, which can further improve the coding efficiency by 0.5-2 dB. When the backward channel is subject to erasure errors or delays, the coding efficiency of our method decreases. We provide an error resilience technique to handle the situation. The error resilience technique reduces the quality degradation due to channel errors and only incurs a small coding efficiency penalty when the channel is free of error.

- **Complexity-Rate-Distortion Analysis of Backward Channel Aware Wyner-Ziv Video Coding**

  We further present a model to study the complexity-rate-distortion tradeoff of backward channel aware Wyner-Ziv video coding. We present three motion estimator: minimum motion estimator, median motion estimator and average motion estimator. Suppose we have several motion vectors derived at the decoder. The minimum motion estimator sends all the motion vectors derived at the decoder to the encoder and the encoder makes a decision to choose the best motion vector with smallest distortion. When the backward channel bandwidth cannot meet the requirement or the encoder complexity becomes a concern, the median or average motion estimator chooses one motion vector at the decoder.

The results show the rate-distortion performance of the average estimator is generally higher than that of the median estimator. If the rate-distortion tradeoff is the only concern, the minimum estimator yields better results than the other two estimators. However, for applications with complexity constraints, our analysis shows that the average estimator could be a better choice. Our proposed model quantitatively describes the complexity-rate-distortion tradeoff among these estimators.

1.3.2 Organization of The Thesis

The primary objective of this thesis is to analyze the performance of low complexity video encoding, and to design new techniques to achieve a high video coding efficiency while maintaining low encoding complexity.

In Chapter 2, we first provide an overview of the theoretical background of Wyner-Ziv coding. Then we describe the Wyner-Ziv video coding testbed we developed, followed by a rate-distortion analysis of motion side estimator and coding with universal prediction.

In Chapter 3, we propose a new backward channel aware Wyner-Ziv approach. The basic idea is to use backward channel aware motion estimation to code the key frames, where motion estimation is done at the decoder and motion vectors are sent back to the encoder. We refer to these backward predictive coded frames as BP frames. Error resilience in the backward channel is also addressed by adaptive coding of the key frames.

A model to describe the complexity and rate-distortion tradeoff for BP frames is presented in Chapter 4. Three estimators, the minimum estimator, the median estimator and the average estimator, are proposed and complexity-rate-distortion analysis is presented.

Chapter 5 concludes the thesis.

2. WYNER-ZIV VIDEO CODING

Wyner-Ziv coding is a new coding scheme based on two theorems presented in 1970s. In this coding scenario, source statistics are exploited at the decoder so that it is feasible to design a simplified encoder. Several practical designs of Wyner-Ziv video codec are based on channel coding methods. In these systems, the complexity of the encoder due to motion estimation in hybrid video coding is shifted to the decoder. Specially, the derivation of side information at the decoder involves high complexity motion estimation to ensure high quality side information.

In this chapter, we describe the general Wyner-Ziv video coding (WZVC) architecture. Section 2.1 gives an overview of the theoretical background behind Wyner-Ziv coding. Section 2.2 describes the overall structure of Wyner-Ziv video coding and the processing units in the system. Section 2.3 presents the rate-distortion analysis of motion side estimator. Section 2.4 presents Wyner-Ziv video coding with universal prediction which achieves low complexity at both the encoder and the decoder.

2.1 Theoretical Background

Two theorems presented in the 1970s [26,27] play key roles in the theoretical foundation of distributed source coding. The Slepian and Wolf theorem [26] proved that the lossless encoding scheme without side information at the encoder may perform as well as the encoding scheme with side information at the encoder. Wyner and Ziv [27] extended the result to establish rate-distortion bounds for lossy compression. There was not much progress on constructive schemes due to the inability of finding a practical channel coding method [94] that achieves the bound.

Information-theoretic duality between source coding with side information (SCSI) at the decoder and channel coding with side information (CCSI) at the encoder is

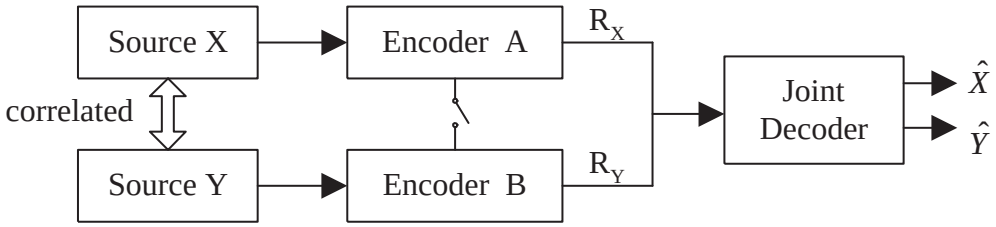


Fig. 2.1. Correlation Source Coding Diagram

discussed in [95]. The second scenario was first studied by Costa in the "dirty paper" problem [96] and exploited again recently because of its expansive applications in data hiding, watermarking and multi-antenna communications. In this section we describe the Slepian-Wolf theorem and Wyner-Ziv theorem in detail.

2.1.1 Slepian-Wolf Coding Theorem for Lossless Compression

Suppose two sources X and Y are encoded as shown in Fig. 2.1. When they are jointly encoded, i.e., the switch between the encoders is on, the admissible rate bound for the error-free case is:

$$R_{X,Y} \geq H(X,Y) \tag{2.1}$$

where $R_{X,Y}$ denotes the total data rate to jointly encode X and Y, H(X,Y) denotes the joint entropy of X and Y [97].

Slepian and Wolf discussed the case when the switch is off. Surprisingly, even though the two sources are encoded separately, the sum of rates R_X and R_Y can still achieve the joint entropy $H(X,Y)$ as long as they are jointly decoded, where R_X represents the data rate used to encode the source X and R_Y represents the data rate

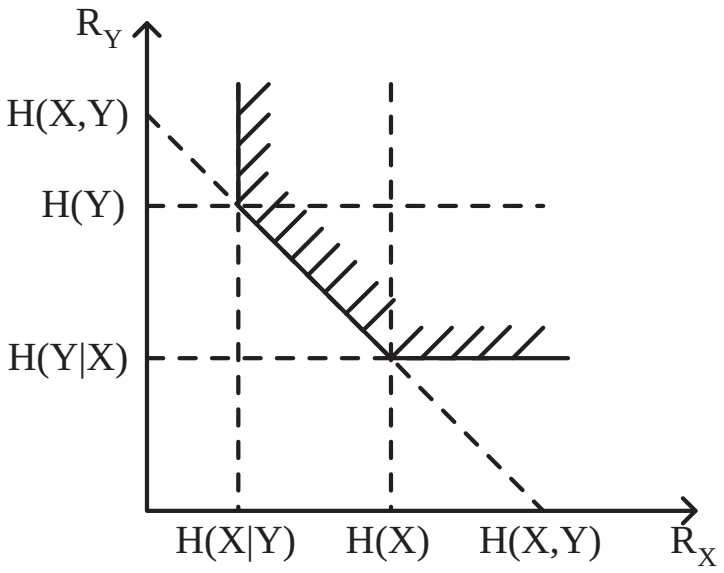


Fig. 2.2. Admissible Rate Region for the Slepian-Wolf Theorem

used to encode the source Y. The Slepian-Wolf Theorem proved the admissible rate bounds for distributed coding of two sources X and Y [26]:

$$R_X \geq H(X|Y) \tag{2.2}$$

$$R_Y \geq H(Y|X) \tag{2.3}$$

$$R_X + R_Y \geq H(X,Y) \tag{2.4}$$

where $H(X|Y)$ denotes the conditional entropy of X given Y, $H(Y|X)$ denotes the conditional entropy of Y given X. Fig. 2.2 shows the admissible rate region for the Slepian-Wolf theorem.

2.1.2 Wyner-Ziv Coding Theorem for Lossy Compression

Wyner and Ziv proved the rate-distortion bounds for lossy compression [27]. Although the rate of separate coding may be greater than the rate of joint coding, the equality is achievable, for example, for a Gaussian source and a mean square error

metric when joint decoding is allowed. Hence, the side information at the encoder is not always necessary to achieve the rate distortion bound. As shown in Fig. 2.3, the source data X and side information Y are both random variables. The decoder has access to side information, while the switch determines whether the encoder has the access to the side information. Wyner and Ziv's theorem proved that

$$R^*(d) \geq R_{X|Y}(d) \tag{2.5}$$

where d is the measure of the distortion between the source X and the reconstruction $\hat{X}$ at the decoder, $0 \leq d < \infty$. $R^*(d)$ denotes the rate distortion function when side information Y is only available at the decoder. $R_{X|Y}(d)$ denotes the rate distortion function when side information Y is available at both the encoder and the decoder. Wyner and Ziv presented a specific case where equality is achieved. They showed that with Gaussian memoryless sources and mean-squared error distortion, no rate loss incurs when the encoder has no access of side information, i.e., X is Gaussian and $Y = X + U$, where U is also Gaussian and independent of X. The distortion is

$$d(X, \hat{X}) = E[(X - \hat{X})^2] \tag{2.6}$$

Under these conditions, the rates required to code the source in both cases are:

$$R^*(d) = R_{X|Y}(d) = \begin{cases} \frac{1}{2} \log \frac{\sigma_U^2 \sigma_X^2}{(\sigma_X^2 + \sigma_U^2)d}, & 0 < d \leq \frac{\sigma_U^2 \sigma_X^2}{\sigma_X^2 + \sigma_U^2} \\ 0, & d \geq \frac{\sigma_X^2 \sigma_U^2}{\sigma_X^2 + \sigma_U^2} \end{cases} \tag{2.7}$$

2.2 Wyner-Ziv Video Coding Testbed

A Wyner-Ziv video codec generally formulates the video coding problem as an error correction or noise reduction problem. Hence existing state-of-the-art channel coding methods are used in the development of Wyner-Ziv codecs. Fig. 2.4 shows an example of side information in video coding. We use the previously reconstructed frame as the initial estimate to decode the current frame. The reference frame, as

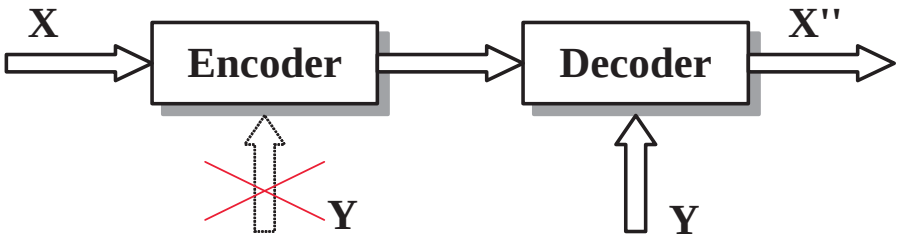


Fig. 2.3. Wyner-Ziv Coding with Side Information at the Decoder

(a) Reference Frame (b) Current Frame

Fig. 2.4. Example of Side Information

shown in Fig. 2.4-(a), can be considered as the side information of the current frame in Fig. 2.4-(b).

Wyner-Ziv video coding using turbo codes and Low Density Parity Check (LDPC) codes are two popular systems in the literature. Both systems show better rate distortion performance than the conventional INTRA frame coding. Various methods have been proposed to improve coding performance. Several papers exploit the relationship between the side information and the original source [98–100]. Analytical result of the performance of uniform scalar quantizer is presented in [101]. A thorough study of the statistics of the feedback channel used to request parity bits is presented in [102]. A Flexible Macroblock Order (FMO)-like algorithm [103] is used to partition the frame such that spatial or temporal side information is generated adaptively to outperform the case with motion interpolated side information only. Hash codewords are sent over to the decoder to aid the decoding of the Wyner-Ziv frame and help to build a low-delay system in [104]. An encoder or decoder based mode decision is made to embed a block-based INTRA mode [105, 106]. A simple frame subtraction process is introduced to code the residual instead of the original frame [107]. In the following we describe our Wyner-Ziv video coding testbed.

2.2.1 Overall Structure of Wyner-Ziv Video Coding

Our Wyner-Ziv video coding (WZVC) testbed is shown in Fig. 2.5. The input video sequence is divided into two groups which are coded by two different methods. Part of the frames are coded using a H.264 INTRA frame coder, which are denoted as key frames. The remaining frames are independently coded using channel coding methods, referred to as Wyner-Ziv frames. The INTRA key frames are used to keep the complexity at the encoder low as well as to generate side information for the Wyner-Ziv frames at the decoder. Only parts of the parity or syndrome bits from the channel encoder are transmitted to the decoder. Hence, the decoding of these frames need side information at the decoder. The side information can be extracted from the neighboring key frames by extrapolation or interpolation. The increase of the distance between two neighboring key frames will degrade the quality of the side information of Wyner-Ziv frame. It is essential to find a good tradeoff between the number of the key frames and the degradation of the side information. Fig. 2.6 shows an example of group of pictures (GOP) of Wyner-Ziv video coding. Every other frame is coded as intra frame and the other frames are coded as Wyner-Ziv frames. The side information of the Wyner-Ziv frame can be derived from the two neighboring intra frames.

Two channel coding methods are supported in the system, turbo codes and low-density-parity-check (LDPC) codes. The Wyner-Ziv frames can be coded either in the pixel domain or the transform domain with the integer transform used in H.264. The coefficients are coded bitplane by bitplane. The most significant bitplane is first coded by the channel encoder, followed by the other bitplanes with less significance. The entire bitplane is coded as a block with the channel coder. Output of the channel coder is sent to the decoder. Only part of the parity bits from the encoder are transmitted through the channel. We assume that the decoder can request more parity bits until the bitplane is correctly decoded.

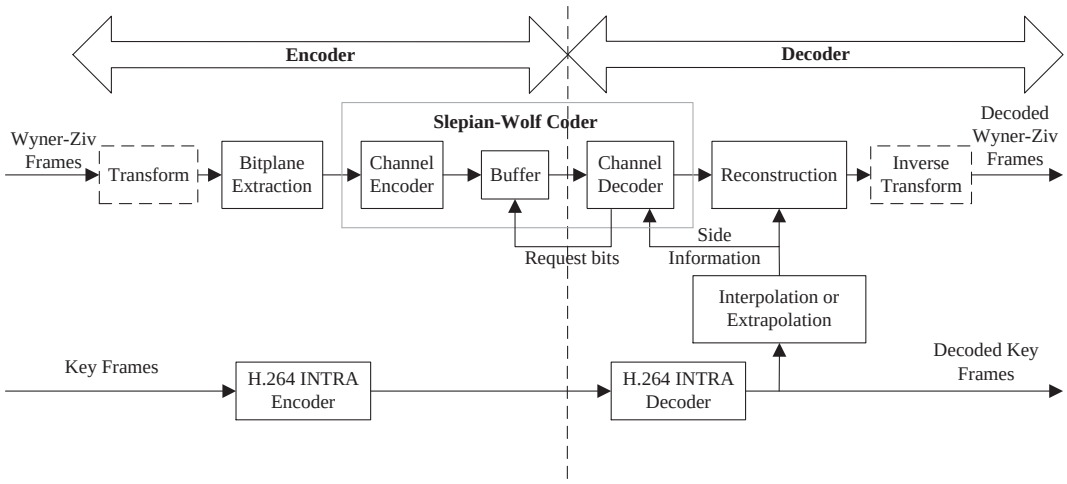


Fig. 2.5. Wyner-Ziv Video Coding Structure

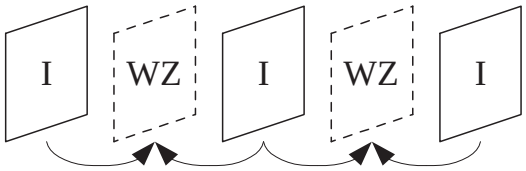


Fig. 2.6. An Example of GOP in Wyner-Ziv Video Coding

At the decoder, the key frames can be independently decoded by H.264 INTRA decoder. To reconstruct the Wyner-Ziv frames, the decoder first derives the side information from the previously decoded key frames. Side information is an initial estimate or noisy version of the current frame. The incoming channel coder bits help to reduce the noise and reconstruct the current Wyner-Ziv frame based on this initial estimate. The decoder assumes a statistical model of the "correlation channel" to exploit the side information. The difference between the original frame and the estimate is modeled as a Gaussian or Laplacian distribution. If the system is coded in transform domain, the side information is also integer transformed. The coefficients either in the pixel domain or in the transform domain are represented by bitplanes. The channel decoder uses the side information in the bitplane representation and the channel coder bits to decode the symbol. If the decoded symbol is consistent with the side information, the decoded symbol is used for the reconstruction with the other decoded symbols from different bitplanes. Otherwise, the reconstruction process uses the side information in bitplane representation as the reconstruction to prevent errors. If the video sequence is coded in the transform domain, inverse integer transform is used to recover the sequence after reconstruction.

2.2.2 Channel Codes (Turbo Codes and LDPC Codes)

The testbed supports two channel coding methods, turbo codes and low-density-parity-check (LDPC) codes. The turbo code is built upon the codec from [108] and the LDPC code is built upon the codec from [109]. The basic structure of two channel coders are described in the following.

Turbo Codes

The structure of turbo encoder is shown in Fig. 2.7 [38–41]. The input X is sent to two identical recursive systematic convolutional (RSC) encoders. Before being transmitted to one of the RSC encoders, the symbols are randomly interleaved. The two

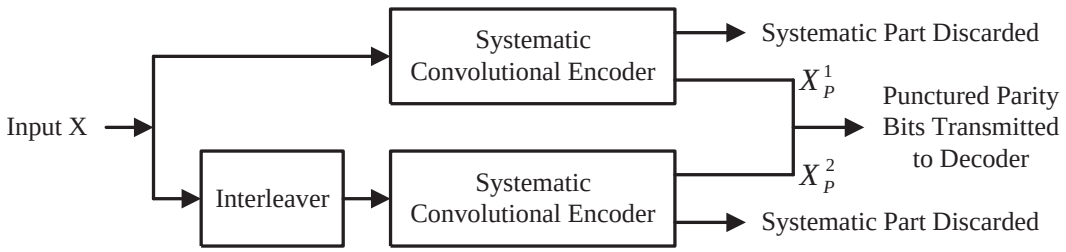


Fig. 2.7. Structure of Turbo Encoder Used in Wyner-Ziv Video Coding

Table 2.1
Generator matrix of the RSC encoders

state	$g_1(D)$	$g_2(D)$	octal form
4	$1+D$	$1+D^2$	(7,5)
8	$1+D^2+D^3$	$1+D^3$	(17,15)
16	$1+D+D^4$	$1+D^2+D^3+D^4$	(31,27)

RSC encoders are parallel-concatenated and hence termed as Parallel Concatenated Convolutional Codes (PCCC). In this application, the systematic parts of the output are discarded and part of the parity bits are sent to the decoder. The puncture deletes selected parity bits to reduce coding overhead.

The structure of the RSC encoders is simple which guarantees low complexity encoding. Generally the RSC codes used in the turbo encoder have the generator matrix

$$G_R(D) = [1 \qquad \frac{g_2(D)}{g_1(D)}] \tag{2.8}$$

Table 2.1 summaries several generator matrices that are frequently used. An example of the RSC code with 16 states is given in Fig. 2.8.

The structure of the turbo decoder as shown in Fig. 2.9 is computationally complex compared with the encoder. X_P^1 and X_P^2 denote the parity bits generated by the two RSC encoders. The input Y denotes the dependency channel output, which is

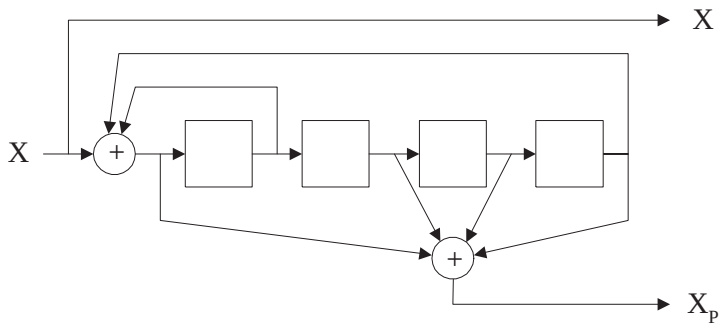


Fig. 2.8. Example of a Recursive Systematic Convolutional (RSC) Code

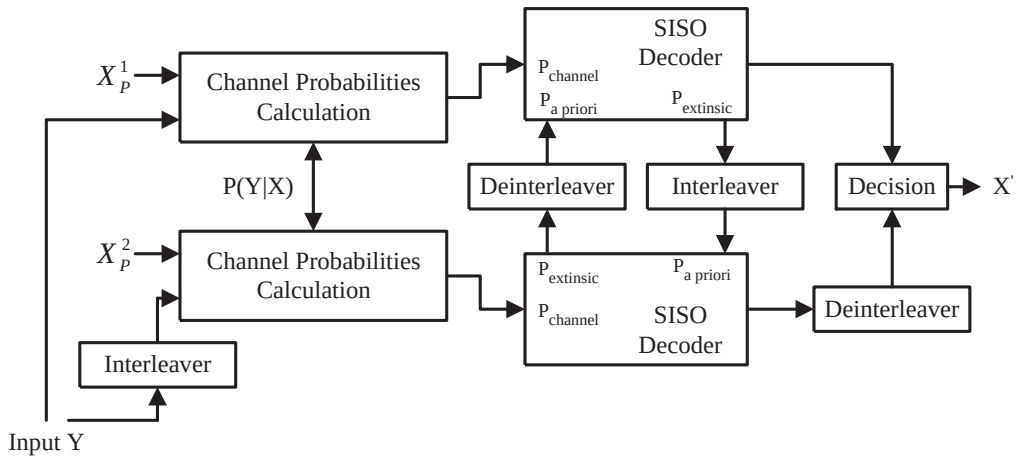


Fig. 2.9. Structure of Turbo Decoder Used in Wyner-Ziv Video Coding

side information available at the decoder in Wyner-Ziv coding. The decoder includes two soft-input soft-output (SISO) constituent decoders [38–41].

Low Density Parity Check (LDPC) Codes

Low Density Parity Check code is an error correcting code that operates close to the Shannon limit [42–45]. In the following, we consider only binary LDPC codes. A LDPC code is a linear block code determined by a generator matrix G or a parity check matrix H. Suppose the linear block code is a (N, K) code. G is a $K \times N$ matrix represented as $G = [I_K : P]$ and H is a $(N - K) \times N$ matrix represented as $H = [P^T : I_{N_K}$. The encoding of the LDPC code is determined by $c = G^T X$ where X is the input vector. All the codewords generated by G satisfy $Hc = 0$. The relationship can be represented by the Tanner graph as shown in Fig. 2.10 where a (7,4) linear code is used. H is a 3×7 matrix with entries

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \tag{2.9}$$

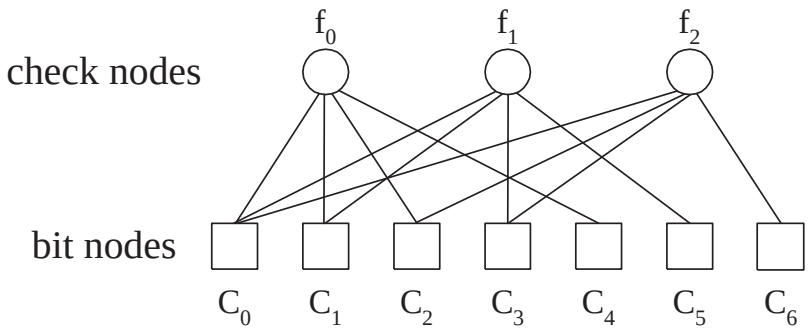


Fig. 2.10. Tanner Graph of a (7,4) LDPC Code

LDPC code has a sparse H, where there are few 1's in the rows and columns. Regular LDPC codes contain exactly fixed number of 1's every column and every row. Irregular LDPC codes has different numbers of 1's every row or column.

The LDPC decoder iteratively estimates the distributions in graph-based model, where the brief propagation algorithm is used. Given the side information Y, the log-likelihood ratio is

$$L(x) = \log \frac{P(x=0|y)}{P(x=1|y)} \tag{2.10}$$

and the estimate is

$$\hat{x} = \begin{cases} 0 & L(x) \geq 0 \\ 1 & L(x) < 0 \end{cases} \tag{2.11}$$

2.2.3 Derivation of Side Information

The key frames can be decoded independently by the H.264 INTRA frame decoder. The previously decoded key frames are used to derive the side information of the Wyner-Ziv frames by extrapolation or interpolation in pixel domain. There are some simple ways to obtain the side information. Suppose frame n is the current frame and frame $(n-1)$ and $(n+1)$ are the neighboring frames. For example, we can use the previous reconstructed frame $(n-1)$ as the side information of the current frame n. Another approach is to take the average of the pixel values from two neigh-

boring frames $(n-1)$ and $(n+1)$. In these cases, the quality of the side information is low and there is no motion estimation at the decoder. To obtain higher quality side information, motion estimation can be done at the decoder which involves high complexity processing. Side information can be obtained by extrapolating the previous reconstructed frame as shown in Fig. 2.11. For every block in current frame n, we search for the motion vector MV_{n-1} of the co-located macroblock in previous frame $(n-1)$. For natural scenes, the motion vectors of neighboring frames are closely related and we can predict the motion vectors of the current frame from the adjacent previously decoded frames. We use MV_{n-1} as an estimate of the motion vector of the current frame MV_n. The patterned reference block in frame $(n-1)$ is derived using MV_n and used as the side information of the current macroblock in frame n.

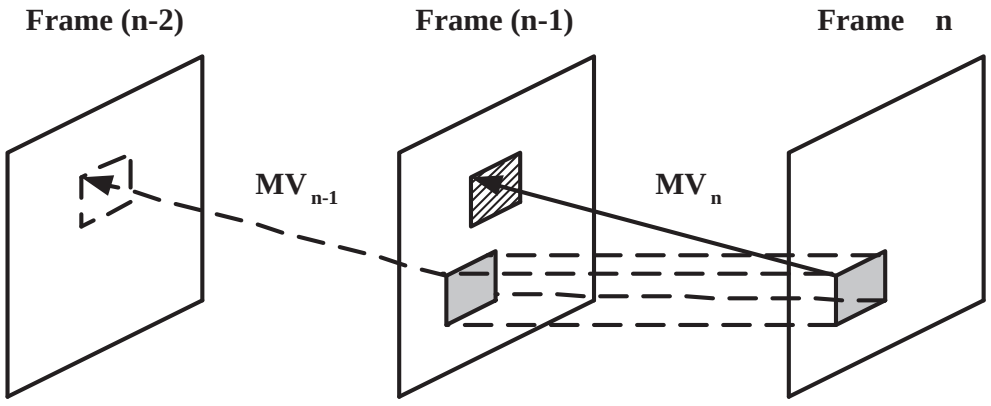


Fig. 2.11. Derivation of Side Information by Extrapolation

We can also use interpolation to obtain the side information. As shown in Fig. 2.12, motion search is done between the $(n-1)$-th key frame $\hat{s}(n-1)$ and $(n+1)$-th key frame $\hat{s}(n+1)$. For each block in the current frame, as shown in Fig. 2.12, the side estimator first uses the co-located block in the next reconstructed frame $\hat{s}(n+1)$ as the source and the previous reconstructed frame $\hat{s}(n-1)$ as the reference to perform forward motion estimation. We denote the obtained motion vector as MV_F. We then use the co-located block in the previous frame as the source and

next reconstructed frame as the reference to perform backward motion estimation. Denote the obtained motion vector as MV_B. The side estimator uses $\frac{MV_F}{2}$ from $\hat{s}(n-1)$ to find the corresponding reference block P_{F1}, and $-\frac{MV_F}{2}$ from $\hat{s}(n+1)$ to find the corresponding reference block P_{F2}. We also use $\frac{MV_B}{2}$ from $\hat{s}(n+1)$ to find the corresponding reference block P_{B1}, and $-\frac{MV_B}{2}$ from $\hat{s}(n-1)$ to find the corresponding reference block P_{B2}. The reference block is

$$P = \frac{P_{F1} + P_{F2} + P_{B1} + P_{B2}}{4} \tag{2.12}$$

This average of the four references is the initial estimate of the side information.

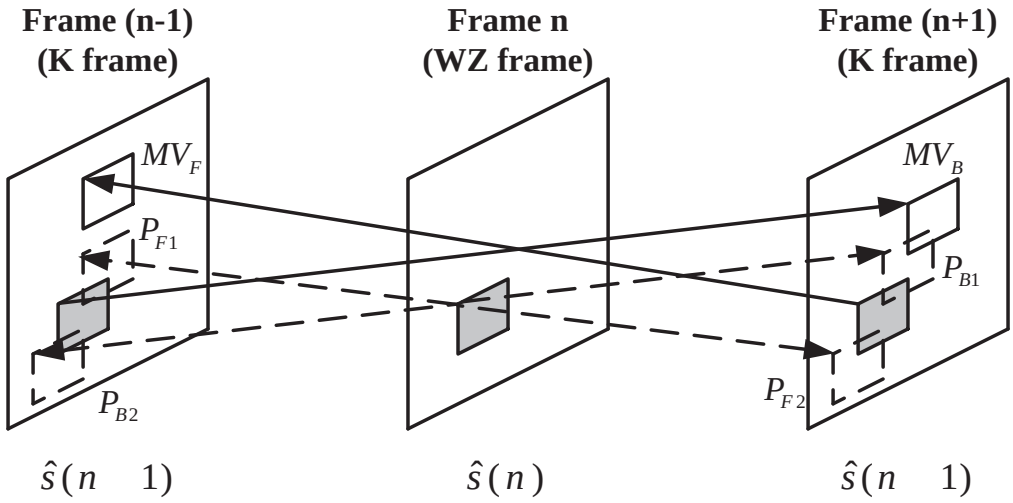


Fig. 2.12. Derivation of Side Information by Interpolation

A refined side estimator can be used to more effectively extract reference information from the decoder. Many current side estimators use the information extracted from the previous reconstructed frames. With input of a Wyner-Ziv frame, the decoder gradually improves the reconstruction of the current frame. It is possible to use the information from the current frame's lower quality reconstruction as well. This is analogous to SNR scalability used in conventional video coding. In that case, a previous frame's reconstruction is first used as a reference, while a lower quality reconstructions of current frame can later be used as references for the enhancement

layers. The detailed implementation of refined side estimator is shown in Fig. 2.13. After the incoming parity bits are used along with the reference to reconstruct a lower quality reconstruction of current frame $\hat{s}_b(n)$, the refined side estimator performs a second motion search. In refined motion search, for every block in $\hat{s}_b(n)$, the best match in the previous and following key frames respectively is obtained and results in two new motion vectors $MV_{F,RSE}$ and $MV_{B,RSE}$. The two best matched blocks in the adjacent key frames are then averaged to construct a new side information. The parity bits received are now used for second-round decoding with this new side information. The new reference uses the information of previous side information and can further improve the quality of the side information.

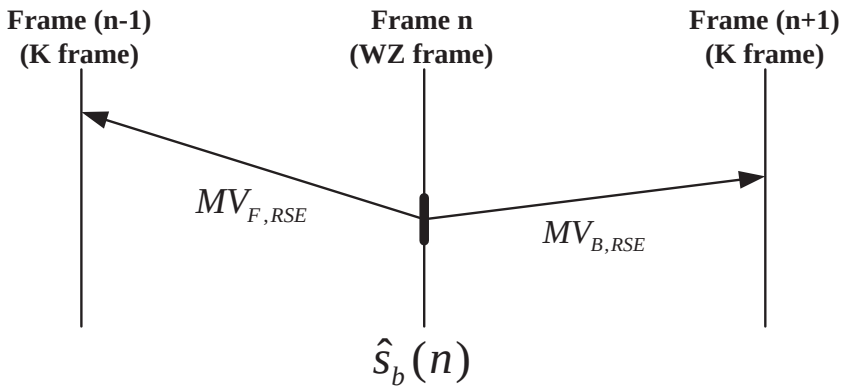


Fig. 2.13. Refined Side Estimator

2.2.4 Experimental Results

We compare our implementation with conventional video coding methods. We test the following coding methods.

- H.263 INTRA: Every frame is coded by H.263+ reference software TMN3.1.1 INTRA mode;

- H.264 INTRA: Every frame is coded by H.264 reference software JM8.0 INTRA mode;

- H.264 IBIB: Every even frame is coded by JM8.0 INTRA mode, while the odd frames are coded by JM8.0 bi-directional mode with quarter-pixel motion search accuracy;

- I-WZ: Every even frame is coded by JM8.0 INTRA mode, while the odd frames are coded as a Wyner-Ziv frame. At the decoder, the side information is derived by interpolation as shown in Fig. 2.12. Motion search is performed with quarter-pixel accuracy. Then refine motion search as shown in Fig. 2.13 is performed to further improve the coding efficiency.

We tested six standard QCIF sequences, *Foreman*, *Coastguard*, *Carphone*, *Silent*, *Stefan* and *Table Tennis* each of which consists of 300 frames. The frame rate is 30 frames per second. The data rate of H.263 INTRA, H.264 INTRA and H.264 IBIB is adjusted by the quantization parameter (QP). For I-WZ, we adjust the Wyner-Ziv frame's data rate by setting the number of bitplanes used for decoding, while the data rate of the key frame is controlled by QP. The rate distortion performance is averaged over 300 frames.

In Fig. 2.14 - 2.19, we show the video coding results. Compared with conventional INTRA coding results, the Wyner-Ziv video coding generally outperforms H.264 INTRA coding by 2-3 dB and H.263+ INTRA coding by 3-4 dB. This shows that by exploiting source statistics in the decoder, a simple encoder can achieve better coding results than independent encoding and decoding methods such as INTRA coding. Compared with H.264 IBIB, the Wyner-Ziv video coding still trails by 2-4 dB.

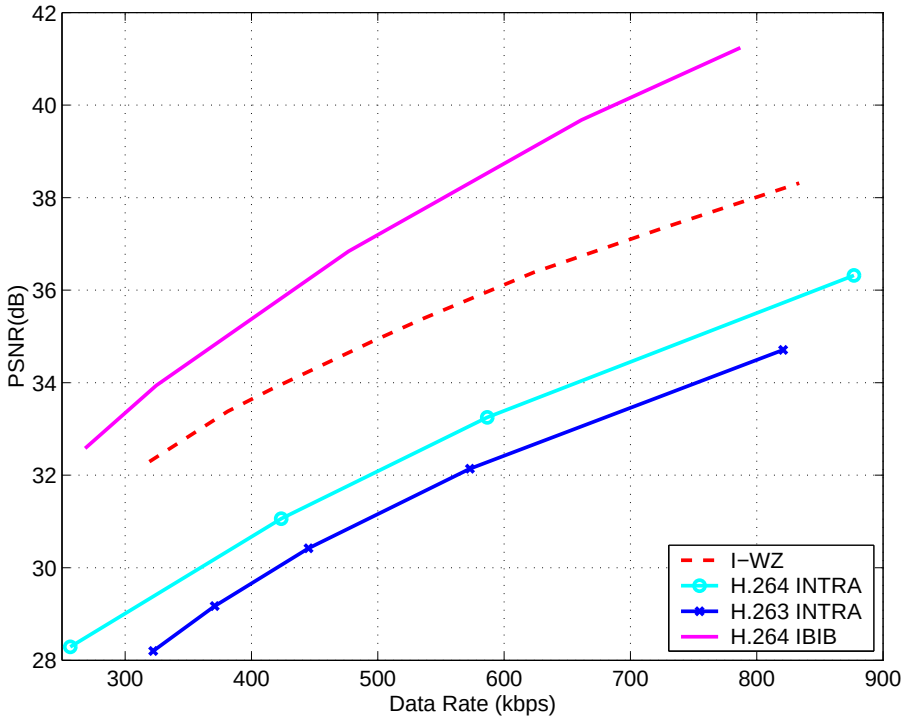


Fig. 2.14. WZVC Testbed: R-D Performance Comparison (*Foreman* QCIF)

2.3 Rate Distortion Analysis of Motion Side Estimation in Wyner-Ziv Video Coding

In this section we study the rate-distortion performance of motion side estimation in Wyner-Ziv video coding (WZVC) [90, 110]. There are three terms leading to the performance loss of Wyner-Ziv coding compared to conventional MCP-based coding. System loss is due to the fact that side information is unavailable at the encoder for WZVC. Source coding loss is caused by the inefficiency of channel coding methods and quantization schemes that cannot achieve Shannon limit. We will focus on the third item, video coding loss, in the following analysis. Video coding loss is due to the fact that the side information is not perfectly generated at the decoder. For MCP-based video coding, the reference of the current frame is generated from the previous reconstructed neighboring frames and the current frame. However, WZVC generates

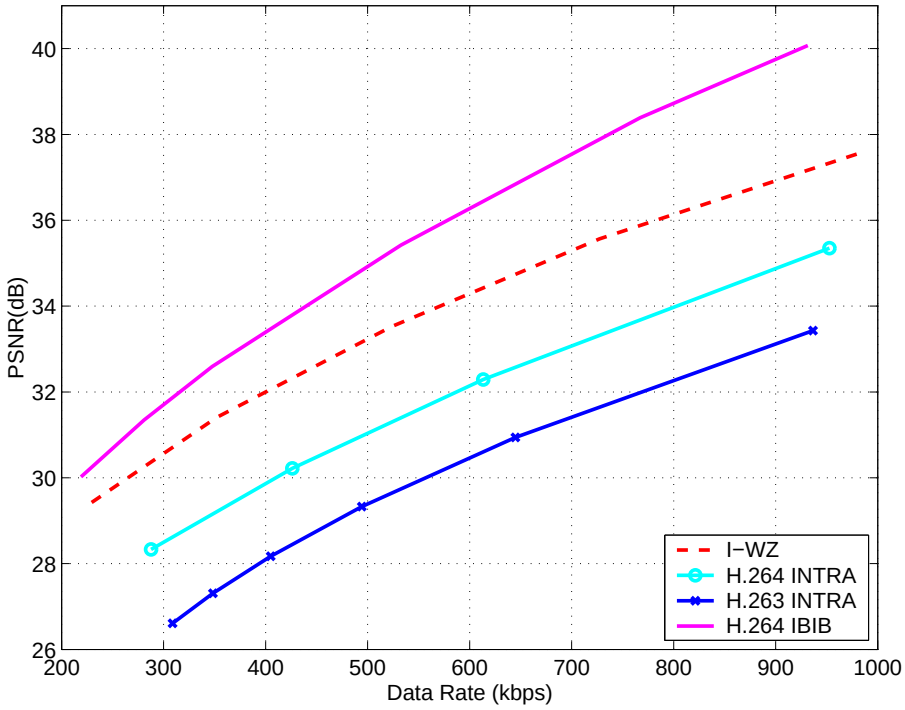


Fig. 2.15. WZVC Testbed: R-D Performance Comparison (*Coastguard* QCIF)

the reference only from the previous reconstructed neighboring frames without access to the current frame.

The rate analysis of residual frame is formulated based on a general power spectrum model [65–67, 90, 110] and it would be applied to WZVC later. Assume $e(n) = s(n) - c(n)$, where $e(n)$ denotes the residual frame, $s(n)$ denotes the original source frame, and $c(n)$ denotes the reference frame. The power spectrum of the residual frame is:

$$\begin{aligned}
\Phi_{ee}(\omega) &= \Phi_{ss}(\omega) - 2Re\{\Phi_{cs}(\omega)\} + \Phi_{cc}(\omega) \\
\Phi_{cs}(\omega) &= \Phi_{ss}(\omega)E\{e^{-j\omega^T\Delta}\} = \Phi_{ss}(\omega)e^{-\frac{1}{2}\omega^T\omega\sigma_\Delta^2} \\
\Phi_{cc}(\omega) &= \Phi_{ss}(\omega) \\
\Phi_{ee}(\omega) &= 2\Phi_{ss}(\omega) - 2\Phi_{ss}(\omega)e^{-\frac{1}{2}\omega^T\omega\sigma_\Delta^2} \\
\frac{\Phi_{ee}(\omega)}{\Phi_{ss}(\omega)} &= 2 - 2e^{-\frac{1}{2}\omega^T\omega\sigma_\Delta^2}
\end{aligned} \tag{2.13}$$

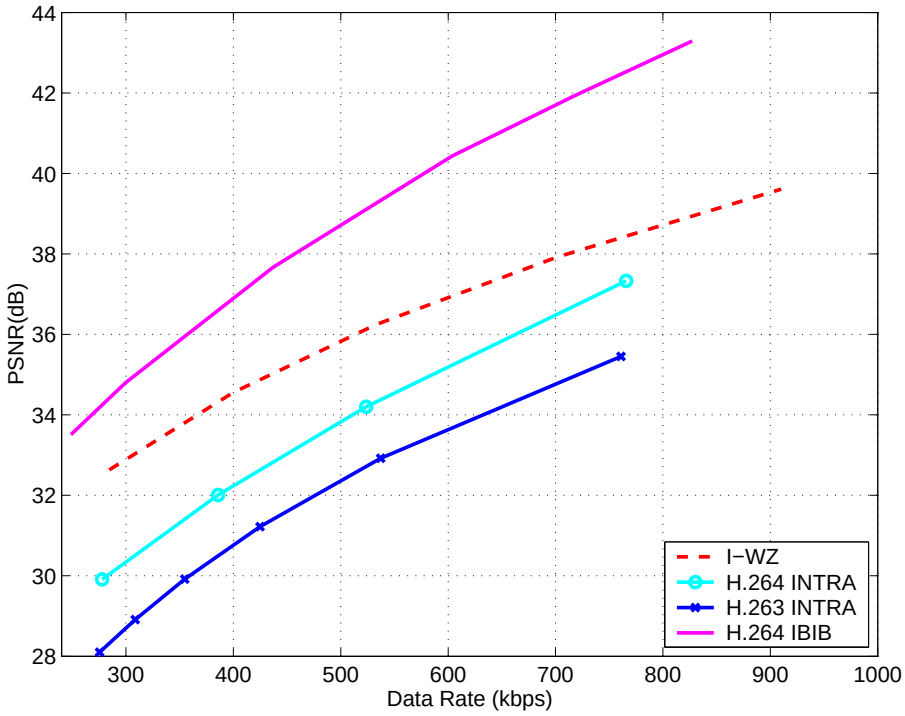


Fig. 2.16. WZVC Testbed: R-D Performance Comparison (*Carphone* QCIF)

where Δ is the error motion vector and σ_Δ^2 is the variance of the error motion vector. The error motion vector is the difference between the derived motion vector and the true motion vector, where the true motion vector is an ideal motion vector with minimum distortion. The rate saving over INTRA-frame coding by MCP or other motion search methods is [111]

$$\Delta R = \frac{1}{8\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \log_2 \frac{\Phi_{ee}(\omega)}{\Phi_{ss}(\omega)} d\omega \tag{2.14}$$

Hence the rate difference between two systems using two motion vectors is

$$\Delta R_{1,2} = \frac{1}{8\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \log_2 \frac{1 - e^{-\frac{1}{2}\omega^T \omega \sigma_{\Delta_1}^2}}{1 - e^{-\frac{1}{2}\omega^T \omega \sigma_{\Delta_2}^2}} d\omega \tag{2.15}$$

Wyner-Ziv video coding is compared with two conventional MCP-based video coding methods, i.e., DPCM-frame video coding and INTER-frame video coding. DPCM-frame coding subtracts the previous reconstructed frame from the current frame and codes the difference. INTER-frame coding performs motion search at the

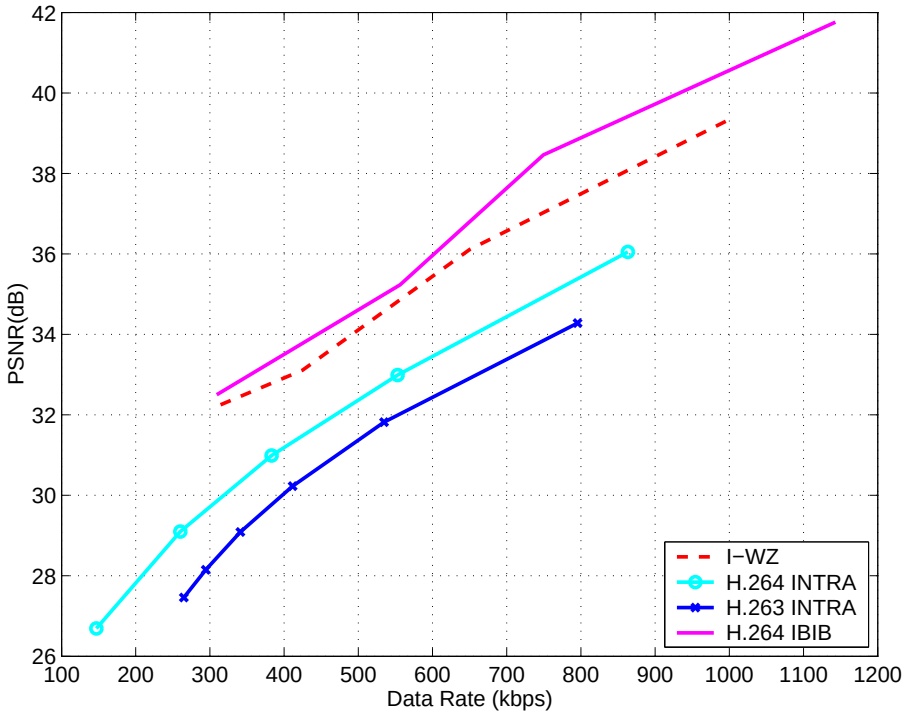


Fig. 2.17. WZVC Testbed: R-D Performance Comparison (*Silent* QCIF)

encoder and codes the residual frame. The rate difference between the three coding methods using (2.15) are

$$\Delta R_{DPCM,WZ} = \frac{1}{8\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}\log_2\frac{1-e^{-\frac{1}{2}\omega^T\omega\sigma_{MV}^2}}{1-e^{-\frac{1}{2}\omega^T\omega(1-\rho^2)\sigma_{MV}^2}}d\omega \quad (2.16)$$

$$\Delta R_{DPCM,INTER} = \frac{1}{8\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}\log_2\frac{1-e^{-\frac{1}{2}\omega^T\omega\sigma_{MV}^2}}{1-e^{-\frac{1}{2}\omega^T\omega(1-\rho^2)\sigma_{\Delta_\beta}^2}}d\omega \quad (2.17)$$

where σ_{MV}^2 denotes the variance of the motion vector, ρ denotes the correlation between the true motion vector and the motion vector obtained by the side estimator, $\sigma_{\Delta_\beta}^2$ denotes the variance of the motion vector error. The rate savings over DPCM-frame video coding by using Wyner-Ziv video coding is more significant when the motion vector variance σ_{MV}^2 is small. This makes sense since for lower motion vector variance, the side estimator has a better chance to estimate a motion vector close to the true motion vector. Wyner-Ziv coding can achieve a gain up to 6 dB (for small motion vector variance) or 1-2 dB (for normal to large motion vector variance)

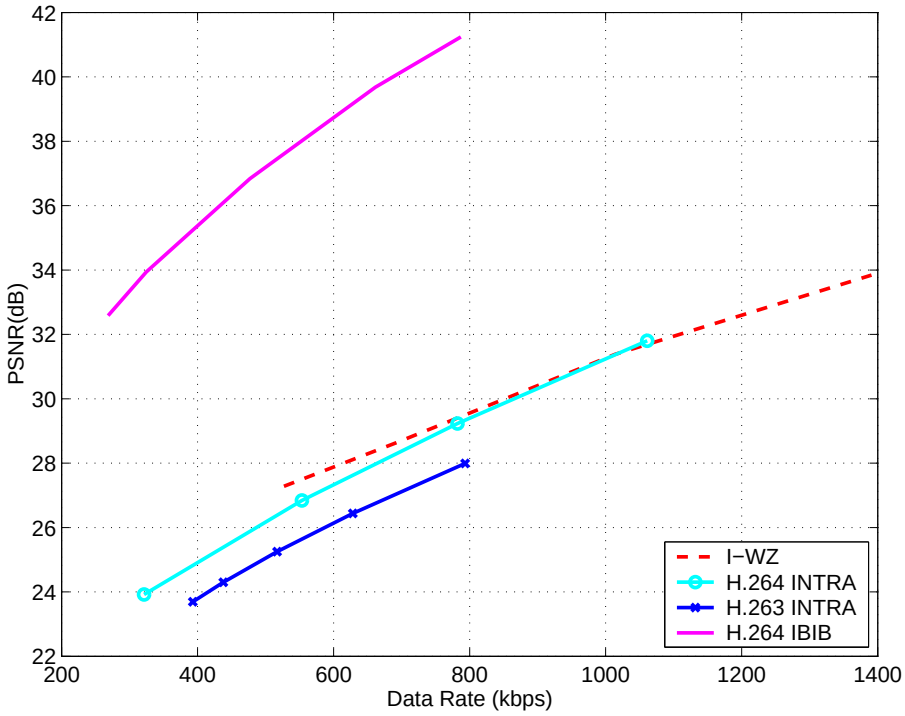


Fig. 2.18. WZVC Testbed: R-D Performance Comparison (*Stefan* QCIF)

over DPCM-frame video coding. INTER-frame coding outperforms Wyner-Ziv video coding by around 6 dB generally. For sequences with small σ_{MV}^2, the improvement is smaller, ranging from 1 - 4 dB depending on the specific side estimator used.

We further study the side estimators using two motion search methods, sub-pixel motion search and multi-reference motion search. In conventional MCP-based video coding, the accuracy of the motion search has a great influence on the coding efficiency. However, Wyner-Ziv video coding is not as sensitive to the accuracy of the motion search. For small σ_{MV}^2, motion search using integer pixel falls behind the method using quarter pixel by less than 0.4 dB. The coding difference with larger σ_{MV}^2 is even smaller. In this case, using $2:1$ subsampling does not affect the coding efficiency significantly. Fig. 2.20 shows an example of *Foreman* QCIF sequence. The half pixel search accuracy can improve around 0.2-0.3 dB over integer motion search accuracy. Quarter pixel search accuracy fails to have noticeable improvement over half pixel

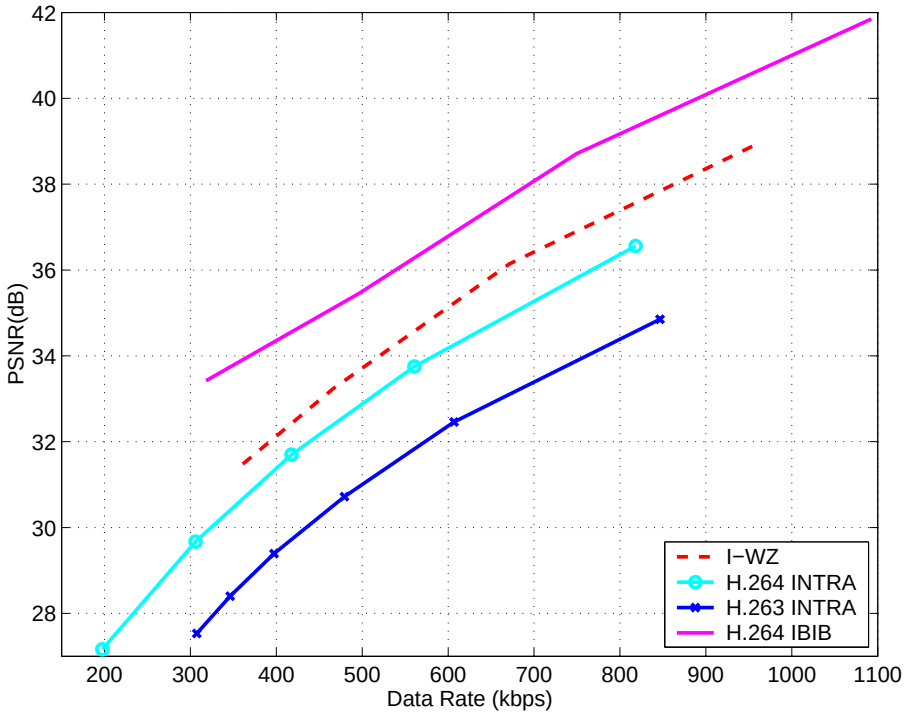


Fig. 2.19. WZVC Testbed: R-D Performance Comparison (*Table Tennis* QCIF)

search accuracy. Considering 2:1 subsampled motion search, it only incurs 0.1 dB coding loss compared to integer motion search accuracy. The experimental results are consistent with our analytical result. When the decoder complexity becomes an issue, the 2:1 subsampled side estimator can be an acceptable alternative. The results for other sequences show similar patterns.

The rate difference between N references and one reference is

$$\Delta R_{N,1} = \frac{1}{8\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} log_2 I_{MR}(\omega, N) d\omega \tag{2.18}$$

and

$$I_{MR}(\omega, N) = \frac{\frac{N+1}{N} - 2e^{-\frac{1}{2}\omega^T \omega \sigma^2_{\Delta a}} + \frac{N-1}{N} e^{-(1-\rho_\Delta \omega^T \omega \sigma^2_{\Delta a})}}{2 - 2e^{-\frac{1}{2}\omega^T \omega \sigma^2_{\Delta a}}} \tag{2.19}$$

where ρ_Δ is the correlation between two motion vector errors and we consider the case $\rho_\Delta = 0$. $\sigma^2_{\Delta a}$ denotes the actual variance of the motion vector error, which is due to the motion search pixel inaccuracy and the imperfect correlation between current motion vectors and previous motion vectors. The analysis of the rate difference using N

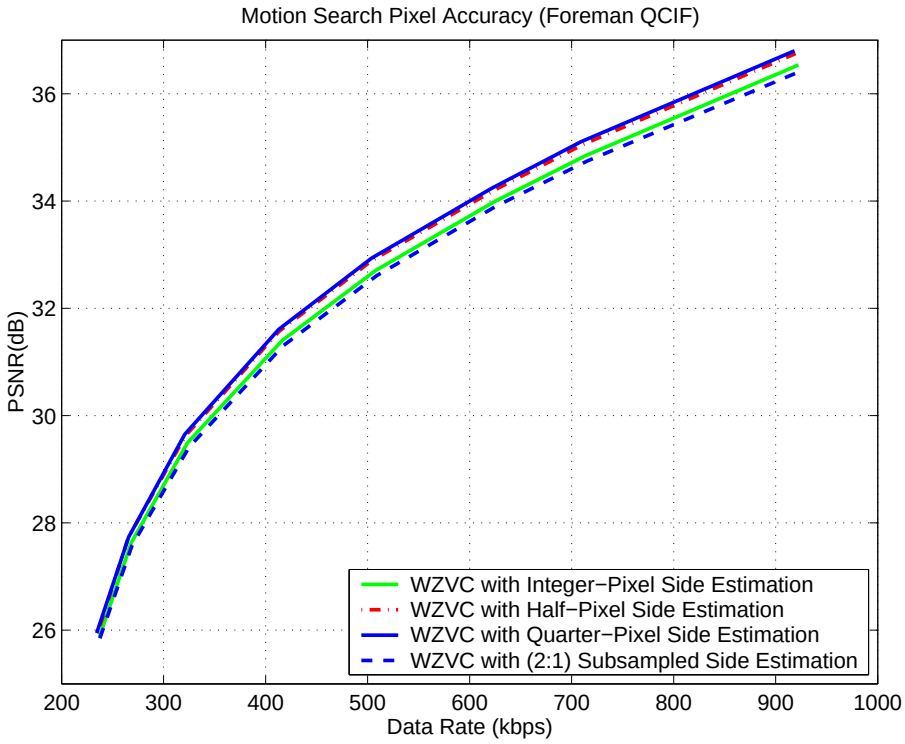


Fig. 2.20. Wyner-Ziv Video Coding with Different Motion Search Accuracies (*Foreman* QCIF)

references over one reference shows that multi-reference motion search can effectively improve the rate distortion performance of Wyner-Ziv video coding. Fig. 2.21 shows the result for *Foreman* QCIF sequence. Using five references can improve the coding efficiency by 0.5-1 dB than using one reference, while using ten references does not have further noticeable improvement over using five references. A similar observation can be obtained from other sequences.

The experimental results confirm the above theoretical analysis. Current Wyner-Ziv video coding schemes still fall far behind the state-of-the-art video codecs. A better motion estimator at the decoder is essential to improve the performance.

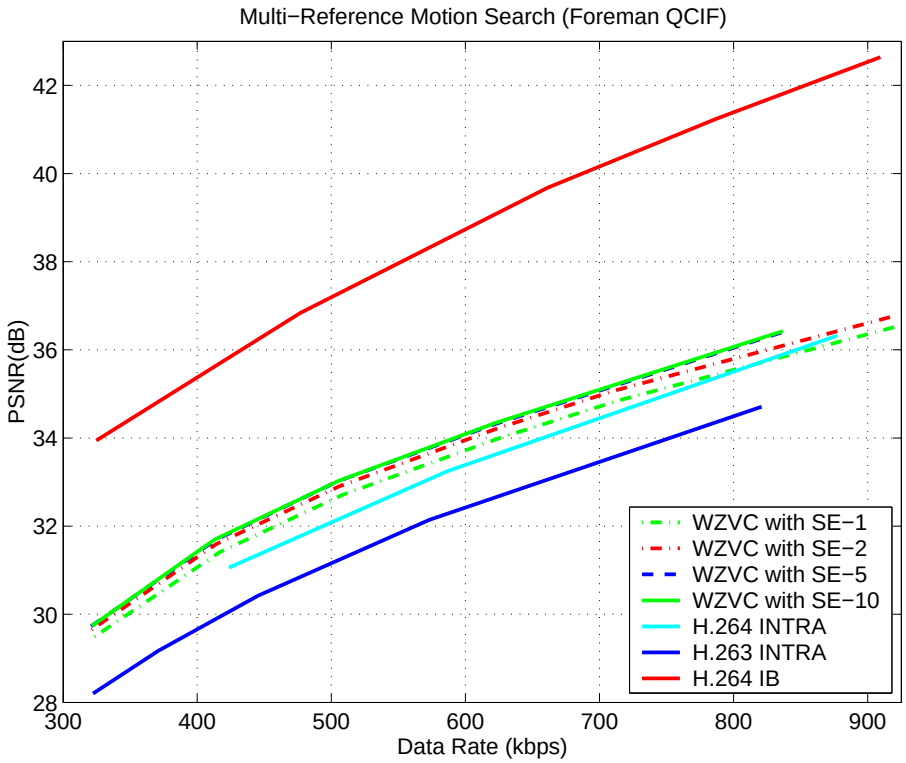


Fig. 2.21. Wyner-Ziv Video Coding with Multi-reference Motion Search (*Foreman* QCIF)

2.4 Wyner-Ziv Video Coding with Universal Prediction

Wyner-Ziv video coding using channel coding methods described above are generally a reverse-complexity system where the decoder has a high burden of complexity. In some scenarios, low complexity at both the encoder and decoder is desirable. For example, wireless handheld cameras/phones belong to the case. To solve the problem, a transcoder can be used as the intermediate part of the system. The usage of the transcoder would increase the cost of the transmission and delay. The goal of this section is to design a Wyner-Ziv video coding approach with low complexity encoder and decoder. To address the problem, we introduce the idea of universal prediction [91, 92]. The definition of universal prediction is stated in [112],

Roughly speaking, a universal predictor is one that does not depend on the unknown underlying model and yet performs essentially as well as if the model were known in advance.

The prediction problem in general can be formulated as to predict x_t based on previous data $x^{t-1} = (x_1, x_2, \cdots, x_{t-1})$. The associated loss function $\lambda(\cdot)$ is used to measure the distance between x^t and the predicted version $z_t = \hat{x}^t$. If the statistic model of the data is well studied, classical prediction theory can be used. However, for natural video sequences, the statistical model is unknown. A universal predictor can be used to predict the future data based on previous data in this case. Merhav and Ziv have shown that in certain cases, a Wyner-Ziv rate-distortion bound can be achieved without binning instead by universal compression in [113].

We use the universal predictor in Wyner-Ziv video coding as the side estimator [91,92]. The replacement of block-based motion estimation side estimator by universal side estimator can reduce the decoder complexity dramatically. Each video frame is formulated as a vector and the pixel values at the same spatial position are grouped as $I(k,l)$, where (k,l) is the spatial coordinate. Denote one sequence of $I(k,l)$ as $X = x_1, x_2, \cdots, x_t$, where t is the temporal index of the sequence. Denote the estimator of X as $Z = z_1, z_2, \cdots, z_t$. The transition matrix from X to Z is denoted as $\prod = \prod(i,j)_{i,j\in[0,255]}$, where $\prod(i,j)$ is the probability when the input is i and the estimator is j. The loss function is denoted as $\Lambda(i,j)$, where i is the input in X and j is the corresponding estimator in Z. Denote the conditional probability on the context as $P(z_t = \alpha | z_1, z_2, \cdots, z_{t-1}$. The optimal estimator z_t is the one minimizing the expected loss. In the extreme case when

$$\Lambda(i,j) = (i-j)^2 \tag{2.20}$$

and

$$\prod(x_t, z_t) = \begin{cases} 1 & x_t = z_t \\ 0 & x_t \neq z_t \end{cases} \tag{2.21}$$

the optimal estimator is a weighted average of the previous occurrence with the same context.

As shown in Fig. 2.22, for each pixel in frame t, we use N previous frames as contexts. In the setup of the experiment, we set $N = 4$ and the context is $(z_{t-4}, z_{t-3}, z_{t-2}, z_{t-1})$. At the decoder, the universal prediction side estimator searches for the occurrence of the context in the previous decoded frames. The optimal side estimator of the current frame is the average of the previous occurrence. For example, suppose the context for current pixel is $(210, 211, 210, 211)$. In the previous frames, the context $(210, 211, 210, 211)$ occurs three times with the following pixel values as 210, 211 and 212 respectively. Therefore, the estimator of the current pixel is $(210 + 211 + 212)/3 = 211$.

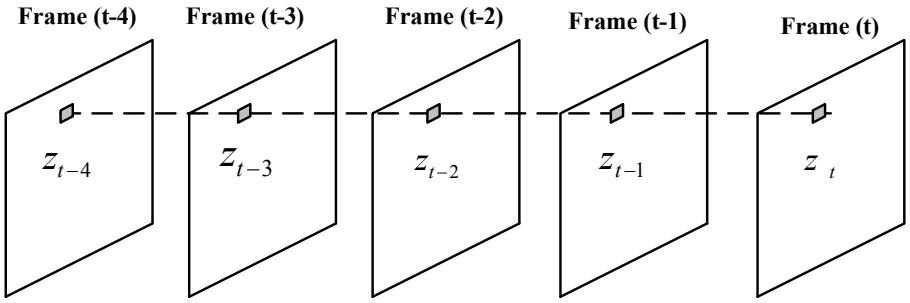


Fig. 2.22. Universal Prediction Side Estimator Context

Fig. 2.23 shows the results for six sequences. We compare the side estimator with universal prediction with the side estimator with block-based motion estimation. We also include the reference frame of H.264 integer motion search for comparison. For all six sequences, the H.264 reference has the best performance among three approaches. Side estimator with motion estimation generally has better quality than side estimator with universal prediction except for the *mobile* sequence. For *coastguard* and *foreman* sequences, the side estimator with motion estimation performs significantly better than the side estimator with universal prediction. In these two sequences, the linear motion model assumption in the side estimator with motion estimation is closely matched. In the other four sequences, the universal prediction side estimator has comparable performance with the motion estimation based side estimator. Con-

sidering the low complexity in the universal prediction side estimator, this method shows great potential. The coding efficiency of Wyner-Ziv video coding using the universal prediction side estimator would be improved if we can refine the correlation model between the original frame and side information.

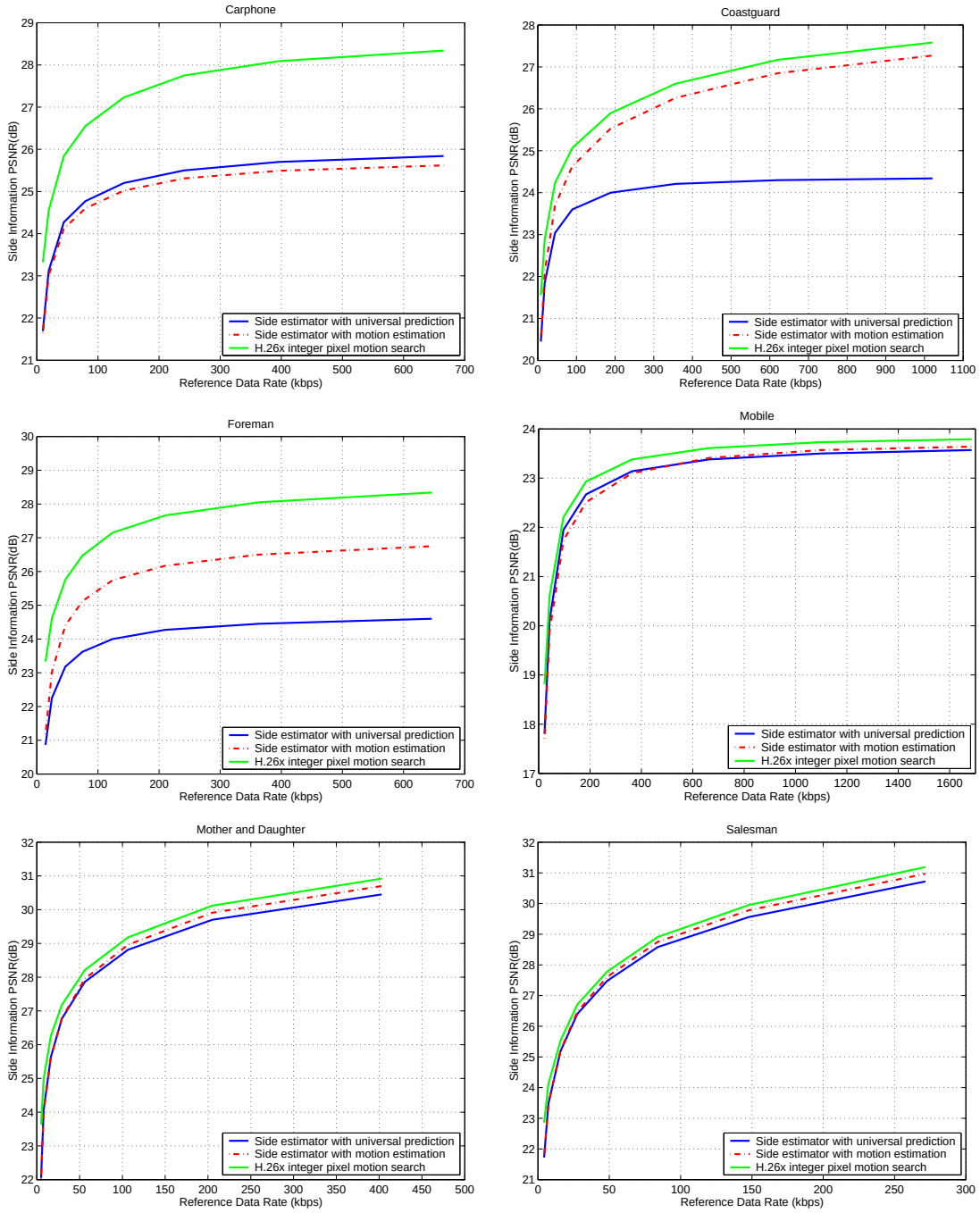


Fig. 2.23. Side Estimator by Universal Prediction

3. BACKWARD CHANNEL AWARE WYNER-ZIV VIDEO CODING

3.1 Introduction

Conventional motion-compensated prediction (MCP) based video compression performs motion estimation at the encoder. A typical encoder requires more computational resources than the decoder. The latest video coding standard H.264 adopted many new coding tools to improve video compression performance and this leads to further complexity increases. While this approach meets the requirements of most applications, it poses a challenge for some applications such as video surveillance, where the encoder has limited power and memory, while the decoder has access to more powerful computational resources. In these applications a simple encoder is preferred with computationally intensive parts left to the decoder.

A Wyner-Ziv video codec generally formulates the video coding problem as an error correction or noise reduction problem. A Wyner-Ziv encoder usually encodes a frame independently using a channel coding method and sends the parity bits to the decoder. Frames encoded this way are referred to as Wyner-Ziv frames. Prior to decoding a Wyner-Ziv frame, the decoder first analyzes the video statistics based on its knowledge of the previously decoded frames and derives the side information for the current frame. This side information serves as the initial estimate, or noisy version, of the current frame. With the parity bits from the encoder, the decoder can gradually reduce the noise in the estimate. Hence the quality of the initial estimate plays an important role in the decoding process. A simple and widely used way to derive the side information is to either extrapolate or interpolate the information from the previously decoded frames as described in Section 2.2.3. The advantage of frame extrapolation is that the frames can be decoded in sequential order, and hence every

frame (except the first few frames) can be coded as Wyner-Ziv frames. However, the quality of the side estimation from the extrapolation process may be unsatisfactory. This has led to research on more sophisticated extrapolation techniques to improve the side estimation. Many Wyner-Ziv coding methods also resort to the use of frame interpolation, which generally produces higher quality side estimates. The problem with interpolation is that it requires some frames, after the current frame in the sequence order, be decoded before the current frame. This means that at least some of these frames, referred to as key frames, cannot be coded as a Wyner-Ziv frame. Instead, they should be coded by conventional methods. Since we need to keep the encoder computationally simple, these frames are often INTRA coded, which costs more data rate than the predictive coding methods. One way to alleviate this problem is to increase the distance between two key frames. However, with the increase in distance, the side estimation quality quickly degrades. The results show that larger key frame distances can only marginally improve the overall coding efficiency, and sometimes even leads to worse coding performance. It is for this reason that many Wyner-Ziv methods code every other frame as Wyner-Ziv frames and key frames alternately.

One concern for Wyner-Ziv video coding is its coding performance compared to state-of-the-art video coding, such as H.264. In conventional video coding, most frames are coded as predictive frames (P Frame) or bi-directionally predictive frames (B Frame) and only very few are coded as INTRA frames due to the large amount of temporal redundancy in a video sequence. INTRA frames consume many more bits than P frames and B frames to achieve identical quality, since they do not take advantage of the temporal correlation across frames. In many Wyner-Ziv video coding schemes, many frames are INTRA coded to guarantee enough side information at the decoder. This inevitably leads to compromise between coding efficiency and encoding complexity.

In this chapter, we address this problem using Wyner-Ziv video coding with Backward Channel Aware Motion Estimation to improve the coding efficiency while main-

taining low-level complexity for the encoder. The rest of this chapter is organized as follows. Section 3.2 gives an overview of Backward Channel Aware Motion Estimation. Section 3.3 discusses the system of Wyner-Ziv video coding with Backward Channel Aware Motion Estimation. Error resilience in backward channel is discussed in Section 3.4. Simulation details and performance evaluations are given in Section 3.5.

3.2 Backward Channel Aware Motion Estimation

Motion-compensated-prediction (MCP) based video coding can efficiently remove the temporal correlation of the video sequence and achieve high compression efficiency. However, motion estimation is highly complex. It is not suitable for power constrained video terminals, such as wireless devices. Wyner-Ziv video coding with Backward Channel Aware Motion Estimation is based on Network-driven motion estimation (NDME) proposed in [114] by Rabiner and Chandrakasan. Network-driven motion estimation was first proposed for wireless video terminals. In NDME the motion estimation task is moved to the decoder.

Fig. 3.1 shows the basic diagram of network-driven motion estimation, which is a combination of motion prediction (MP) and conditional replenishment (CR). Conditional replenishment is a standard low complexity video coding method without motion estimation. It codes the difference between the current frame and the previous frame. We can consider it as the special case of INTER coding with zero motion vector. CR is efficient for reducing the temporal correlation for slow-motion video sequences. For video sequences with high motion, conditional replenishment may not be able to provide high compression efficiency. Motion prediction makes use of the assumption of constant motion in the video sequence. The computationally intensive operation of motion estimation is moved to the high power decoder, which may be at the base station or in the wired network [114]. To estimate the motion vector of frame n at the decoder, motion estimation is performed on the reconstructed frames

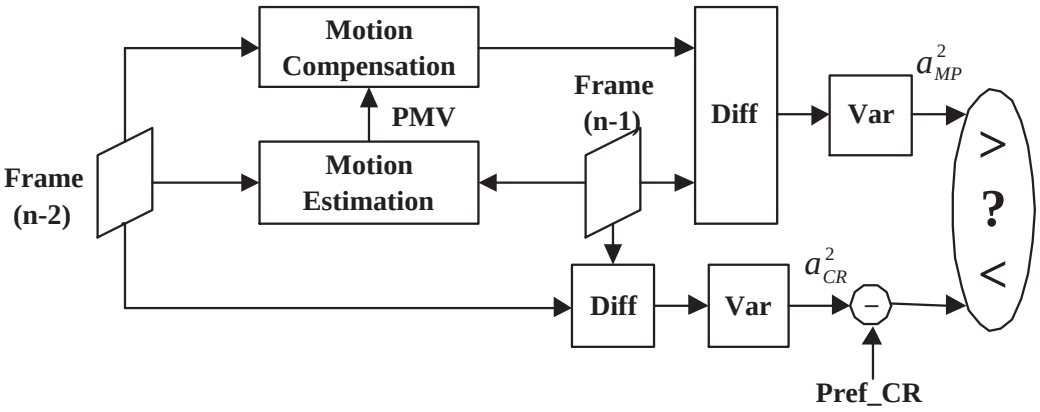


Fig. 3.1. Adaptive Coding for Network-Driven Motion Estimation (NDME)

$(n-1)$ and $(n-2)$ to find the motion vector of frame $(n-1)$. Then the predicted motion vector (PMV) of frame n is estimated by

$$PMV(n) = MV(n-1) \tag{3.1}$$

PMV shows a high correlation with the true motion vector derived at the encoder. CR is a preferable choice for low-motion part since it does not need to send the predicted motion vectors back to the encoder. Therefore, the NDME scheme uses adaptive coding to choose between motion prediction and conditional replenishment. The choice is made at the decoder and hence the adaptive scheme improves the coding efficiency without adding computational complexity at the encoder. Since CR saves bits to send the predicted motion vector, it is a more favorable choice when there is little motion. Denote the variances of MP and CR as a^2_{MP} and a^2_{CR} respectively. MP mode is chosen when

$$a^2_{MP} < a^2_{CR} - \text{Pref\_CR} \tag{3.2}$$

where Pref_CR is a constant bias parameter towards the selection of CR.

Fig. 3.2 shows the flow chart of network-driven motion estimation algorithm. The first two frames are INTRA coded. The motion vector of nth frame is derived from the previous reconstructed frames at the decoder. Then the decoder adaptively chooses

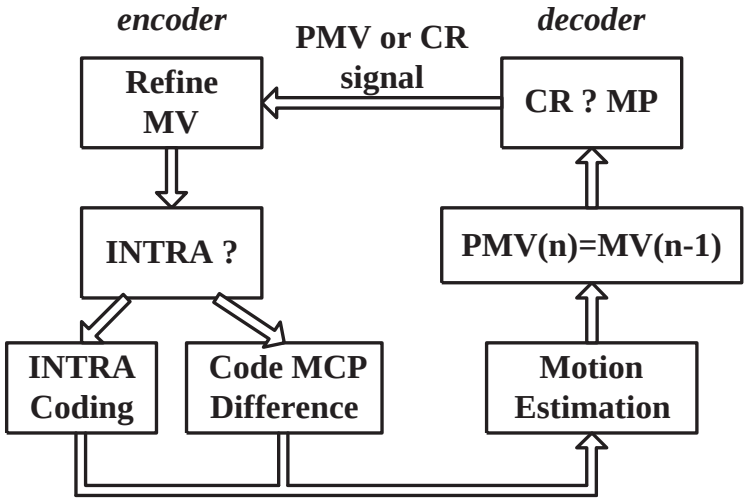


Fig. 3.2. Network-Driven Motion Estimation (NDME)

motion prediction or conditional replenishment. A signal indicating the choice is sent back to the encoder along with the predicted motion vector if MP mode is chosen. No matter which mode is sent, the encoder refines the motion vector to further find the more accurate motion vector. The refinement of the motion vector is performed by searching the ± 1 pixel positions around the received motion vector. If a scene change is detected or the encoder and decoder lose synchronization, an INTRA frame is inserted to refresh the sequence. Experimental results show that NDME can reduce the encoder complexity significantly with a slight increase in the data rate compared with encoder-based motion estimation.

3.3 Backward Channel Aware Wyner-Ziv Video Coding

In this section we extend Wyner-Ziv video coding by coding the key frames with the use of Backward Channel Aware Motion Estimation (BCAME). We shall refer to our Wyner-Ziv video coding method based on BCAME, as BCAWZ. The basic idea of BCAME is to perform motion estimation at the decoder and send the motion

information back to the encoder through a backward channel, which is similar to the idea of NDME in Section 3.2. Hence we are able to improve the coding efficiency of the key frames and the side estimation quality without significantly increasing the encoder complexity.

For natural video sequences, the motion of objects is continuous and adjacent video frames are closely correlated. Therefore, it is possible to predict the motion of the current frame using the information of its adjacent frames. In conventional MCP-based video coding, the current frame is accessible to the encoder and the motion vector is estimated by comparing the current frame with the reference frame. In BCAME, the motion search is performed at the decoder without accessing the current frame. The motion vectors are sent back to the encoder through a feedback channel. For a sequence we encode the first and third frames as INTRA frames. All the other odd frames are coded with BCAME, and we refer to these backward predictively coded frames as BP frame. All the even frames are coded as a Wyner-Ziv frame. A BP frame is coded as follows. Assuming the two BP frames prior to the current BP frame, as shown in Fig. 3.3, have been decoded at the decoder. For each block in the current BP frame, we use its co-located block in one of the previous two BP frames as the source and the other BP frame as the reference. Block based motion search is done at the decoder to estimate the motion vector. The motion vectors are sent back to a motion vector buffer at the encoder through the backward channel that is usually available in most Wyner-Ziv methods. This buffer is updated when the next frame's motion vectors are received. At the encoder, we use the received motion vectors with the previous reconstructed BP frames to generate the motion compensated reference for the current BP frame. The residue between the current BP frame and its motion compensated reference is then transformed and entropy coded. Depending on which of the previous decoded BP frames is used as the source (or the reference) at the decoder, we can at least obtain two sets of motion vectors as shown in Fig. 3.3 and 3.4.

3.3.1 Mode Choices in BCAME

Mode I: Forward Motion Vector

Mode I is shown in Fig. 3.3. Frame A and B are the previous two reconstructed BP frames stored in the frame buffer at the decoder. The temporal distances between adjacent frames are denoted as TD_{AB} and TD_{BC}. To find the motion vector in the current macroblock, we use the motion information of the colocated macroblock in the previous frame by assuming that a constant translational motion velocity remains across the frames. For each block in the current frame, we use the co-located block in B and search for its best match in A and obtain the forward motion vector MV_F. Assuming linear motion field, the motion vector for the block in the current BP frame is then

$$MV_I = \frac{TD_{BC}}{TD_{AB}} MV_F$$

Since we code the BP frame and Wyner-Ziv frame alternately, $TD_{AB} = TD_{BC} = 2$, $MV_I = MV_F$.

Mode II: Backward Motion Vector

The second mode is obtained in a similar way as Mode I but we use the co-located frame in A as the source, as shown in Fig. 3.4. We search for the best matched block in B. This motion vector is referred to as the backward motion vector MV_B. Again assuming linear motion, the motion vector for the current frame is

$$MV_{II} = -\frac{TD_{AC}}{TD_{AB}} MV_B$$

Here $TD_{AB} = 2$ and $TD_{AC} = 4$, $MV_{II} = -2MV_B$.

Mode III: Encoder-Based Mode Selection

These two sets of motion vectors are sent back to the encoder, where the original current BP frame is available. The encoder can then do a mode selection to choose

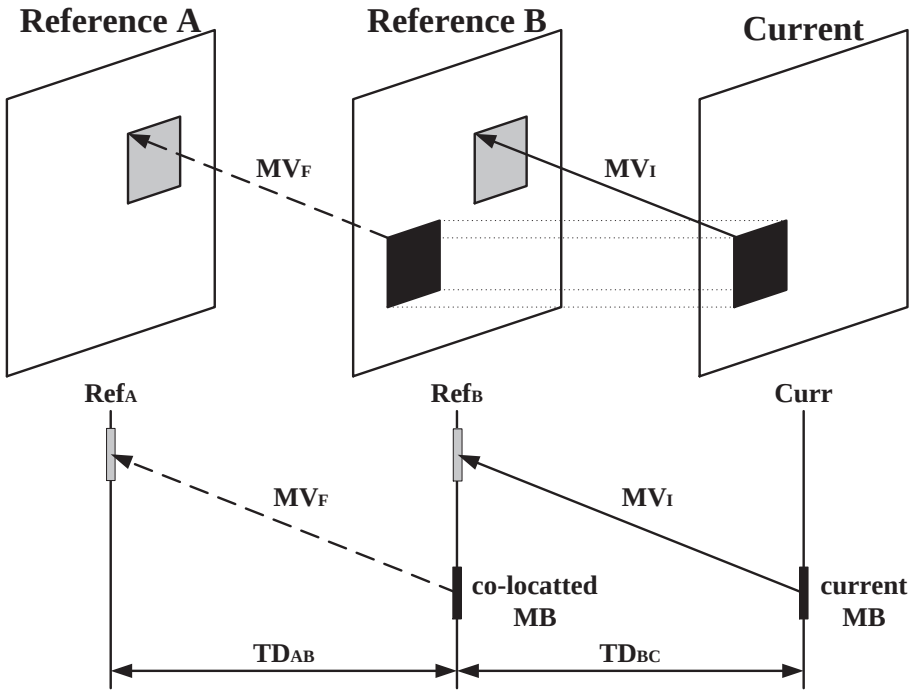


Fig. 3.3. Mode I: Forward Motion Vector for BCAME

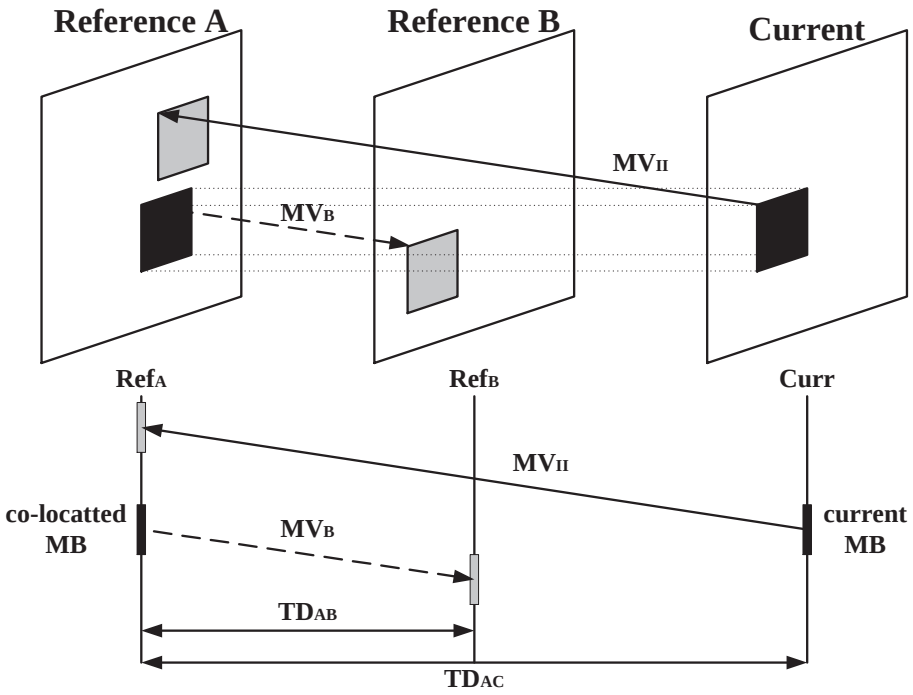


Fig. 3.4. Mode II: Backward Motion Vector for BCAME

the best matched motion vector based on metrics such as mean squared error (MSE) or sum of absolute difference (SAD).

$$\text{Optimal Mode} = \arg \min_{k \in \{I, II\}} \frac{\sum_{(i,j)} D[x(i,j) - \hat{x}^{(k)}(i,j)]}{N \times N}, \tag{3.3}$$

where k denotes the index of the modes, $x(i,j)$ denotes the original pixel value in the position (i,j), $\hat{x}^{(k)}(i,j)$ denotes the reconstructed pixel value using mode k, N represents the size of the macroblock, and the summation is over all the pixels of the current macroblock. According to this measurement of fidelity, we obtain the optimal mode with highest peak signal-to-noise ratio (PSNR). The mode decision result is sent to the decoder along with the transform coefficients of the current BP frame. We refer to this mode as Mode III. Although this mode selection scheme uses equation (3.3) to make the decision and thus places more computational load on the encoder compared to using Mode I or Mode II only, the experimental results shown in Fig. 3.6 and 3.7 prove that it is worth the additional work.

Compared to other Wyner-Ziv video coding methods, BCAWZ provides an efficient approach to predictively code the key frames without greatly increasing the encoder complexity. We note that since these two motion vectors are also needed to generate the interpolated side estimate at the decoder for the Wyner-Ziv frame between A and B, hence the increase of the decoder's complexity is marginal.

3.3.2 Wyner-Ziv Video Coding with BCAME

The key idea behind BCAWZ is to INTER-code the key frames that were INTRA-coded without significantly increasing the encoder's computational complexity. A Wyner-Ziv video coding scheme with BCAME is shown in Fig. 3.5. For a BP frame, after the motion vector is received from the decoder, the motion-compensated reference is generated and the residual frame is obtained. This residual is transformed and entropy coded as in H.264. Wyner-Ziv frames are also coded in the transform domain. Every Wyner-Ziv frame is coded with the integer transform proposed in H.264 [1]. Each coefficient is then represented by 11 magnitude bits and a sign bit. The bits

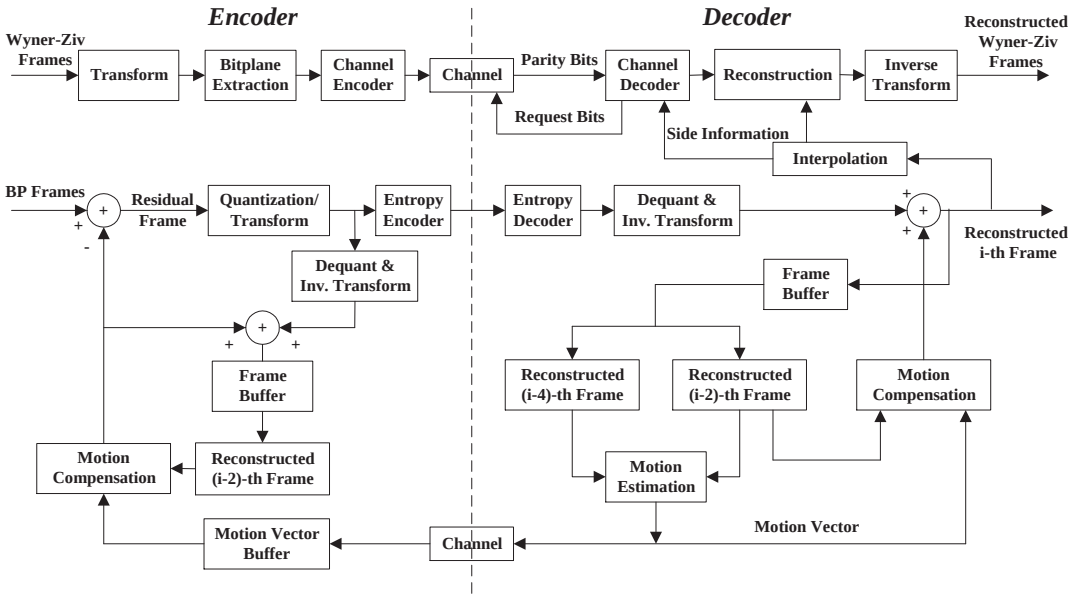


Fig. 3.5. Backward Channel Aware Wyner-Ziv Video Coding

at the same bitplanes are coded with a low-density parity-check (LDPC) code. The parity bits for each bitplane are sent to the decoder in the descending order of bit significance. At the decoder, a side estimate is generated by interpolating the adjacent key frames with the motion search. The side estimate is then transform coded and represented by bitplanes. Parity bits generated from the corresponding bitplanes at the encoder are used to correct the errors in each bitplane, also in the descending order of bit significance, until the total rate budget for this frame is achieved. We should note that the backward channel is connectionless, which transmits the motion vectors from the decoder to the encoder. The encoder does not provide any feedback to the decoder upon receiving the motion vectors.

3.4 Error Resilience in the Backward Channel

When video data is transmitted over the network, error-free delivery of the data packets are typically unrealistic due to either traffic congestion or impairments of the physical channel. The errors can also lead to de-synchronization of the encoder

and the decoder. Error resilient video coding techniques [115] have been developed to mitigate transmission errors. Conventional motion-compensated prediction (MCP) based video coders, such as H.264 and MPEG-4, include several error resilience methods [14, 116–118]. Methods to address the problem of forward channel errors have been extensively studied. We now consider the scenario when the backward channel is subjected to only erasure errors or delays. The case of a purely noisy backward channel is not considered here. We also assume the backward channel is one way and connectionless. Since the motion vectors sent back to the encoder play a crucial role in predictive coding, it is important to make sure the motion vectors are resilient to transmission errors and delays. In an error free coding scenario, the decoder sends the motion vectors of the i-th frame, denoted as $\mathrm{MV_i}$. The encoder receives $\mathrm{MV_i}$ and generates the residual frame $\mathrm{RF_i}$ and sends the bitstream through the forward channel. At the decoder, the decoder reconstructs the frame by the received $\mathrm{RF_i}$ and the stored motion vector $\mathrm{MV_i}$. This changes when there is an erasure error or delay at the backward channel. In this case, the motion vector is not updated and the encoder continues to use the motion vectors of the $(i-2)$-th frame. The encoder generates the residual frame denoted as $\mathrm{RF_i'}$ with $\mathrm{MV_{i-2}}$. The decoder reconstructs the frame with the residual data $\mathrm{RF_i'}$ and the motion vector $\mathrm{MV_i}$. Thus the reconstructed frames at the encoder and the decoder lose synchronization, which causes the drift problem and can propagate to the rest of the sequence.

To address this problem, a two-stage adaptive coding procedure is proposed. We first propose a simple resynchronization method. A synchronization marker is used to provide a periodic synchronization check. An entry denoting the index of the frame is inserted before sending the motion information in the bit stream. The bandwidth needed to send this extra field is negligible. With this index, the encoder can quickly report an error when the received index does not match the index at the encoder.

The encoder then codes the key frames adaptively based on the decision of the synchronization detector. When desynchronization is detected, the encoder ignores the motion information and codes the key frames as an INTRA frame. This frame

type selection decision is sent to the decoder and the decoder decodes this frame as an INTRA frame. For the next key frames, the decoder continues to send the motion vectors back. After synchronization is reestablished, the encoder resumes coding the key frames as BP frames.

3.5 Experimental Results

We implemented our scheme based on the H.264 reference software JM8.0 [119]. The results of *Foreman*, *Coastguard*, *Carphone*, and *Mobile* QCIF sequences are used form our experiments. Each sequence consists of 300 frames at 30 frame/second. The rate-distortion (R-D) results include both the key frames and Wyner-Ziv frames. The objective visual quality is measured by the peak signal to noise ratio (PSNR) of the luminance component.

Fig. 3.6-3.9 show the R-D performances of the four sequences. We first compare the results of BCAWZ with Mode I and Wyner-Ziv video coding with INTRA coded key frame. It is shown that by using BCAWZ, the performance can improve by 1 - 2dB. The improvement is less significant at the high data rate than the low data rate. This is because at the low data rate the video quality is more dependent on the reference quality. With the mode selection in Mode III, the performance can be further improved by 0.5-2 dB. When BCAWZ is compared with conventional video coding, we find that BCAWZ can achieve 4-5 dB gain over H.264 INTRA coding. However, compared with the state-of-the-art predictive coding, BCAWZ still trails H.264 by as much as 4-6 dB, where the H.264 results are coded with the I-B-P-B-P frame structure with quarter pixel motion search and only the first frame is INTRA coded. Generally the performance of BCAWZ is better for slow motion sequences, such as *Carphone* sequence, where the motion vectors of the neighboring frames are continuous and the correlation of the neighboring motion vectors are higher. Fig. 3.10 shows an example of comparisons of BCAWZ (mode III) and WZ with INTRA key frames. The test sequence is *Foreman* CIF sequence with 30 frames/second and both

are coded at 511 kbits/second. The frames shown here are the 20th frame, which is a key frame in both scenarios. Fig. 3.10-(a) is an INTRA coded key frame with PSNR 28.6552 dB. Fig. 3.10-(b) is a BP frame with PSNR 34.6404 dB. The average PSNR difference for the entire sequence is 3.5 dB.

Video sequences contain both spatial and temporal redundancy. In INTRA coding, only spatial redundancy is reduced for the sequences. Backward channel aware motion estimation removes the temporal correlation as INTER coding does, which achieves better coding efficiency compared to INTRA coding. Using BCAME, BCAWZ can achieve 1-3 dB gain on top of Wyner-Ziv video coding schemes with INTRA key frames [29] [106]. However, BCAWZ contains some discontinuous motion artifacts, such as some small blockiness.

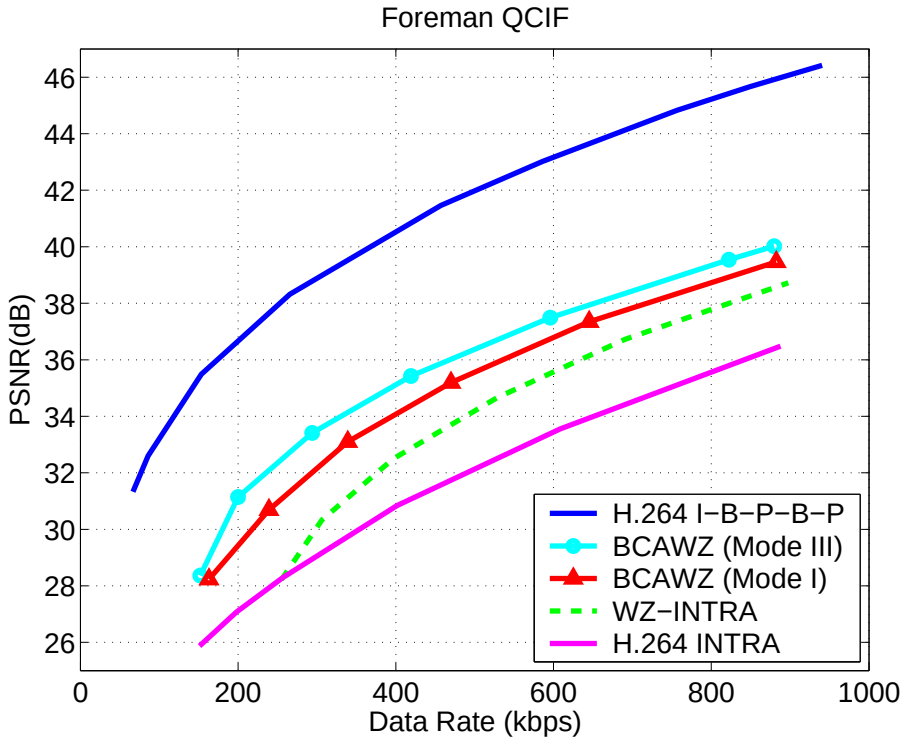


Fig. 3.6. BCAWZ: R-D Performance Comparison (*Foreman* QCIF)

Fig. 3.11 shows the backward channel bandwidth as a percentage of the forward channel bandwidth. For both sequences with Mode I, the backward channel band-

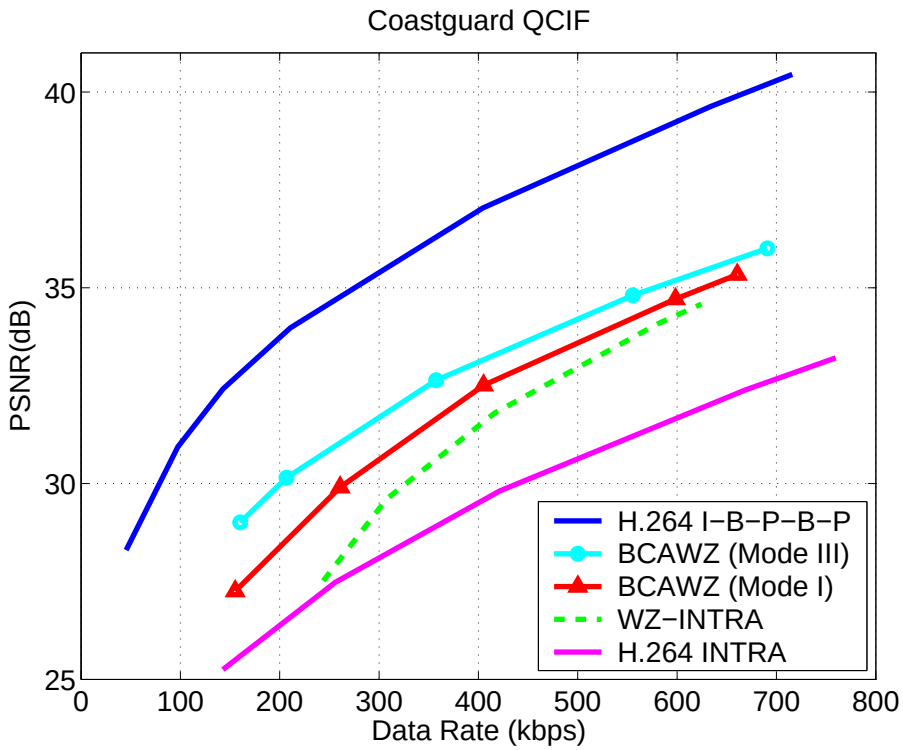


Fig. 3.7. BCAWZ: R-D Performance Comparison (*Coastguard* QCIF)

width is 5-8% of the forward channel at lower data rate. This percentage reduces below 3% for mid to higher data rate. To use the backward motion vectors, Mode III needs roughly twice as much backward channel bandwidth as Mode I. Such backward channel usage can be readily satisfied in many communication systems.

Practical bandlimited channels generally suffer from various types of degradations such as bit/erasure errors and synchronization problems. In the following we study the error resilience performance for the backward channel. We assume that there is no channel error while sending the parity bits. We test the case when the motion vectors for the 254th frame in the *Foreman* sequence are delayed by two frames. Without the signal of frame number, the encoder does not update the motion vector buffer and still uses the motion vectors from previous frames. We also test a one-frame motion vector loss in the 200th frame in the *Coastguard* sequence. In this scenario, the motion vectors of the 200th frame are all lost and the encoder still uses the motion vectors

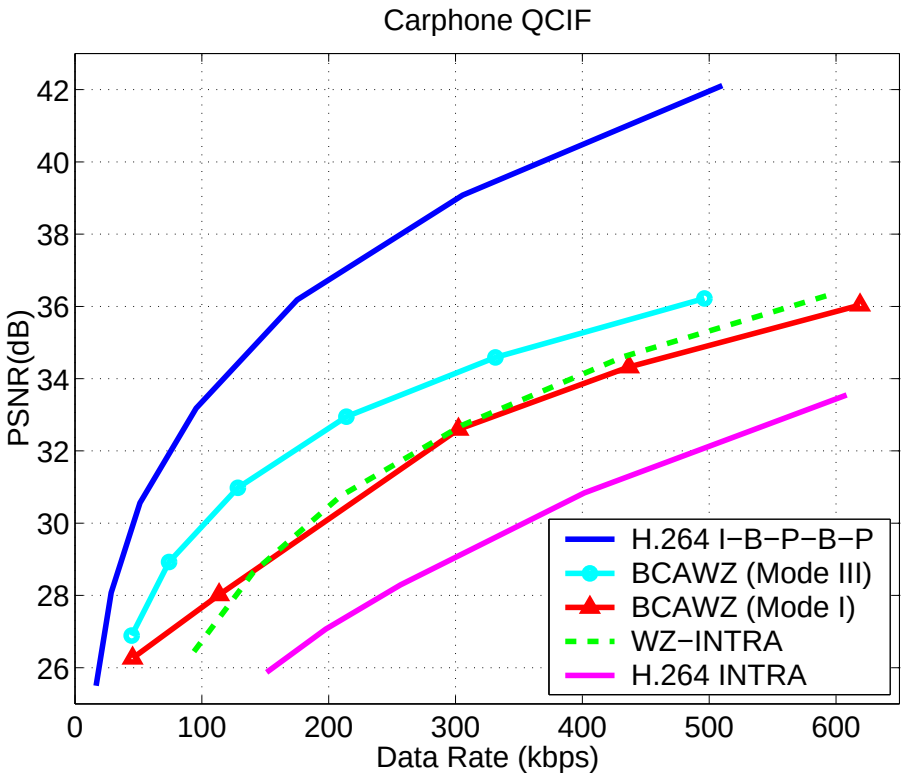


Fig. 3.8. BCAWZ: R-D Performance Comparison (*Carphone* QCIF)

at the buffer from the previous frame without synchronization detection. In the experiment we observe that when a delay or erasure occurs, the video coding efficiency without error resilience sharply drops. The quality degradation continues until the end of the sequence even though the motion vectors of the following BP frames are correctly received. This is because, as described in Section 3.4, when a delay occurs, the encoder uses a different set of motion vectors, hence the reconstructed frame at the encoder is different from the decoder. As this reconstructed frame is used as reference for the following frames, the drift propagates across the sequence. If there are more than one-frame motion vectors loss or delay, the desynchronization problem would be even worse due to the mismatch of the reconstructed reference frames. In contrast, the adaptive coding scheme can detect the desynchronization and insert an INTRA frame, hence stopping the drift propagation. In Fig. 3.12 and Fig. 3.13, the

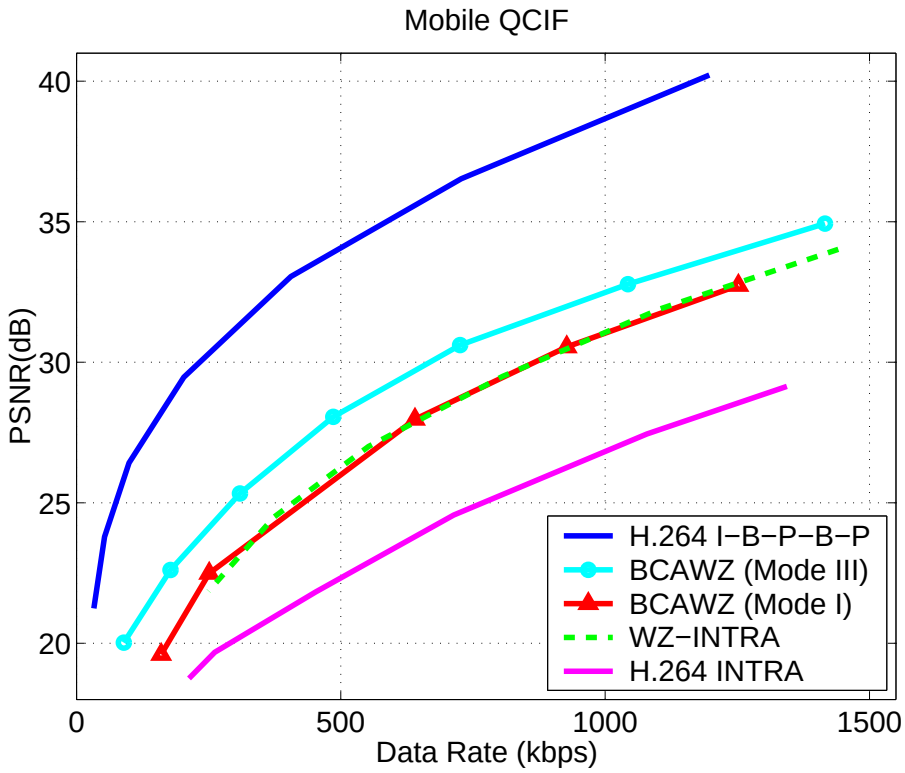


Fig. 3.9. BCAWZ: R-D Performance Comparison (*Mobile* QCIF)

(a) 20th Frame (WZ with INTRA Key Frames)

(b) 20th Frame (BCAWZ)

Fig. 3.10. Comparisons of BCAWZ and WZ with INTRA Key Frames at 511KBits/Second (*Foreman* CIF)

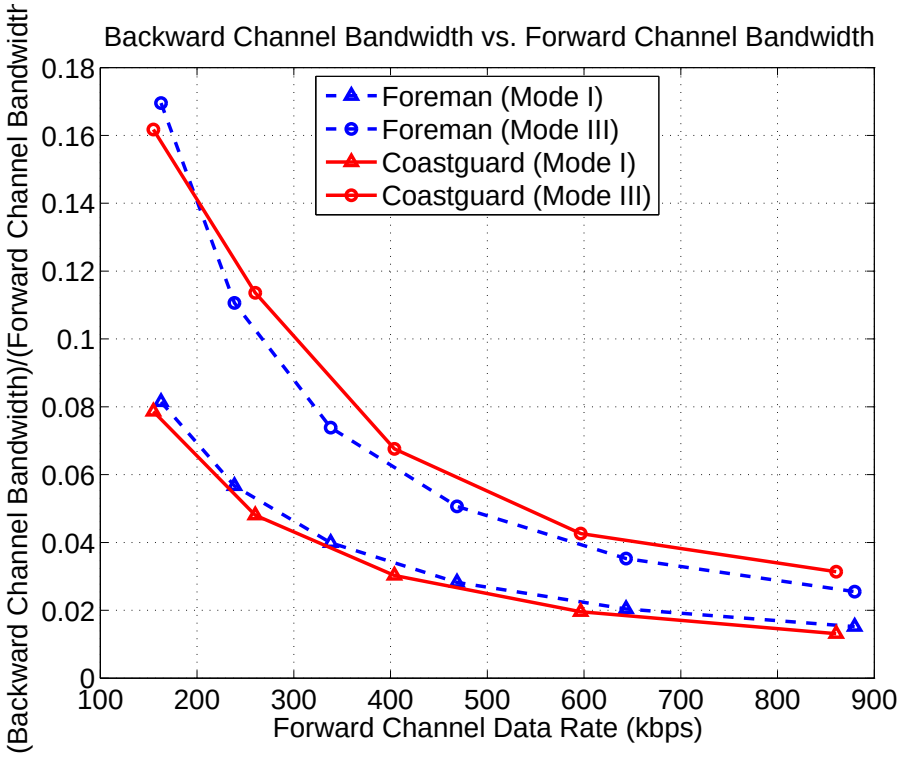


Fig. 3.11. Backward Channel Usage in BCAWZ

proposed error resilience scheme can effectively recover from backward channel delay or erasure errors. It only incurs a very small penalty in terms of R-D performance compared with the results with an error free backward channel.

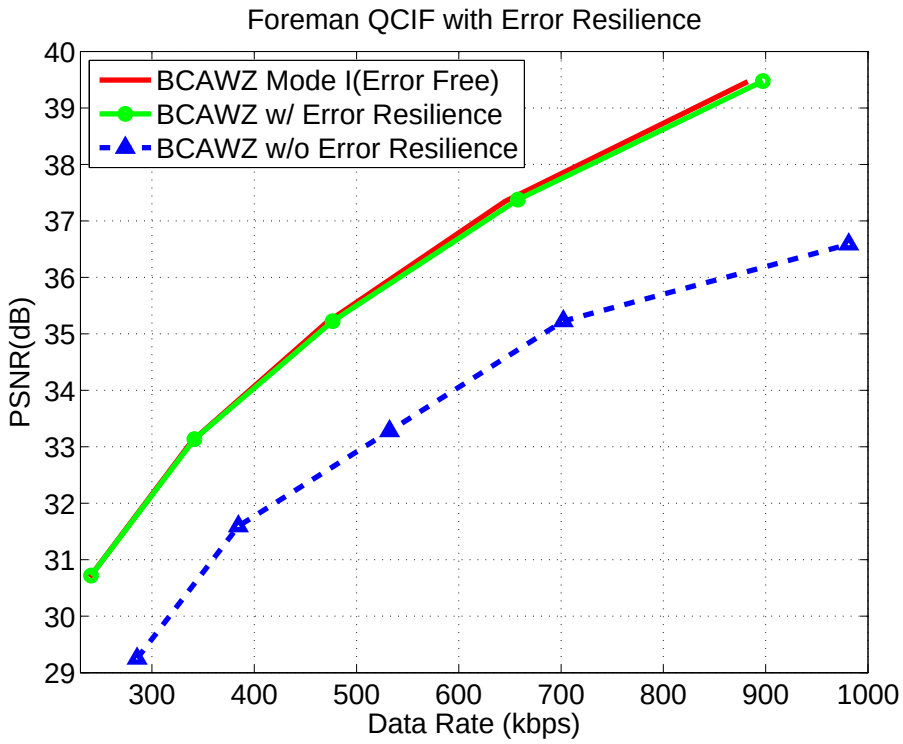


Fig. 3.12. R-D Performance with Error Resilience (*Foreman* QCIF) (Motion Vector of the 254th Frame is Delayed by Two Frames)

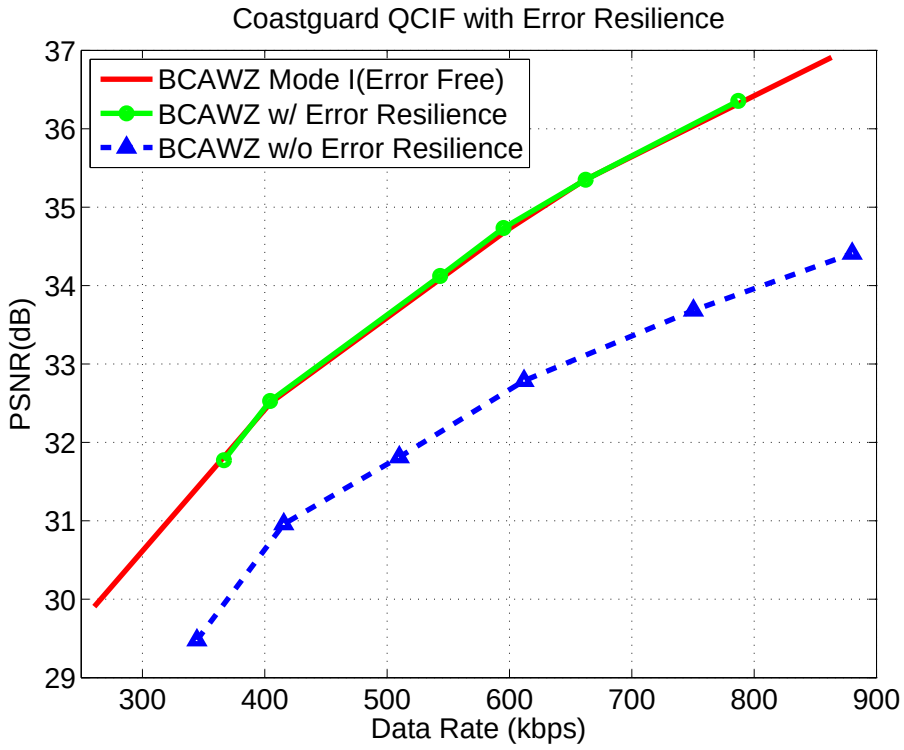


Fig. 3.13. R-D Performance with Error Resilience (*Coastguard* QCIF) (Motion Vector of the 200th Frame is Lost)

4. COMPLEXITY-RATE-DISTORTION ANALYSIS OF BACKWARD CHANNEL AWARE WYNER-ZIV VIDEO CODING

4.1 Introduction

Many Wyner-Ziv video coding schemes encode a video sequence into two types of frames, key frames and Wyner-Ziv frames. Key frames are encoded using conventional video coding methods such as H.264 and Wyner-Ziv frames are encoded using channel coding techniques [30, 49]. At the decoder, the reconstructed key frames serve as side information used to reconstruct the Wyner-Ziv frames.

In Chapter 3, we presented a WZVC scheme that uses Backward Channel Aware Motion Estimation (BCAME) to encode the key frames, where motion estimation was performed at the decoder and motion information was sent back to the encoder. We refer to these backward predictively coded frames as BP frames. As shown in Chapter 3, the coding efficiency of a Wyner-Ziv video codec significantly depends on the coding efficiency of the key frames. This leads to the need for complexity-rate-distortion (CRD) optimization for applications where the encoder is subject to limited computational resources.

Complexity constrained rate distortion analysis and optimization for video coding has been of interest in the research community. In [120], CRD analysis for a motion-compensated prediction (MCP) based video encoding is developed by modeling the complexity and rate-distortion tradeoff for different encoding parameters. Another MCP-based CRD analysis is proposed in [121] by considering the complexity of different types of mode selection. In [122], CRD analysis is proposed for a wavelet based video encoder by modeling the complexity for different spatiotemporal decomposition structures.

In this chapter we present a model to quantitatively analyze the complexity and rate-distortion tradeoff for BP frames. Three estimators, the minimum estimator, the median estimator and the average estimator, are proposed and analyzed.

4.2 Overview of Complexity-Rate-Distortion Analysis in Video Coding

Rate-distortion (R-D) analysis has been studied in the research community for a long time. Recently the complexity factor is taken into consideration, which is closely related to the power consumption for the video coding terminals. In a power constrained device, complexity management is veryimportant to maintain the durability and efficiency. Frameworks to measure the complexity are proposed for different scenarios [120–124]. 3D models to study the relationship between rate, distortion and complexity are presented to optimize the tradeoff among the three factors. In this section we give an overview of complexity-rate-distortion analysis.

4.2.1 Power-Rate-Distortion Analysis for Wireless Video Communication

In wireless video communication, power constraint is a challenge to the battery-operated devices. A power constrained rate-distortion analysis is done in [125]. The power considered includes video processing power as well as the power consumption in channel coding and transmission. The joint power control and bit allocation schemes can achieve significant power saving compared with schemes without optimization.

Another framework to study the power-rate-distortion (P-R-D) relationship is presented in [120]. Generally wireless appliances operate with limited energy, hence it is essential to manage the energy consumption. The model evaluates P-R-D behavior of the conventional video coding system. Guided by the model, the system is able to automatically configure the control parameters given a power supply level. The power here refers to video processing power only.

The typical video encoder has three major modules: ME (motion estimation), PRECODING, and ENC (entropy encoding). The data representation module PRE-

CODING includes DCT, IDCT, QUANT (quantization), DQUANT (inverse quantization) and RECON (reconstruction) modules.

The complexity control parameter set consists of three parameters $\{x, y, z\}$. The complexity of ME is determined by the number of SSD/SAD performed every frame. The ME control parameter x is defined as:

$$x = \frac{\lambda_{ME}}{\lambda_{ME}^{max}} \tag{4.1}$$

where λ_{ME} denotes the number of sum of absolute difference (SAD) computed per frame and λ_{ME}^{max} is the maximum value of λ_{ME}. The complexity of PRECODING is determined by the number of nonzero MBs. The PRECODING control parameter y is defined as:

$$y = \frac{\lambda_{PRE}}{M} \tag{4.2}$$

where λ_{PRE} denotes the number of nonzero macroblocks in the frame and M is the total number of macroblocks in a frame. The complexity of ENC is determined by the data rates. The other control parameter z is defined as:

$$z = \frac{\lambda_F}{f_{max}} \tag{4.3}$$

where λ_F is the encoding frame rate and f_{max} is the maximum frame rate with a default value of 30 frames/s. The complexity is transformed to power for measurement.

The final R-D optimized power control problem is formulated as:

$$\begin{aligned} \min_{\{x,y,z\}} D_v(R; x, y, z) &= (1-z)^2(\beta_0 + \beta_1) \\ &+ 2(2z - z^2)(\beta_0 + \beta_1 e^{-\beta_2 x}) \times [\frac{1}{2}(1-y)^2 + y(1 + a_0 y) \cdot 2^{-2\gamma \frac{R}{y}}] \\ s.t. \Phi(P) &= z(C_1 x + c_2 y + C_3 R) \end{aligned} \tag{4.4}$$

where $D_v(R; x, y, z)$ denotes the video presentation quality that is closed related to the distortion, R denotes the coding data rate in the unit of bits per pixel, P denotes power consumption, and γ is a model constant related to the distortion. $\Phi(P)$ defines the relationship between the power consumption and the complexity control

parameters with constants C_1, C_2, and C_3. ME model parameters β_0, β_1, and β_2 are estimated from the statistics of previous frames. Since the typical operational time of the power supply is at least several hours, the power control parameters are re-evaluated and adjusted every few seconds or longer.

4.2.2 H.264 Encoder and Decoder Complexity Analysis

There have also been several decoder-base complexity-rate-distortion analysis. The initial result to analyze the computational complexity of H.26L decoder is presented in [126]. Interpolation for the sub-pixel positions and deblocking filter are reported as the most time consuming processes. Platform-independent optimization of the two units is also proposed. A comprehensive complexity analysis for H.264 baseline profile is presented in [123]. The H.264 decoder is shown in Fig. 1.3 which consists of a series of sub-functions. After entropy decoding, different types of syntax elements in the bitstream are de-multiplexed. Important header information is extracted to determinate the mode. The residual coefficients are obtained after inverse quantization and inverse transform. Depending on the mode, a spatial prediction or temporal prediction is generated from previous reconstructed pixel values. It is added to the residue to form the reconstructed value for the current block. A table is formed to study the frequency of occurrence for every sub-function. Another table is established to compute the numbers of basic operation units (addition, multiplication, shift) for every sub-function. The complexity of the decoder is measured by the number of basic operation units. Using this metric, the decoder complexity of H.264 is approximately two to three times that of H.263.

A joint evaluation of performance and complexity for H.264 is presented in [127]. The processing time is used as the metric of complexity. Both the encoder and the decoder complexity are measured. The memory access frequency (total number of data transfers from/to memory per second) and the memory usage are also considered. Eighteen different configurations are set up for the experimental results, where the

search range of motion estimation, block size, rate-distortion optimization method, and entropy coding method are defined for every configuration. The complexity is normalized by the simplest configuration and hence complexity measurement is platform-free. Compared with previous standards (H.263 and MPEG-4), the decoder complexity of H.264 increases by a factor of two and the encoder increase is larger than one order of magnitude.

A model to compute the complexity of decoding for a generic system is discussed in [124]. Two complexity-function variables are defined to model the decoding complexity, the percentage of nonzero transform coefficients and the percentage of decoded motion vectors out of the maximum number of possible motion vectors. The complexity of entropy decoding and inverse transform is related to the first variable. The complexity of motion compensation and fractional-pixel interpolation is related to the second variable. The estimation of computational complexity is quantified with the number of integer addition and multiplications per pixel. With the model, the multimedia receiver can adaptively negotiate with the server for a bitstream with a specific decoding complexity.

4.2.3 Complexity Scalable Motion Compensated Wavelet Video Encoding

There is also research on complexity analysis of wavelet based video coding. A comprehensive framework for the analysis of wavelet-based video encoding complexity is presented in [122]. The complexity is measured in terms of the number of motion estimation (ME) computations. Since motion-compensated temporal filter (MCTF) wavelet video coding schemes have inherent scalability, a closed-form expression with different coding options is formulated. The expression shows that the complexity is related to the layers(depth) of wavelet decomposition. Using the formulation, the complexity under specific settings can be predicted independent of the video content. Therefore, it can be used to optimize the tradeoff between rate, distortion

and complexity. The optimized system can achieve significant reduction of complexity compared to full implementation with an insignificant penalty in rate-distortion performance.

4.3 Backward Channel Aware Wyner-Ziv Video Coding

As discussed in Chapter 3, we propose a Wyner-Ziv video coding with backward channel aware motion estimation (BCAME). We show the basic coding diagram in Fig. 4.1. The basic idea of BCAME is to perform motion estimation at the decoder and send the motion information back to the encoder through a backward channel. The motion information is used to encode the key frames at the encoder. The frames coded in this way are referred to BP frames. Hence, we can exploit the temporal correlation of the video sequence while shifting the motion estimation task from the encoder to the decoder. Wyner-Ziv frames are coded in the transform domain using channel codes, such as turbo codes or low-density-parity-check (LDPC) codes. Only part of the parity bits or syndrome bits are sent to the decoder. At the decoder, the side information is derived from the previously decoded key frames. The derived side information is used as the initial estimate to decode the Wyner-Ziv frames with the parity bits received from the encoder. The derivation of side information generally involves motion estimation and the motion information can be used for BCAME, hence the additional computational complexity at the decoder is marginal.

A BCAME encoder can generate the motion compensated frames using the motion vectors received by the backward channel. If several motion vectors are sent to the encoder, the encoder can make a mode decision based on the distortion and choose the best available motion vector. For BP frames, the residual frame between the original frame and motion compensated reference frame is encoded in the similar way as an H.264 encoder. Compared to the INTRA coded frames, BP frames can significantly improve the coding efficiency with minimal usage of the backward channel. Compared

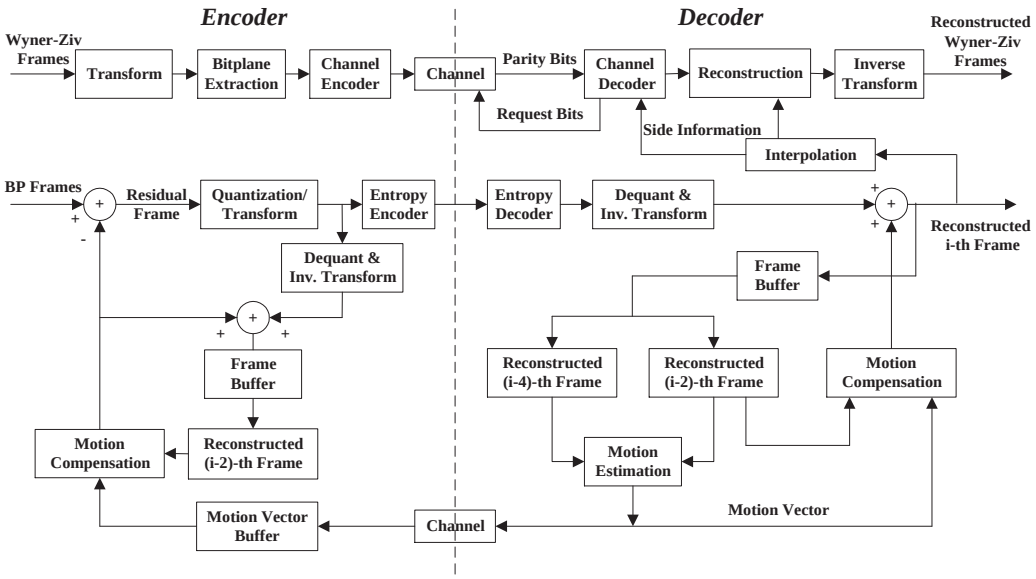


Fig. 4.1. Backward Channel Aware Wyner-Ziv Video Coding

to the P frames, BP frames moves the computationally intensive task of motion estimation to the decoder and results in a reduction of complexity at the encoder.

The computational complexity of a Wyner-Ziv video encoder includes that of Wyner-Ziv frames and BP frames. Since the complexity of Wyner-Ziv frames at the encoder are relatively insignificant, we will focus on the analysis of the BP frames. The computational complexity of BP frames can vary depending upon the number of candidate motion vectors received at the encoder. When N motion vectors are received, the encoder will need to compare the distortions resulted from these N motion vectors. In terms of computational complexity, this is comparable to a motion search when there are only N candidate motion vectors in the search area, whereas the traditional P frames require a motion search for every pixel and sub-pixel inside its search window.

4.4 Complexity-Rate-Distortion Analysis of BP Frames

4.4.1 Problem Formulation

Assume there are N motion vectors estimated at the decoder. Denote these N motion vectors as MV_1, MV_2, $\cdots$, MV_N. We assume that the N motion vectors are 2-D independent and identically distributed (i.i.d.) random variables [128] having the joint probability density function (pdf) $f(x, y)$ and the cumulative probability distribution function (cdf) $F(x, y)$. Denote the true motion vector as $MV_T = (x_n, y_n)$. A true motion vector is an ideal motion vector with minimum mean squared error between the original frame and the reference frame.

Signal power spectrum and Fourier analysis tools are used to analyze the rate distortion performance in [129]. We extend the idea to apply it for Wyner-Ziv video coding. As shown in [90, 110] the rate difference between two coders using different motion vectors is,

$$\Delta R_{1,2} = \frac{1}{8\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \log_2 \frac{1 - e^{-\frac{1}{2}\omega^T \omega \sigma^2_{\Delta_1}}}{1 - e^{-\frac{1}{2}\omega^T \omega \sigma^2_{\Delta_2}}} d\omega \tag{4.5}$$

where $\sigma^2_{\Delta_i}$ $(i = 1, 2)$ are the variances of the error motion vectors. The error motion vector is the difference between the derived motion vector and the true motion vector. In the following we analyze three different motion estimators, namely, the minimum motion estimator, the medium motion estimator, and the average motion estimator.

4.4.2 The Minimum Motion Estimator

In this case, all N i.i.d. motion vectors are sent to the encoder. At the encoder, each of the N motion vectors is used with the reference frame for motion compensation. The motion vector leading to the minimum distortion between the original frame and motion compensated reference frame is selected. The minimum motion estimator matches Mode III as described in Section 3.3.1, where $N = 2$. As in many traditional fast motion search methods, we assume the motion field is homogeneous and unimodal, then the motion search is equivalent to choosing the motion vector nearest to the true motion vector. Hence the corresponding error motion vector is $(X - x_n, Y - y_n)$, where X and Y are the horizontal and vertical motion vectors. We use capital letters to denote a random variable unless otherwise specified.

We introduce a new random variable Z to model the distance between the received motion vector and the true motion vector,

$$Z = \sqrt{(X - x_n)^2 + (Y - y_n)^2} \tag{4.6}$$

Hence the problem can be formulated as searching for the motion vector with the smallest Z.

This problem can be solved using order statistics and extreme value theory [130, 131]. More specifically, let $X_1, X_2, \cdots, X_n$ be i.i.d. random variables. Denote $X_{(1)}, X_{(2)}, \cdots, X_{(n)}$ as the corresponding order statistics, where the first order statistic $X_{(1)}$ is the minimum of $X_1, X_2, \cdots, X_n$. The probability density function of the kth order statistic can be formulated as [130]

$$f_{X_{(k)}}(x) = \frac{n!}{(k-1)!(n-k)!} F(x)^{k-1}[1 - F(x)]^{n-k} f(x) \tag{4.7}$$

The distribution of the motion vectors are highly dependent on the motion estimation method and the content of the sequence. We now consider the case when the received motion vectors have a 2-D Gaussian distribution as used in [18, 129],

$$f_{X,Y}(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-x'_n)^2+(y-y'_n)^2}{2\sigma^2}} \tag{4.8}$$

where σ^2 is the motion vector variance, x'_n and y'_n are the mean of the horizontal and vertical motion vectors respectively. This assumption matches the study in [132, 133] that shows the distribution of the motion vectors have a small variance at the center area. To facilitate our discussion, we denote the deviation from the true motion vector as $\delta_x = x'_n - x_n$ and $\delta_y = y'_n - y_n$.

Case I: $\delta_x \neq 0$ **or** $\delta_y \neq 0$

In this case the means of the motion vectors sent back from the decoder are different from the true motion vector. The random variable Z as defined in (4.6) is a Rician distribution,

$$f_Z(z) = \frac{z}{\sigma^2} \exp(-\frac{z^2+\nu^2}{2\sigma^2}) I_0(\frac{\nu x}{\sigma^2}) \quad z \geq 0 \tag{4.9}$$

where $I_0(t) = \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{t\cos\theta} d\theta$ is a modified Bessel function and $\nu^2 = \delta_x^2 + \delta_y^2$.

The distribution of the first order statistic of Z, or the smallest Z, can be derived from (4.7) with $k = 1$. A plot of $Z_{(1)}$ with different values for N is shown in Fig. 4.2.

The variance of the error motion vectors with N motion vectors sent back from the decoder is

$$\sigma^2_{\Delta_N} = \int_0^\infty z^2 f_{Z_{(1)}}(z) dz = \int_0^\infty z^2 N[1 - F_Z(z)]^{N-1} f_Z(z) dz \tag{4.10}$$

where $f_Z(z)$ is derived in (4.9) and $F_Z(z)$ is the cumulative probability distribution function of $f_Z(z)$. From (4.5), the rate difference in using N motion vectors compared to using only one motion vector

$$\Delta R_{1,N} = \frac{1}{8\pi^2} \int_{-\pi}^{\pi}\int_{-\pi}^{\pi} \log_2 \frac{1 - e^{-\frac{1}{2}\omega^T \omega \sigma^2_{\Delta_N}}}{1 - e^{-\frac{1}{2}\omega^T \omega \sigma^2_{\Delta_1}}} d\omega \tag{4.11}$$

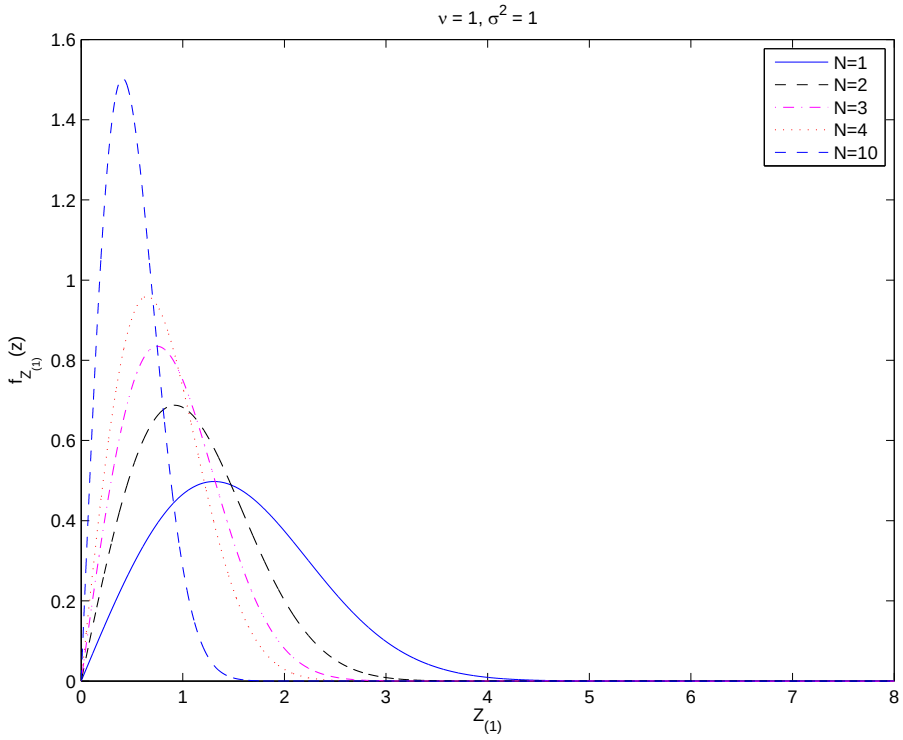


Fig. 4.2. The Probability Density Function of $Z_{(1)}$

The numerical rate difference with $\nu = 1$ and $\nu = \frac{1}{2}$ are shown in Fig. 4.3 and 4.4 respectively for various σ^2. Fig. 4.3 shows that sending more motion vectors through the backward channel can improve the coding efficiency of BP frames. For example, in the case of $\nu = 1, \sigma^2 = 1$, compared to only sending one motion vector to the encoder, sending five motion vectors can lead to a rate saving of 0.2276 bit/sample, or an improvement of roughly $0.2276 \times 6.02 = 1.3702$ dB in peak signal-to-noise ratio (PSNR). It is noted that this rate saving is more significant when σ^2 is smaller. When σ^2 is large (such as $\sigma^2 = 10$), sending more motion vectors to the encoder has very limited impact on rate savings. In other words, when there is fast or large irregular motion in a video sequence, sending more motion vectors is not justified. We also note that when ν is smaller, i.e, when the motion vectors extracted at the decoder from

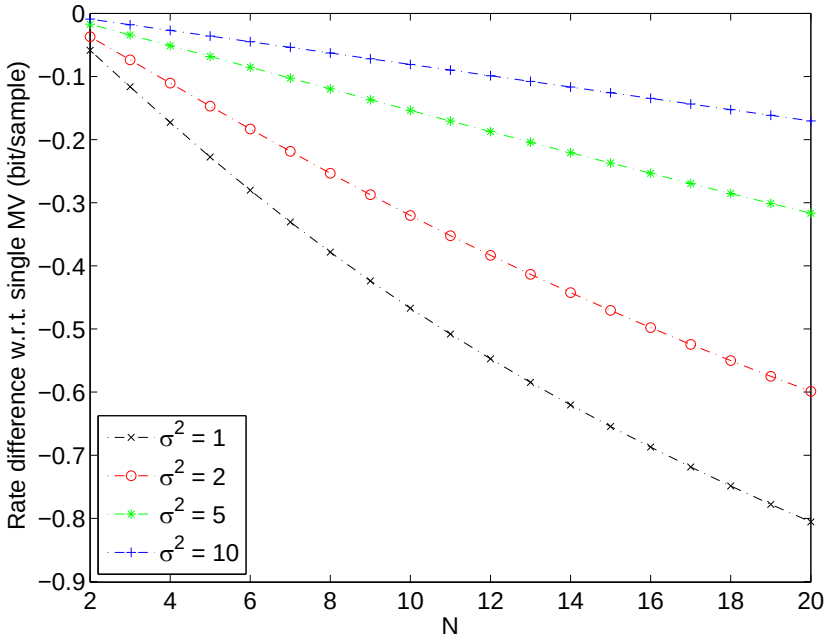


Fig. 4.3. Rate Difference of the Minimum Motion Estimator ($\delta_x = \delta_y = \frac{\sqrt{2}}{2},\ \nu = 1$)

the previous reconstructed frames are closer to the true motion vector on average, the rate saving is more significant.

Case II: $\delta_x = 0$ **and** $\delta_y = 0$

In this case, the mean of the received motion vector is identical to the true motion vector. This is a special case of (4.9) with $\nu = 0$. And the distribution of Z as defined in (4.6) is a Rayleigh distribution with

$$f_Z(z) = \frac{z}{\sigma^2} e^{-\frac{z^2}{2\sigma^2}}, \quad z \geq 0 \tag{4.12}$$

$$F_Z(z) = 1 - e^{-\frac{z^2}{2\sigma^2}}, \quad z \geq 0 \tag{4.13}$$

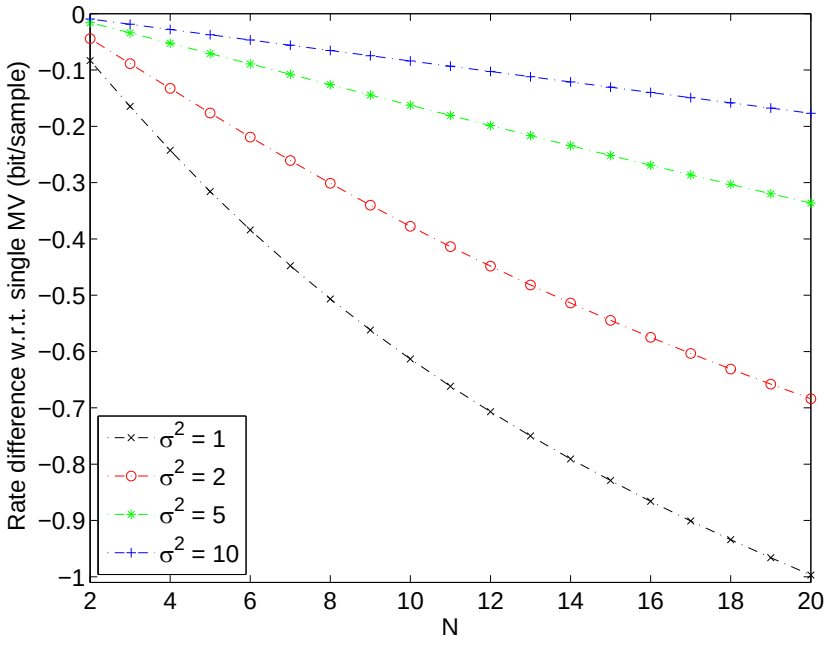


Fig. 4.4. Rate Difference of the Minimum Motion Estimator ($\delta_x = \delta_y = \frac{\sqrt{2}}{4}$, $\nu = \frac{1}{2}$)

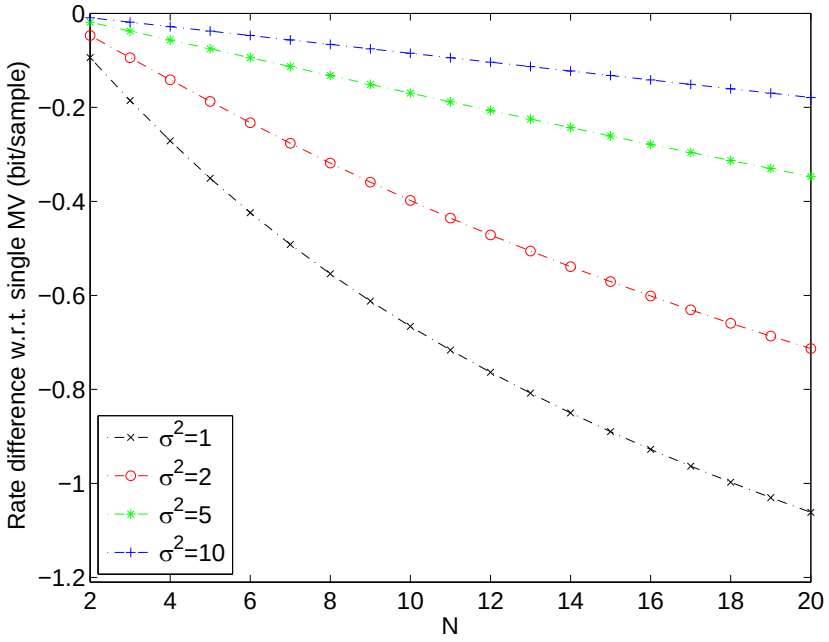


Fig. 4.5. Rate Difference of the Minimum Motion Estimator ($\delta_x = \delta_y = 0,\ \nu = 0$)

The first order statistic and the variance of the error motion vectors

$$\begin{aligned} f_{Z_{(1)}}(z) &= N[1 - F_Z(z)]^{N-1} f(z) = N\frac{z}{\sigma^2}(e^{-z^2/2\sigma^2})^N \\ \sigma^2_{\Delta_N} &= E[Z^2_{(1)}] = \int_0^\infty z^2 f_{Z_{(1)}}(z)dz \\ &= \int_0^\infty z^2 N\frac{z}{\sigma^2}(e^{-z^2/2\sigma^2})^N dz = \frac{2}{N}\sigma^2 \end{aligned} \tag{4.14}$$

The rate difference with respect to the single motion vector is

$$\Delta R_{1,N} = \frac{1}{8\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi} \log_2 \frac{1 - e^{-\frac{1}{2}\omega^T\omega(\frac{2}{N}\sigma^2)}}{1 - e^{-\frac{1}{2}\omega^T\omega(2\sigma^2)}} d\omega \tag{4.15}$$

The results are shown in Fig. 4.5. Compared to Fig. 4.3 and 4.4, the rate saving is higher than Case I when $\nu \neq 0$.

In terms of computational complexity, the motion search complexity of the minimum motion estimator is linear with the number N of the motion vectors sent back from the decoder. Since the encoder complexity in this case depends largely on the the motion search complexity, (4.11) and (4.15) describe not only the rate-distortion

tradeoff but also the complexity-rate-distortion tradeoff for the minimum estimator. Furthermore, we note that the backward channel bandwidth usage can also assume to be linear with the number N, hence (4.11) and (4.15) also describe the tradeoff between the rate distortion performance and the backward channel bandwidth usage.

4.4.3 The Median Motion Estimator

The minimum motion estimator in Section 4.4.2 requires that N motion vectors should be sent through the backward channel and the encoder then needs to make a motion search to find the best candidate motion vector. This leads to higher requirement on encoding complexity and backward channel bandwidth. When the application cannot satisfy this requirement, a motion vector needs to be selected at the decoder among the N motion vector candidates and only this motion vector will be sent to the encoder.

One way to choose such a motion vector is to use the median motion vector among the N motion vectors. The median motion vector is constructed using the median of the horizontal motion vectors and the median of the vertical motion vectors. When N is odd, i.e., $N = 2p + 1$, we choose $m = (p + 1)$th order statistic. When N is even ($N = 2p$), pth order statistic is chosen. Consider the Gaussian case in (4.8), the horizontal and vertical motion vectors are independent with

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-x'_n)^2}{2\sigma^2}} \tag{4.16}$$

$$f_Y(y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y-y'_n)^2}{2\sigma^2}} \tag{4.17}$$

Since X and Y follow the same distributions, we will discuss X only and the performance of Y can be analyzed similarly.

The cumulative probability distribution function of the Gaussian distribution in (4.16) can be expressed as

$$F_X(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}\sigma} e^{-(t-x'_n)^2/2\sigma^2} dt = 1 - \frac{1}{2}\text{erfc}(\frac{x - x'_n}{\sqrt{2}\sigma}) \tag{4.18}$$

Table 4.1

The variance of the error motion vectors for the minimum and median motion estimators ($\nu = 0, \sigma^2 = 1$)

N	1	3	5	7	9
minimum	2.000	0.667	0.400	0.286	0.222
median	2.000	0.897	0.574	0.421	0.332

where the error function erfc(x) is defined as erfc$(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt$. The probability density function of the median order statistic is

$$\begin{aligned} f_{X_{(m)}}(x) &= \frac{N!}{(m-1)!(N-m)!} F(x)^{m-1} [1 - F(x)]^{N-m} f(x) \\ &= \frac{N!}{(m-1)!(N-m)!} [1 - \frac{1}{2}\text{erfc}(\frac{x - x_n'}{\sqrt{2}\sigma})]^{m-1} \\ &\times [\frac{1}{2}\text{erfc}(\frac{x - x_n'}{\sqrt{2}\sigma})]^{N-m} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-x_n')^2}{2\sigma^2}} \end{aligned} \tag{4.19}$$

where the error function erfc(x) is defined as erfc$(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt$. Since we are interested in the distance to the true motion vector x_n, we define a new variable $X_d = X - x_n$; the distribution of X_d and the variance of the error motion vector $E[X_d^2]$, according to (4.7), are

$$\begin{aligned} f_{X_d}(x) &= \frac{N!}{(m-1)!(N-m)!} [1 - \frac{1}{2}\text{erfc}(\frac{x - \delta_x}{\sqrt{2}\sigma})]^{m-1} \\ &\times [\frac{1}{2}\text{erfc}(\frac{x - \delta_x}{\sqrt{2}\sigma})]^{N-m} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\delta_x)^2}{2\sigma^2}} \end{aligned} \tag{4.20}$$

$$E[X_d^2] = \int_{-\infty}^{\infty} x^2 f_{X_d}(x) dx \tag{4.21}$$

Since X and Y are independent, the total variance of the error motion vector $\sigma_\Delta^2 = E[X_d^2] + E[Y_d^2]$. Table 4.1 gives a summary of $\sigma_{\Delta_N}^2$ for the case $\nu = 0, \sigma^2 = 1$. The median motion estimator yields larger variance of the error motion vectors than the minimum estimator.

The rate difference with respect to the single motion vector is shown in Fig. 4.6-4.8. Comparing the results of the minimum motion estimator and the median motion

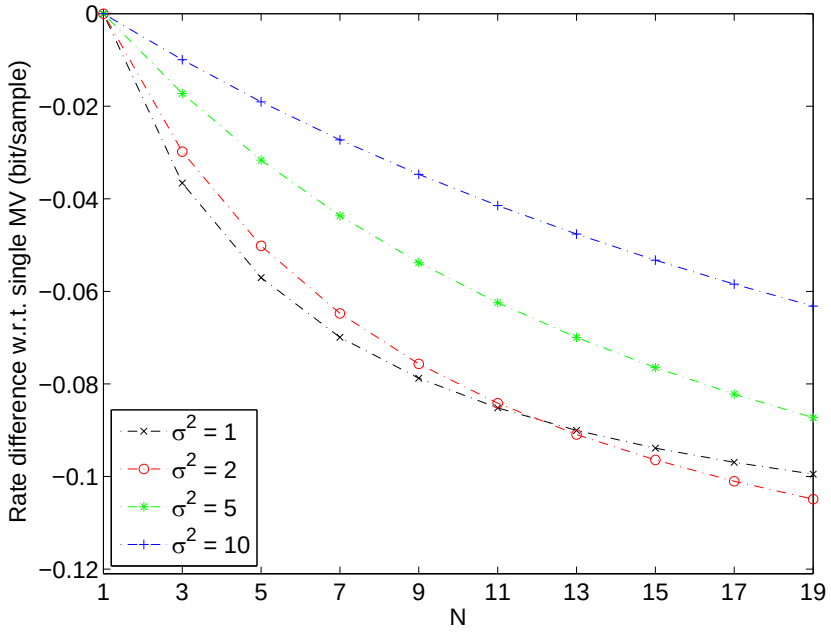


Fig. 4.6. Rate Difference of the Median Motion Estimator ($\delta_x = \delta_y = \frac{\sqrt{2}}{2},\ \nu = 1$)

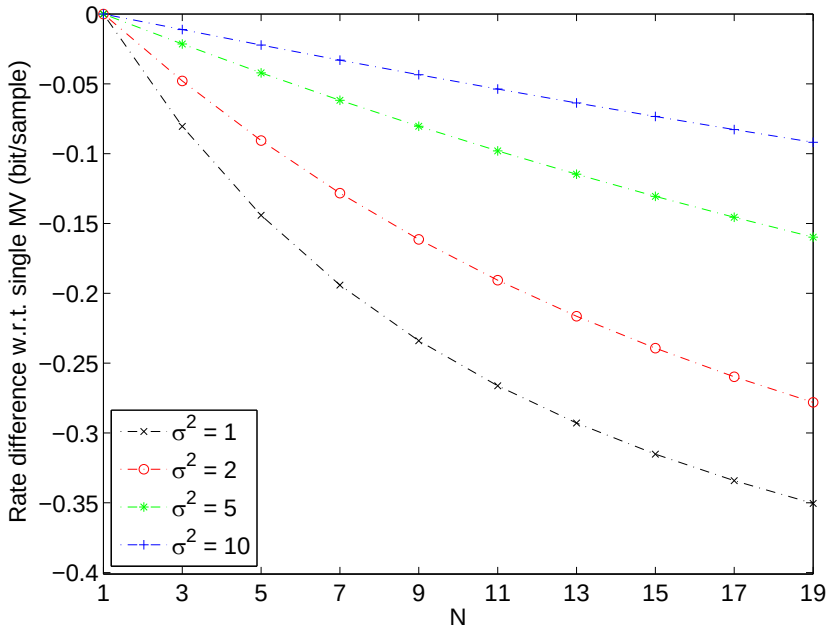


Fig. 4.7. Rate Difference of the Median Motion Estimator ($\delta_x = \delta_y = \frac{\sqrt{2}}{4},\ \nu = \frac{1}{2}$)

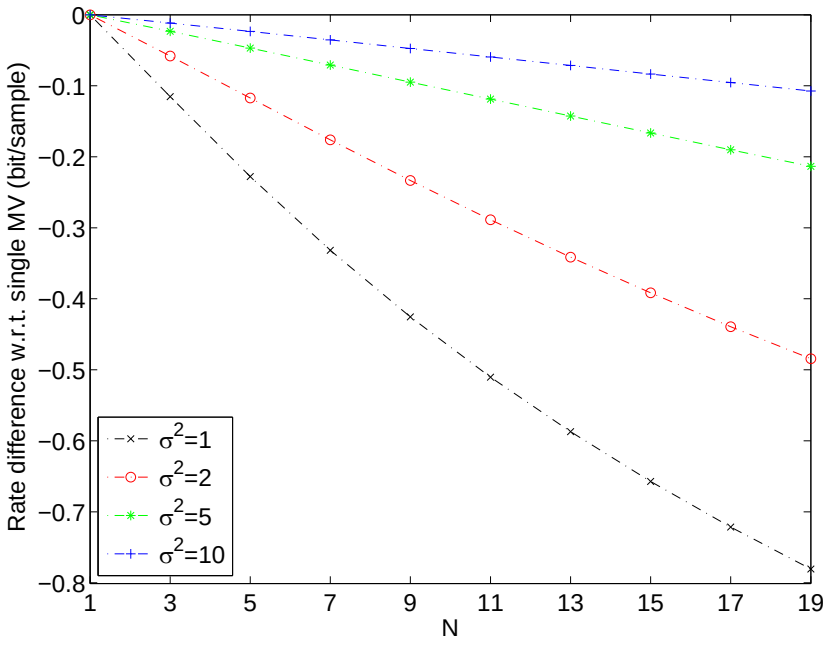


Fig. 4.8. Rate Difference of the Median Motion Estimator ($\delta_x = \delta_y = 0,\ \nu = 0$)

estimator, the rate saving of the median motion estimator is smaller than that of the minimum motion estimator. For example, when $\nu = 1, \sigma^2 = 1$ and $N = 5$, using the minimum estimator can achieve a rate saving of 0.2276 bit/sample, while using the median estimator can only achieve a rate saving of 0.0570 bit/sample. It is interesting to note that in Fig. 4.6, when N is large ($N \geq 12$), the rate saving when $\sigma^2 = 2$ is actually greater than the $\sigma^2 = 1$ case. This is because when the motion vector error is closer to the motion vector variance, while the rate saving will continue to improve with the increase of N, such improvement will grow at a slower pace. Fig. 4.8 shows the rate difference for the case $\delta_x = \delta_y = 0$. In this case, the rate saving is generally more significant than the case $\nu > 0$. And the increase in N can also noticeably improve the coding efficiency. In this case, the use of multiple motion vectors in the median motion estimator provides a more significant rate saving.

4.4.4 The Average Motion Estimator

Another way to choose the motion vector candidates at the decoder is to send the average of the motion vectors, which is referred to as the average motion estimator. With the N motion vectors available at the decoder, we send the average motion vector $\bar{X} = \frac{1}{N}(X_1 + X_2 + \cdots + X_N)$ and $\bar{Y} = \frac{1}{N}(Y_1 + Y_2 + \cdots + Y_N)$ to the encoder. The sample average of a Gaussian distribution is also Gaussian [134], so

$$\bar{X} \sim N(x'_n, \frac{\sigma^2}{N}), \quad \bar{Y} \sim N(y'_n, \frac{\sigma^2}{N}) \tag{4.22}$$

The variance of the error motion vector is

$$\begin{aligned} \sigma^2_{\Delta_N} &= E[(\bar{X} - x_n)^2] + E[(\bar{Y} - y_n)^2] \\ &= (\delta_x^2 + \frac{\sigma^2}{N}) + (\delta_y^2 + \frac{\sigma^2}{N}) = \nu^2 + 2\frac{\sigma^2}{N} \end{aligned} \tag{4.23}$$

The result of the variance is shown in Table 4.2 with comparisons to the minimum and median motion estimators ($\nu = 1, \sigma^2 = 1$). The variance of the error motion vectors using the average motion estimator is higher than the minimum motion estimator. Compared with the median motion vector, the variance using the average motion

Table 4.2

Comparisons of the variance of the error motion vectors for the minimum, median and average motion estimators

	$\nu = 1, \sigma^2 = 1$			$\nu = \frac{1}{2}, \sigma^2 = \frac{1}{2}$		
N	Minimum	Median	Average	Minimum	Median	Average
1	3.0000	3.0000	3.0000	1.2500	1.2500	1.2500
3	1.0519	1.8973	1.6667	0.4233	0.6987	0.5833
5	0.6406	1.5737	1.4000	0.2550	0.5368	0.4500
7	0.4609	1.4209	1.2857	0.1825	0.4604	0.3929
9	0.3600	1.3322	1.2222	0.1421	0.4161	0.3611

estimator is lower. The rate difference with respect to the single motion vector using the average motion estimator is shown in Fig. 4.9-4.11. With the same requirement of the encoder complexity and the bandwidth of the backward channel, the average motion estimator gives better rate distortion performance than the median motion estimator. Similar to Fig. 4.6 and for the same reason mentioned in Section 4.4.3, we can observe a similar *crossover* effect in Fig. 4.9 between the curves of $\sigma^2 = 1$ and $\sigma^2 = 2$ when $N \geq 8$. In the case $\nu = 0$, (4.23) reduces to $\sigma^2_{\Delta_N} = 2\frac{\sigma^2}{N}$, which is identical to (4.14). In other words, when $\nu = 0$, the average motion estimator performs as well as the minimum motion estimator. While when $\nu \neq 0$, the minimum motion estimator can achieve higher coding efficiency at the cost of higher encoding complexity and more backward channel bandwidth.

4.4.5 Comparisons of Minimum, Median and Average Motion Estimators

Fig. 4.12 shows two examples of the comparisons of three motion estimators. The green curve denotes the minimum motion estimator, the red curve denotes median motion estimator, and the black curve denotes the average motion estimator. In Fig. 4.12-(b), ν is small and σ^2 is large, there are not significant differences between the

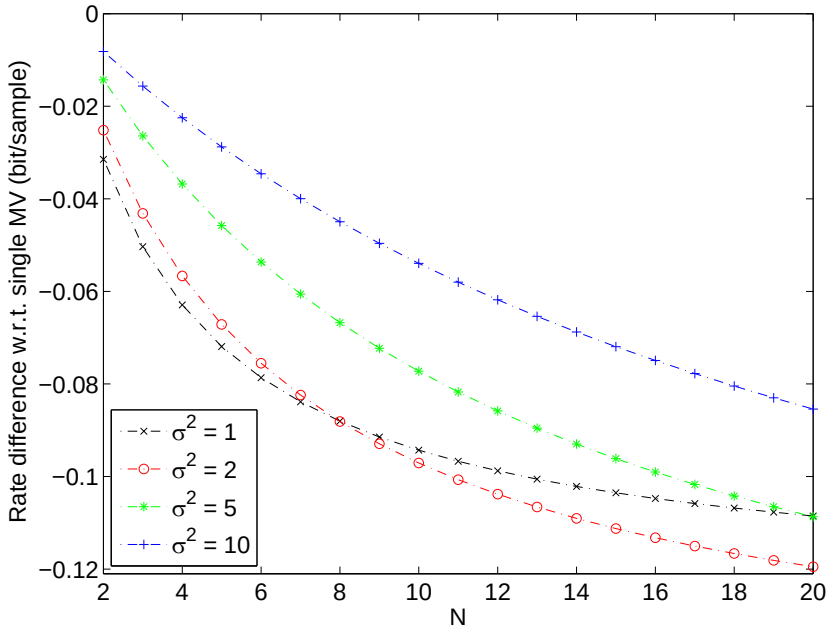


Fig. 4.9. Rate Difference of the Average Motion Estimator ($\delta_x = \delta_y = \frac{\sqrt{2}}{2}$, $\nu = 1$)

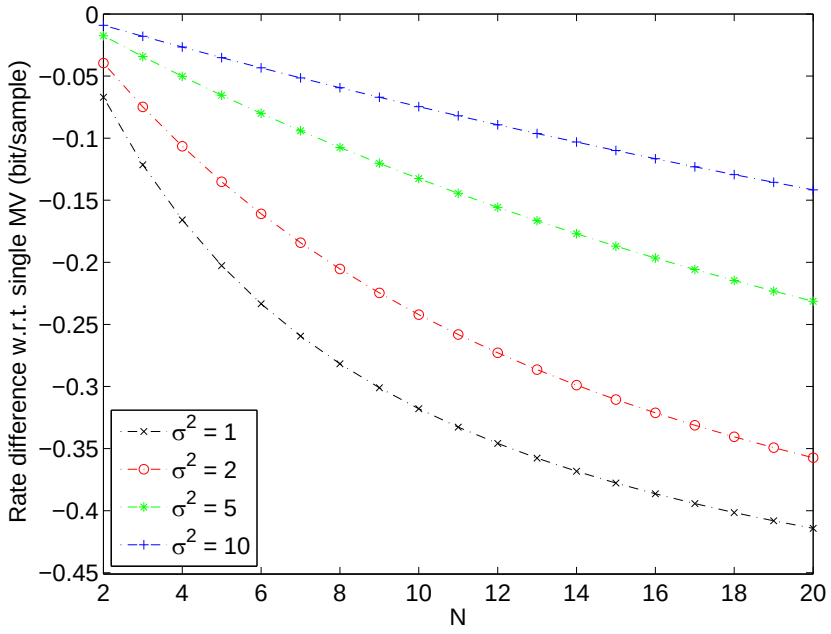


Fig. 4.10. Rate Difference of the Average Motion Estimator ($\delta_x = \delta_y = \frac{\sqrt{2}}{4}$, $\nu = \frac{1}{2}$)

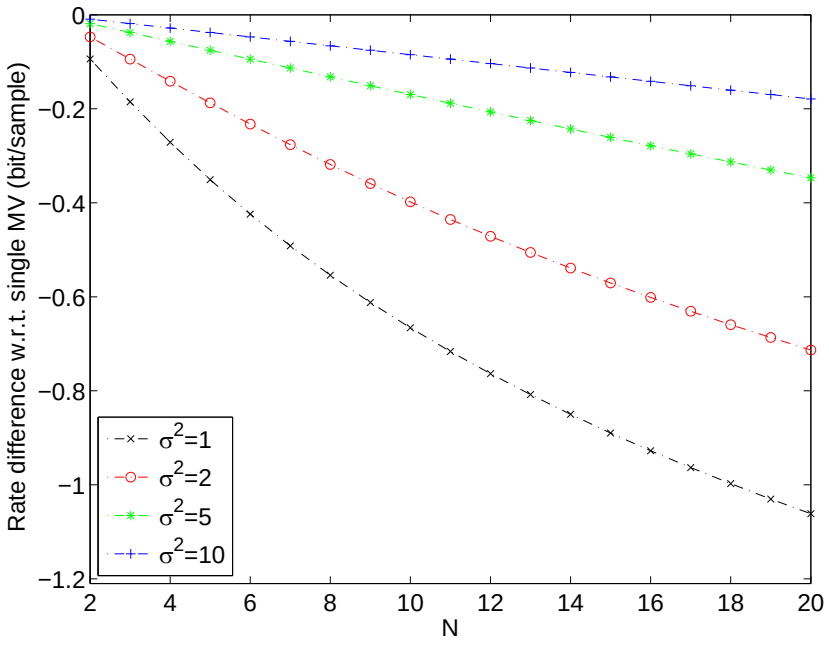
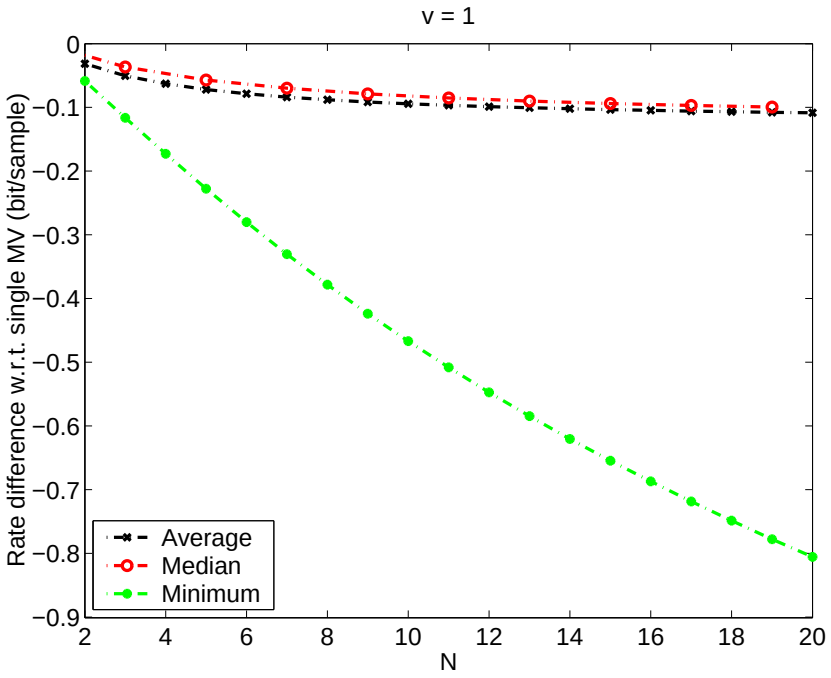
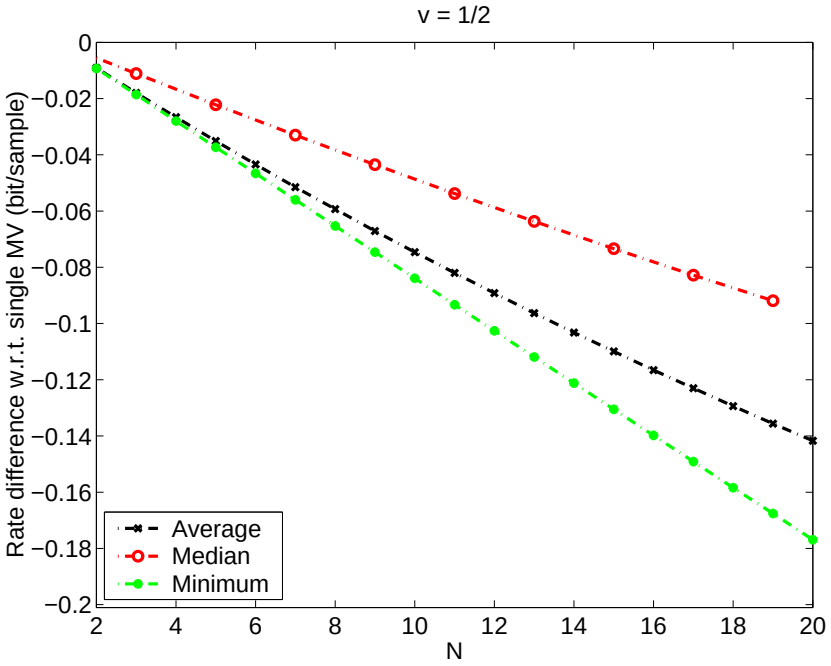


Fig. 4.11. Rate Difference of the Average Motion Estimator ($\delta_x = \delta_y = 0,\ \nu = 0$)

three estimators. In Fig. 4.12-(a), the minimum motion estimator performs much better than the other two motion estimators. The rate-distortion performance of the minimum motion estimator is best among the three motion estimators. And typically the average motion estimator performs better than the median estimator. However, the minimum motion estimator has higher requirements on the encoder complexity and the backward channel bandwidth.



(a) $\delta_x = \delta_y = \frac{\sqrt{2}}{2}\,\sigma^2 = 1$



(b) $\delta_x = \delta_y = \frac{\sqrt{2}}{4}\,\sigma^2 = 10$

Fig. 4.12. Comparisons of the Minimum, Median and Average Motion Estimators

5. CONCLUSIONS

The advance of video compression and wireless communication systems has made it possible to capture and transmit video signals almost any time and anywhere. A new approach to video coding, known as Wyner-Ziv video coding, was proposed in recent years. The main characteristic of Wyner-Ziv coding is that side information is available only at the decoder. Sensor networks and mobile video are some of the applications where Wyner-Ziv coding may be very useful. These applications are characterized the need for reduced encoding complexity due to the limited availability of resources (e.g. battery life and memory). In this thesis, we studied the new approaches for low complexity video coding. Complexity-rate-distortion tradeoff of the new approach is extensively analyzed.

5.1 Contributions of The Thesis

The main contributions of the thesis are:

- **Wyner-Ziv Video Codec Architecture Testbed**

  We studied the Wyner-Ziv video codec architecture and built a Wyner-Ziv video coding testbed. The video sequences are divided into two different parts using different coding schemes. Part of the sequence is coded by conventional INTRA coding scheme and referred as key frames. The other part of the sequence is coded as Wyner-Ziv frames using channel coding methods. Both turbo codes and low-density-parity-check (LDPC) codes are supported in the system. The sequences can be coded either in pixel domain and transform domain. Only parts of the parity bits from the channel coders are transmitted to the decoder. Hence, the decoding of the Wyner-Ziv frames need side information at the decoder. The side information can be extracted from the key frames by

extrapolation or interpolation. We study various methods for side information generation. We also analyze the rate-distortion performance of Wyner-Ziv video coding compared with conventional INTRA or INTER coding.

- **Backward Channel Aware Wyner-Ziv Video Coding**

  In Wyner-Ziv video coding, many key frames have to be INTRA coded to keep the complexity low at the encoder and provide side information for the Wyner-Ziv frames. However, the usage of INTRA frames limits the coding performance. We propose a new Wyner-Ziv video coding method that uses backward channel aware motion estimation to code the key frames. The main idea is to do motion estimation at the decoder and send the estimated motion vector back to the encoder. In this way, we can keep the complexity low at the encoder with minimum usage of the backward channel. Our experimental results show that the scheme can significantly improve the coding efficiency by 1-2 dB compared with Wyner-Ziv with INTRA-coded key frames. We also propose to use multiple modes of motion decision, which can further improve the coding efficiency by 0.5-2 dB. When the backward channel is subject to erasure errors or delays, the coding efficiency of our method decreases. We provide an error resilience technique to handel the situation. The rate distortion performance of the error resilience technique can avoid the quality degradation due to channel errors and only incurs a small penalty compared with error free channel.

- **Complexity-Rate-Distortion Analysis of Backward Channel Aware Wyner-Ziv Video Coding**

  In the previous section, we study the rate-distortion performance of backward channel aware Wyner-Ziv video coding. We further present a model to study the complexity-rate-distortion coding efficiency of backward channel aware Wyner-Ziv video coding. We present three motion estimator: minimum motion estimator, median motion estimator and average motion estimator. Suppose we have several motion vectors derived at the decoder. The minimum motion estima-

tor sends all the motion vectors derived at the decoder to the encoder and the encoder makes a decision to choose the best motion vector with smallest distortion. When the backward channel bandwidth cannot meet the requirement or the encoder complexity becomes a concern, the median or average motion estimator chooses one motion vector at the decoder. The results show the rate-distortion performance of the average estimator is generally higher than that of the median estimator. If the rate-distortion tradeoff is the only concern, the minimum estimator yields better results than the other two estimators. However, for applications with complexity constraints, our analysis shows that the average estimator could be a better choice. Our proposed model quantitatively describes the complexity-rate-distortion tradeoff among these estimators.

5.2 Future Work

In this thesis, we proposed to use backward channel aware motion estimation in Wyner-Ziv video coding. In the future, we could study various ways to improve the accuracy of motion vectors. For example, we could refine the received motion vector by searching near its neighbors. When making the mode decision at the encoder, we could consider the rate factor to make a rate-distortion optimized decision. We could also insert intra blocks to BP frames where the correlation between the neighboring frames are weak.

We also presented a model used to study the complexity-rate-distortion coding efficiency of backward channel aware Wyner-Ziv video coding. For future work, we will investigate the use of our model to examine the impact of INTRA-coded frames and INTER-coded frames. The comparisons of INTRA frame, BP frame and INTER-frame can be done for their CRD performance. Based on the CRD model, we could optimize the coding structure to minimize the Lagrange cost of the rate and distortion with a complexity constraint.

The backward channel aware Wyner-Ziv video coding would be suitable for video surveillance applications, where a low complexity encoder is needed and feedback channel is available between the cameras and the base station. We plan to study the requirements of a practical video surveillance system and build and refine the backward channel aware Wyner-Ziv video coding structure to fulfill the task.

LIST OF REFERENCES

LIST OF REFERENCES

[1] G. J. Sullivan and T. Wiegend, "Video compression: From concepts to the H.264/AVC standard," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 18– 31, January 2005.

[2] V. Bhaskaran and K. Konstantinides, *Image and Video Compression Standards: Algorithms and Architecture*, 2nd ed. Boston, MA: Kluwer Academic Publishers, 1997.

[3] A. Bovik, Ed., *Handbook of Image and video Processing.* San Diego, CA: Academic Press, 2000.

[4] D. T. Lee, "JPEG2000: Retrospective and new developments," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 32–41, January 2005.

[5] T. Sikora, "Trends and perspectives in image and video coding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 6– 17, January 2005.

[6] G. K. Wallace, "The JPEG still picture compression standard," *Communications of the ACM*, vol. 34, no. 4, pp. 30–44, April 1991.

[7] "Information technology−JPEG 2000 image coding system−Part 1: Core coding system," Int. Standards Org./Int. Electrotech. Comm. (ISO/IEC), ISO/IEC 15444-1:2000 and ITU-T T.800.

[8] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The JPEG2000 still image coding system: An overview," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 4, pp. 1103–1127, November 2000.

[9] A. M. Tekalp, *Digital Video Processing.* Prentice Hall, 1995.

[10] B. Girod, "Lecture Notes of EE398 - Image Communication II, Stanford University," http://www.stanford.edu/class/ee398/, Spring 2003-2004.

[11] *Video Coding for Low Bit Rate Communications*, ITU-T Recommendation H.263 Version 2, January 1998.

[12] G. Cote, B. Erol, M. Gallant, and F. Kossentini, "H.263+: video coding at low bit rates," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 7, pp. 849–866, November 1998.

[13] *Coding of Audiovisual Objects - Part 2: Visual*, ISO/IEC ISO/IEC 14496-2 (MPEG-4), 1999.

[14] R. Talluri, "Error-resilient video coding in the ISO MPEG-4 standard," *IEEE Communications Magazine*, vol. 36, no. 6, pp. 112–119, June 1998.

[15] Y. Wang, J. Ostermann, and Y.-Q. Zhang, *Video Processing and Communications*. Upper Saddle River, NJ: Prentice Hall, 2002.

[16] 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*, vol. 13, no. 7, pp. 560– 576, July 2003.

[17] G. J. Sullivan, P. Topiwala, and A. Luthra, "The H.264/AVC advanced video coding standard: Overview and introduction to the fidelity range extensions," in *Proceedings of the SPIE: Applications of Digital Image Processing XXVII*, San Jose, CA, January 18-22 2004, pp. 454–474.

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

[19] G. J. Sullivan, J.-R. Ohm, A. Ortega, E. J. Delp, A. Vetro, and M. Barni, "dsp forum - future of video coding and transmission," *IEEE Signal Processing Magazine*, vol. 23, no. 6, pp. 76–82, November 2006.

[20] M. Bosch, F. Zhu, and E. J. Delp, "Spatial texture models for video compression," in *Proceedings of the IEEE International Conference on Image Processing*, San Antonio, TX, September 2007.

[21] L. Liu, Y. Liu, and E. J. Delp, "Content-adaptive motion estimation for efficient video compression," in *Proceedings of the SPIE International Conference on Visual Communications and Image Processing*, San Jose, California, January 28 - February 1 2007.

[22] ——, "Enhanced intra prediction using context-adaptive linear prediction," accepted by the Picture Coding Symposium, Lisbon, Portugal, November 2007.

[23] G. Abdollahian and E. J. Delp, "Finding regions of interest in home videos based on camera motion," in *Proceedings of the IEEE International Conference on Image Processing*, San Antonio, TX, September 2007.

[24] Y. Chen, C. Lettsome, M. J. Smith, and E. J. Delp, "A low bit-rate video coding approach using modified adaptive warping and long-term spatial memory," in *Proceedings of SPIE International Conference on Visual Communications and Image Processing*, San Jose, CA, January 28 - Feburary 1 2007.

[25] A. Aaron, S. Rane, , E. Setton, and B. Girod, "Transform-domain Wyner-Ziv codec for video," in *Proceedings of the SPIE: Visual Communications and Image Processing*, San Jose, CA, Januaray 2004.

[26] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471–480, July 1973.

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

[28] A. Aaron and B. Girod, "Compression with side information using turbo codes," in *Proceedings of the IEEE Data Compression Conference (DCC)*, Snowbird, UT, April 2-4 2002, pp. 252–261.

[29] A. Aaron, E. Setton, and B. Girod, "Towards practical Wyner-Ziv coding of video," in *Proceeding of IEEE International Conference on Image Processing, ICIP-2003*, Barcelona, Spain, September 14-17 2003, pp. 869–872.

[30] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proceedings of the IEEE: Special Issue on Advances in Video Coding and Delivery*, vol. 93, no. 1, pp. 71– 83, January 2005.

[31] J. Garcia-Frias and Y. Zhao, "Compression of correlated binary sources using turbo codes," *IEEE Communications Letters*, vol. 5, no. 10, pp. 417–419, October 2001.

[32] 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*, vol. 6, no. 10, pp. 440 – 442, October 2002.

[33] A. Liveris, Z. Xiong, and C. Georghiades, "A distributed source coding technique for correlated images using turbo-codes," *IEEE Communications Letters*, vol. 6, no. 9, pp. 379–381, September 2002.

[34] R. Puri and K. Ramchandran, "Prism: An uplink-friendly multimedia coding paradigm," in *Proceedings of the IEEE International Conference on Image Processing*, vol. 1, Barcelona, Spain, September 14-17 2003.

[35] Z. Li, "New methods for motion estimation with applications to low complexity video compression," Ph.D. Thesis, School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, December 2005.

[36] Y. Liu, "Layered scalable and low complexity video encoding: New approaches and theoretic analysis," Ph.D. Thesis, School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, August 2004.

[37] L. Liu, "New approaches for low complexity video coding and reliable transmission," preliminary report, School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, December 2005.

[38] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo-codes," in *International Conference on Communications*, Geneva, Switzerland, May 23-26 1993, pp. 1064–1070.

[39] C. Berrou and A. Glavieux, "Near optimium error correcting coding and decoding: Turbo codes," *IEEE Transactions on Communications*, vol. 44, no. 10, pp. 1261 – 1271, October 1996.

[40] W. E. Ryan, "Concatenated convolutional codes and iterative decoding," in Wiley Encyclopedia of Telecommunications, J. G. Proakis ed., John Wiley and Sons, 2003. [Online]. Available: http://www.ece.arizona.edu/ ryan/

[41] C. Schlegel and L. Perez, *Trellis and Turbo Coding*. New York, NY: Wiley-IEEE Press, 2003.

[42] R. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1963.

[43] D. MacKay and R. M. Neal, "Near shannon limit performance of low density parity check codes," *IEE Electronics Letters*, vol. 32, no. 18, pp. 1645 – 1646, August 1996.

[44] D. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 399–431, March 1999.

[45] W. E. Ryan, "An introduction to LDPC codes," in CRC Handbook for Coding and Signal Processing for Recording Systems, B. Vasic ed., CRC Press, 2004. [Online]. Available: http://www.ece.arizona.edu/ ryan/

[46] J. Ascenso, C. Brites, and F. Pereira, "Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding," in *5th EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services*, Slovak Republic, July 2005.

[47] C. Brites, J. Ascenso, and F. Pereira, "Improving transform domain wyner-ziv video coding performance," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Toulouse, France, May 2006.

[48] X. Artigas and L. Torres, "An approach to distributed video coding using 3D face models," in *Proceedings of the IEEE International Conference on Image Processing*, Atlanta, Georgia, October 8-12 2006.

[49] L. Liu and E. J. Delp, "Wyner-Ziv video coding using LDPC codes," in *Proceedings of the IEEE 7th Nordic Signal Processing Symposium (NORSIG 2006)*, Reykjavik, Iceland, June 7-9 2005.

[50] X. Artigas and L. Torres, "Iterative generation of motion-compensated side information for distributed video coding," in *Proceedings of the IEEE International Conference on Image Processing*, vol. 1, Genova, Italy, September 11-14 2005, pp. 833 – 836.

[51] J. Ascenso, C. Brites, and F. Pereira, "Motion compensated refinement for low complexity pixel based distributed video coding," in *IEEE International Conference on Advanced Video and Signal Based Surveillance*, Como, Italy, September 2005.

[52] DISCOVER, "Distributed coding for video services - application scenarios and functionalities for DVC," http://www.discoverdvc.org/deliverables/Discover-D4.pdf.

[53] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): design and construction," *IEEE Transactions on Information Theory*, vol. 49, no. 3, pp. 626 – 643, March 2003.

[54] R. Puri and K. Ramchandran, "PRISM: a video coding paradigm based on motion-compensated prediction at the decoder," submitted. [Online]. Available: http://www-wavelet.eecs.berkeley.edu/ rpuri/researchlinks/papers/purirvc2003.pdf.gz

[55] P. Ishwar, V. Prabhakaran, and K. Ramchandran, "Towards a theory for video coding using distributed compression principles," in *Proceedings of the IEEE International Conference on Image Processing*, vol. 2, Barcelona, Spain, September 2003.

[56] A. Sehgal, A. Jagmohan, and N. Ahuja, "Wyner-Ziv coding of video: an error-resilient compression framework," *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 249 – 258, April 2004.

[57] A. Jagmohan, A. Sehgal, and N. Ahuja, "Predictive encoding using coset codes," in *Proceeding of IEEE International Conference on Image Processing, ICIP-2002*, Rochester, New York, September 22-25 2002, pp. 29–32.

[58] S. Shamai, S. Verdu, and R. Zamir, "Systematic lossy source/channel coding," *IEEE Transactions on Information Theory*, vol. 44, pp. 564–579, March 1998.

[59] 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 IEEE International Conference on Image Processing, ICIP-2004*, Singapore, October 24-27 2004, pp. 3101–3104.

[60] ——, "Systematic lossy forward error protection for error-resilient digital video broadcasting," in *Proceedings of the SPIE: Visual Communications and Image Processing*, San Jose, CA, Januaray 2004.

[61] L. Liang, P. Salama, and E. J. Delp, "Unequal error protection using Wyner-Ziv coding," in *Proceedings of SPIE International Conference on Visual Communications and Image Processing*, San Jose, CA, January 28 - Feburary 1 2007.

[62] Q. Xu and Z. Xiong, "Layered Wyner-Ziv video coding," in *Proceedings of the SPIE: International Conference on Image and Video Communications and Processing(IVCP)*, San Jose, CA, January 18-22 2004.

[63] Q. Xu, "Layered wyner-ziv video coding for noisy channel," Master Thesis, Department of Electrical Engineering, Texas A&M University, College Station, TX, August 2004.

[64] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable H.264/MPEG4-AVC extension," in *Proceedings of the IEEE International Conference on Image Processing*, Atlanta, Georgia, October 8-12 2006.

[65] J. Prades-Nebot, G. W. Cook, and E. J. Delp, "An analysis of the efficiency of different SNR-scalable strategies for video coders," *IEEE Transactions on Image Processing*, vol. 15, no. 4, pp. 848– 864, April 2006.

[66] G. Cook, J. Prades-Nebot, Y. Liu, and E. Delp, "Rate-distortion analysis of motion-compensated rate scalable video," *IEEE Transactions on Image Processing*, vol. 15, no. 8, pp. 2170 – 2190, August 2006.

[67] Y. Liu, P. Salama, Z. Li, and E. Delp, "An enhancement of leaky prediction layered video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 11, pp. 1317– 1331, November 2005.

[68] G. Cook, "A study of scalability in video compression: Rate-distortion analysis and parallel implementation," Ph.D. dissertation, Purdue University, West Lafayette, IN, December 2002.

[69] K. Shen and E. J. Delp, "Wavelet based rate scalable video compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 1, pp. 109 – 122, February 1999.

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

[71] J. Reichel, H. Schwarz, and M. Wien, "Joint scalable video model JSVM-7," JVT-T202, Klagenfurt, Austria, July 2006.

[72] F. Wu, S. Li, and Y.-Q. Zhang, "A framework for efficient progressive fine granularity scalable video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 3, pp. 332–344, March 2001.

[73] A. R. Reibman, L. Bottou, and A. Basso, "Scalable video coding with managed drift," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 2, pp. 131– 140, February 2003.

[74] A. Smolic and P. Kauff, "Interactive 3-D video representation and coding technologies," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 98– 110, January 2005.

[75] "Introduction to multi-view video coding," ISO/IEC JTC 1/SC 29/WG 11, Doc N7328, Poznan, Poland, July 2005.

[76] K. Muler, P. Merkle, H. Schwarz, T. Hinz, A. Smolic, and T. Wiegand, "Multi-view video coding based on H.264/AVC using hierarchical B-frames," in *Proceedings of Picture Coding Symposium (PCS)*, Beijing, China, April 2006.

[77] E. Martinian, A. Behrens, J. Xin, A. Vetro, and H. Sun, "Extensions of H.264/AVC for multiview video compression," in *Proceedings of the IEEE International Conference on Image Processing*, Atlanta, Georgia, October 8-12 2006.

[78] A. Vetro, Y. Su, H. Kimata, and A. Smolic, "Joint multiview video model (JMVM) 2.0," JVT-U207, Hangzhou, China, October 2006.

[79] X. Guo, Y. Lu, F. Wu, and W. Gao, "Inter-view direct mode for multiview video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 12, pp. 1527–1532, December 2006.

[80] E. Martinian, A. Behrens, J. Xin, and A. Vetro, "View synthesis for multi-view video compression," in *Proceedings of Picture Coding Symposium (PCS)*, Beijing, China, April 2006.

[81] J. Garbas, U. Fecker, T. Trger, and A. Kaup, "4D scalable multi-view video coding using disparity compensated view filtering and motion compensated temporal filtering," in *Proceedings of the IEEE International Workshop on Multimedia Signal Processing (MMSP)*, Victoria, Canada, October 2006.

[82] X. Guo, Y. Lu, F. Wu, W. Gao, and S. Li, "Distributed multi-view video coding," in *Proceedings of the SPIE International Conference on Visual Communications and Image Processing*, San Jose, California, January 2006.

[83] L. Liu, Y. Liu, and E. J. Delp, "Network-driven Wyner-Ziv video coding using forward prediction," in *Proceedings of the SPIE International Conference on Image and Video Communications and Processing (IVCP)*, San Jose, California, January 2005, pp. 641–651.

[84] L. Liu, P. Sabria, L. Torres, and E. J. Delp, "Error resilience in network-driven Wyner-Ziv video coding," in *Proceedings of the SPIE International Conference on Visual Communications and Image Processing*, San Jose, California, January 2006.

[85] L. Liu, Z. Li, and E. J. Delp, "Backward channel aware Wyner-Ziv video coding," in *Proceedings of the IEEE International Conference on Image Processing*, Atlanta, Georgia, October 8-12 2006, pp. 1677–1680.

[86] ——, "Complexity-constrained rated-distortion optimization for Wyner-Ziv video coding," in *Proceedings of the SPIE International Conference on Multimedia on Mobile Devices*, San Jose, California, January 28 - February 1 2007.

[87] L. Liu, F. Zhu, M. Bosch, and E. J. Delp, "Recent advances in video compression:whats next?" in *Proceedings of the International Symposium on Signal Processing and its Applications*, Sharjah, United Arab Emirates (U.A.E.), February 2007.

[88] L. Liu, Z. Li, and E. J. Delp, "Complexity-rate-distortion analysis of backward channel aware wyner-ziv video coding," in *Proceedings of the IEEE International Conference on Image Processing*, San Antonio, TX, September 2007.

[89] L. Liu, F. Zhu, M. Bosch, and E. J. Delp, "Recent advances in video compression: what's next?" in *book chapter in Statistical Science and Interdisciplinary Research: Pattern Recognition, Image Processing, and Video Processing*, Eds. Bhabatosh Chanda, et al, World Scientific Press 2007.

[90] Z. Li, L. Liu, and E. J. Delp, "Rate distortion analysis of motion side estimation in Wyner-Ziv video coding," *IEEE Transactions on Image Processing*, vol. 16, pp. 98–113, January 2007.

[91] ——, "Wyner-Ziv video coding with universal prediction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 11, pp. 1430–1436, November 2006.

[92] ——, "Wyner-Ziv video coding with universal prediction," in *Proceedings of the SPIE International Conference on Visual Communications and Image Processing*, San Jose, CA, January 2006.

[93] ——, "Wyner-Ziv video coding: A motion estimation perspective," in *Proceedings of the SPIE International Conference on Visual Communications and Image Processing*, San Jose, CA, January 2006.

[94] S. Verdu, "Fifty years of shannon theory," *IEEE Transactions on Information Theory*, vol. IT-44, pp. 2057–2078, October 1998.

[95] 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*, vol. 49, no. 5, pp. 1181 – 1203, May 2003.

[96] M. Costa, "Writing on dirty paper," *IEEE Transactions on Information Theory*, vol. 29, no. 3, pp. 439– 441, May 1983.

[97] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York, NY: Wiley series in telecommunications, 1991.

[98] A. Trapanese, M. Tagliasacchi, S. Tubaro, J. Ascenso, C. Brites, and F. Pereira, "Improved correlation noise statistics modeling in frame-based pixel domain wyner-ziv video coding," in *International Workshop on Very Low Bitrate Video Coding*, Sardinia, Italy, September 2005.

[99] R. Westerlaken, R. K. Gunnewiek, and R. Lagendijk, "The role of the virtual channel in distributed source coding of video," in *Proceedings of the IEEE International Conference on Image Processing*, Genova, Italy, September 2005, pp. 581 – 584.

[100] M. Dalai, R. Leonardi, and F. Pereira, "Improving turbo codec integration in pixel-domain distributed video coding," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, Toulouse, France, May 14-19 2006.

[101] V. Sheinin, A. Jagmohan, and D. He, "Uniform threshold scalr quantizer performance in Wyner-Ziv coding with memoryless, additive laplacian correlation channel," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, Toulouse, France, May 14-19 2006.

[102] C. Brites, J. Ascenso, and F. Pereira, "Feedback channel in pixel domain Wyner-Ziv video coding: Myths and realities," in *Proceedings of the 14th European Signal Processing Conference*, Florence, Italy, September 2006.

[103] M. Tagliasacchi, A. Trapanese, S. Tubaro, J. Ascenso, C. Brites, and F. Pereira, "Exploiting spatial redundancy in pixel domain Wyner-Ziv video coding," in *Proceedings of the IEEE International Conference on Image Processing*, Atlanta, Georgia, October 8-12 2006.

[104] A. Aaron, S. Rane, and B. Girod, "Wyner-Ziv video coding with hash-based motion compensation at the receiver," in *Proceedings of the IEEE International Conference on Image Processing*, Singapore, October 2004.

[105] A. Trapanese, M. Tagliasacchi, S. Tubaro, J. Ascenso, C. Brites, and F. Pereira, "Embedding a block-based intra mode in frame-based pixel domain Wyner-Ziv video coding," in *International Workshop on Very Low Bitrate Video (VLBV 2005)*, Sardinia, Italy, September 15-16 2005.

[106] M. Tagliasacchi, A. Trapanese, S. Tubaro, J. Ascenso, C. Brites, and F. Pereira, "Intra mode decision based on spatio-temporal cues in pixel domain Wyner-Ziv video coding," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, Toulouse, France, May 14-19 2006.

[107] A. Aaron, D. Varodayan, and B. Girod, "Wyner-Ziv residual coding of video," in *Proceedings of Picture Coding Symposium (PCS)*, Beijing, China, April 2006.

[108] [Online]. Available: http://www.eccpage.com/

[109] D. MacKay. [Online]. Available: http://www.inference.phy.cam.ac.uk/mackay/

[110] Z. Li and E. J. Delp, "Wyner-Ziv video side estimator: conventional motion search methods revisited," in *Proceedings of the IEEE International Conference on Image Processing*, vol. 1, Genova, Italy, September 11-14 2005, pp. 825 – 828.

[111] T. Berger, *Rate distortion theory: A mathematical basis for data compression.* Prentice-Hall, 1971.

[112] N. Merhav and M. Feder, "Universal prediction," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2124–2147, October 1998.

[113] N. Merhav and J. Ziv, "On the Wyner-Ziv problem for individual sequences," *IEEE Transactions on Information Theory*, vol. 52, no. 3, pp. 867– 873, March 2006.

[114] W. B. Rabiner and A. P. Chandrakasan, "Network-driven motion estimation for wireless video terminals," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 4, pp. 644–653, August 1997.

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

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

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

[118] W.-Y. Kung, C.-S. Kim, and C.-C. Kuo, "Spatial and temporal error concealment techniques for video transmission over noisy channels," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 7, pp. 789– 803, July 2006.

[119] JM8.0, http://iphome.hhi.de/suehring/tml/.

[120] Z. He, Y. Liang, L. Chen, I. Ahmad, and D. Wu, "Power-rate-distortion analysis for wireless video communication under energy constraints," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 5, pp. 645– 658, May 2005.

[121] H. F. Ates, B. Kanberoglu, and Y. Altunbasak, "Rate-distortion and complexity joint optimization for fast motion estimatin in H.264 video coding," in *Proceedings of the IEEE International Conference on Image Processing*, Atlanta, Georgia, October 8-12 2006.

[122] D. S. Turaga, M. van der Schaar, and B. Pesquet-Popescu, "Complexity scalable motion compensated wavelet video encoding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 8, pp. 982– 993, August 2005.

[123] 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*, vol. 13, no. 7, pp. 704–716, Jul. 2003.

[124] M. van der Schaar and Y. Andreopoulos, "Rate-distortion-complexity modeling for network and receiver aware adaptation," *IEEE Transactions on Multimedia*, vol. 7, no. 3, pp. 471– 479, June 2005.

[125] Q. Zhang, Z. Ji, W. Zhu, and Y.-Q. Zhang, "Power-minimized bit allocation for video communication over wireless channels," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 6, pp. 398–410, Jun. 2002.

[126] V. Lappalainen, A. Hallapuro, and T. Hamalainen, "Complexity of optimized H.26L video decoder implementation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 717–725, Jul. 2003.

[127] S. Saponara, K. Denolf, G. Lafruit, C. Blanch, and J. Bormans, "Performance and complexity co-evaluation of the advanced video coding standard for cost-effective multimedia communications," *EURASIP Journal on Applied Signal Processing*, vol. 2004, no. 2, pp. 220–235, 2004.

[128] J. Konrad and E. Dubois, "Bayesian estimation of motion vector fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 9, pp. 910–927, September 1992.

[129] B. Girod, "The efficiency of motion-compensating prediction for hybrid coding of video sequences," *IEEE Journal on Selected Areas in Communications*, vol. 5, no. 7, pp. 1140–1154, August 1987.

[130] E. J. Gumbel, *Statistics of Extremes*. New York: Columbia University Press, 1958.

[131] L. Milstein, D. Schilling, and J. Wolf, "Robust detection using extreme-value theory," *IEEE Transactions on Information Theory*, vol. 15, no. 3, pp. 370–375, May 1969.

[132] H. Blume, G. Herczeg, O. Erdler, and T. Noll, "Object based refinement of motion vector fields applying probabilistic homogenization rules," *IEEE Transactions on Consumer Electronics*, vol. 48, no. 3, pp. 694 – 701, August 2002.

[133] G. Ding and B. Li, "Motion vector estimation using line-square search block matching algorithm for video sequences," *EURASIP Journal on Applied Signal Processing*, vol. 2004, no. 1, pp. 1750 – 1756, 2004.

[134] A. Papoulis, *Probability, Random Variables and Stochastic Processes*. McGraw-Hill Companies, 1991.

VITA

VITA

Limin Liu was born in Fujian, P. R. China. She earned her B.E. degree in electrical engineering in 2001 from Tsinghua University, Beijing, China.

From Fall 2001 to Fall 2002, she studied in Klipsch School of Electrical and Computer Engineering at New Mexico State University. Since Spring 2003, she has been pursuing her Ph.D. degree in the School of Electrical and Computer Engineering at Purdue University, West Lafayette, Indianan. She has been a research assistant in the Video and Image Processing Laboratory (VIPER) under the supervision of Professor Edward J. Delp. Her research work has been funded by a grant from the Indiana Twenty-First Century Research and Technology Fund. Her current research interests include image and video compression (JPEG/JPEG2000/MPEG-2/H.263/MPEG-4/H.264/AVC), image and video post processing and multimedia communications.

In the summer of 2005, she worked as a summer intern at Thomson Corporate Research, Indianapolis, Indiana. She worked on the design and development of Advanced 4:4:4 Profile, which was adopted by the Joint Video Team (JVT) and became a new profile in H.264. In the summer of 2006, she worked at Sun Microsystems Laboratories, Menlo Park, California, as a summer intern. In the summer of 2007, she worked as an intern at IBM Thomas J. Watson Research Center, Yorktown Heights, NY on Wyner-Ziv coding with particular interest on the mode decision in Wyner-Ziv video coding.

Limin Liu is a student member of the IEEE professional society and Women in Engineering. Limin Liu is a member of the Eta Kappa Nu (HKN) national electrical engineering honor society.

Limin Liu's publications for her research work include:

Book Chapters:

1. **Limin Liu**, Fengqing Zhu, Marc Bosch, and Edward J. Delp, "Recent Advances in Video Compression: What's Next? " book chapter in *Statistical Science and Interdisciplinary Research: Pattern Recognition, Image Processing and Video Processing*, Eds. Bhabatosh Chanda, et al, World Scientific Press, 2007

Journal papers:

1. **Limin Liu**, Zhen Li, and Edward J. Delp, "Video Surveillance Using Backward Channel Aware Wyner-Ziv Video Coding," submitted to *IEEE Transactions on Circuits and Systems for Video Technology*.
2. Zhen Li, **Limin Liu**, and Edward J. Delp, "Rate Distortion Analysis of Motion Side Estimation in Wyner-Ziv Video Coding " *IEEE Transactions on Image Processing*, Vol. 16, No. 1, January 2007, pp.98-113.
3. Zhen Li, **Limin Liu**, and Edward J. Delp, "Wyner-Ziv Video Coding with Universal Prediction," *IEEE Transactions on Circuits and Systems for Video Technology*, Vo. 16, No. 11, November 2006, pp.1430-1436.

Conference papers:

1. **Limin Liu**, Yuxin Liu, and Edward J. Delp, "Enhanced Intra Prediction Using Context-Adaptive Linear Prediction, " accepted by *the Picture Coding Symposium*, November, 2007, Lisbon, Portugal.
2. **Limin Liu**, Zhen Li, and Edward J. Delp, "Complexity-Rate-Distortion Analysis of Backward Channel Aware Wyner-Ziv Video Coding, " *Proceedings of the IEEE International Conference on Image Processing*, September 16-19, 2007, San Antonio, TX.
3. **Limin Liu**, Fengqing Zhu, Marc Bosch, and Edward J. Delp, "Recent Advances in Video Compression: What's Next? " *Proceedings of the IEEE International Symposium on Signal Processing and its Applications*, February 12-15, 2007, Sharjah, United Arab Emirates (Plenary Paper).

4. **Limin Liu**, Yuxin Liu, and Edward J. Delp, "Content-adaptive Motion Estimation for Efficient Video Compression, " *Proceedings of the SPIE International Conference on Video Communications and Image Processing*, January 28 - February 1, 2007, San Jose, CA.

5. **Limin Liu**, Zhen Li, and Edward J. Delp, "Complexity-constrained Rate-distortion Optimization for Wyner-Ziv Video Coding, " *Proceedings of the SPIE International Conference on Multimedia on Mobile Devices*, January 28 - February 1, 2007, San Jose, CA.

6. **Limin Liu**, Zhen Li, and Edward J. Delp, "Backward Channel Aware Wyner-Ziv Video Coding, " *Proceedings of the IEEE International Conference on Image Processing*, October 8-11, 2006, Atlanta, GA, pp. 1677-1680.

7. **Limin Liu** and Edward J. Delp, "Wyner-Ziv Video Coding Using LDPC Codes, " *Proceedings of the IEEE 7th Nordic Signal Processing Symposium*, June 7-9, 2006, Reykjavik, Iceland.

8. **Limin Liu**, Pau Sabria, Luis Torres, and Edward J. Delp, "Error Resilience in Network-Driven Wyner-Ziv Video Coding, " *Proceedings of the SPIE International Conference on Video Communications and Image Processing*, January 15-19, 2006, San Jose, CA.

9. Zhen Li, **Limin Liu**, and Edward J. Delp, "Wyner-Ziv Video Coding with Universal Prediction," *Proceedings of the SPIE International Conference on Video Communications and Image Processing*, January 15-19, 2006, San Jose, CA.

10. Zhen Li, **Limin Liu**, and Edward J. Delp, "Wyner-Ziv Video Coding: A Motion Estimation Perspective," *Proceedings of the SPIE International Conference on Video Communications and Image Processing*, January 15-19, 2006, San Jose, CA.

11. **Limin Liu**, Yuxin Liu, and Edward J. Delp, "Network-driven Wyner-Ziv Video Coding Using Forward Prediction," *Proceedings of the SPIE International Con-*

ference on Image and Video Communications and Processing, January 16-20, 2005, San Jose, CA, Vol. 5685, pp.641-651.