DECODER-LEARNING BASED DISTRIBUTED SOURCE CODING FOR HIGH-EFFICIENCY, LOW-COST AND SECURE MULTIMEDIA COMMUNICATIONS

A Dissertation presented to the Faculty of the Graduate School
University of Missouri

In Partial Fulfillment of the Requirements for the Degree

Doctor of Philosophy

by
WEI LIU

Dr. Wenjun Zeng, Dissertation Advisor
DECEMBER, 2008

The undersigned, appointed by the Dean of the Graduate School, have examined the dissertation entitled

DECODER-LEARNING BASED DISTRIBUTED SOURCE CODING FOR HIGH-EFFICIENCY, LOW-COST AND SECURE MULTIMEDIA COMMUNICATIONS

Presented by Wei Liu

A candidate for the degree of Doctor of Philosophy

And hereby certify that in their opinion it is worthy of acceptance.

Professor Wenjun Zeng

Professor Yi Shang

Professor Yunxin Zhao

Professor Xinhua Zhuang

Professor Zhihai He

ACKNOWLEDGEMENTS

I would like to first sincerely thank my advisor, Dr. Wenjun Zeng, for his help, support and guidance during my Ph.D. study at University of Missouri. He has encouraged me to not only be trained as a qualified engineer, but also grow as an independent thinker. His vision and motivation are also paramount. Without his mentorship, I could not have ever reached the heights or explored the depths of this work. I would also like to thank my other committee members, in alphabetical order, Dr. Zhihai He, Dr. Yi Shang, Dr. Yunxin Zhao and Dr. Xinhua Zhuang, for their invaluable comments and guidance for this work.

I would like to gratefully acknowledge the support from my family. I especially thank my wife, Lina Dong, for her constant encouragement and unwavering love. I also want to express my sincere gratitude to my parents and Lina's parents, for their faith in us and their support and encouragement when we are studying abroad.

I thank all the members of the MNMC lab, past and present, for the enjoyable stay in the past four years. I am especially thankful to the people that I have cooperated with, listed in alphabetical order: Christine Dartigue, Hongbing Jiang, Aleksandre Lobzhanidze, Mian Pan, Qia Wang, Qiuming Yao, Zhiming Zhang and Yingnan Zhu, for their unselfish assistance and helpful suggestions.

I thank the Department of Computer Science for the enduring effort to provide high-class education and to create an environment that values innovation and creativity.

And finally, let me thank all the friends and colleagues for their suggestions, comments and friendship.

The research was supported in part by NSF grant CNS-0423386, in part by the Gilliom Cyber Security Fellowship, and in part by a grant from the University of Missouri System Research Board.

TABLE OF CONTENTS

CHAPTER 4

CHAPTER 5

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Multimedia applications are becoming more and more an integral part of our daily lives. For most multimedia applications, high-performance compression and cost-efficient communication of the multimedia data are essential. Conventional image and video compression leverages the source statistics at the encoder side, which is not suitable for the so-called up-link transmissions. In such emerging applications (e.g. wireless sensor networks, video surveillance and camera arrays), the encoders usually have limited functionalities and power supplies. Therefore it is desired to shift the burden of exploiting source dependency to the decoder side. The resulting new coding paradigm is called distributed source coding (DSC).

Most existing works on practical DSC only achieve good results when dealing with ideal sources, where *a priori* knowledge about the source statistics is assumed. For real-world sources such as images and videos, such knowledge is not really available in general, and it is very difficult for the decoder to learn the source correlation accurately because there is less information available. For example, in distributed video coding (DVC), decoder-side motion estimation (ME) is employed to generate motion compensated prediction (MCP) for decoding. Without access to the current frame, the decoder has to derive the motion information through temporal-domain extrapolation. This hurts the performance of MCP, as well as the coding efficiency of DVC.

In this dissertation, we focus on designing decoder-side learning schemes for better

understanding of the source statistics, based on which practical DSC systems can be built for high-efficiency, low-cost, and secure multimedia communications. For DVC, we propose to use multi-resolution motion refinement (MRMR) for the decoder-side motion learning, where low-resolution versions of the current frame are progressively decoded, based on which the motion field is refined and used for the decoding of the next resolution level. We will present both theoretical analysis and a practical wavelet-domain codec. It is observed that huge bit-rate saving can be achieved over motion extrapolation based approaches. On the other hand, unlike conventional ME, decoder-side ME does not have to transmit the overhead motion information, making it possible to further improve MRMR by exploiting more detailed motion (e.g. describing the motion field with higher spatial resolution and amplitude precision, using more candidate motion vectors, etc.). Our MRMR predictor with extensive motion exploration has achieved performance comparable to the H.264/AVC predictor.

Similar idea can be employed in compression of encrypted images or videos. In such applications, it is assumed that the encoder does not have access to the secret key, therefore distributed coding can be applied to enable decoder-side source dependency exploitation. In this dissertation, we propose the use of resolution-progressive compression, where low-resolution reconstructions are used for the learning of both intra-frame and inter-frame correlations, without any assumption of the underlying source/motion models. Our practical lossless codec for encrypted images/videos has shown significant advantages in both coding efficiency improvement and complexity

reduction, when compared to existing approaches.

In real-world applications, power optimization is an equally important issue as bit-rate reduction. In this dissertation, we also address the fundamental rate-allocation problem for distributed coding of multiple correlated sources. The goal is to find the optimal rate-point that allows lossless reconstruction of the sources, while minimizing the overall transmission power consumption of a wireless sensor network. A novel water-filling model is established, based on which a greedy yet optimal algorithm is proposed for the decoder to solve the rate-allocation problem in a recursive manner. The feasibility and optimality of the proposed solution are analyzed mathematically. Compared to the exhaustive search approach, our algorithm achieves dramatic reduction in computational complexity.

Chapter 1

Introduction

Nowadays, with the development of digital technologies and the popularity of the Internet, video applications, such as digital video disk (DVD), digital video camcorder and Video on Demand (VoD) are becoming more and more inseparable from human lives. In most video applications, due to the huge amount of data, compression or coding of the video signals is an important issue.

Most conventional video coding standards are designed for the “downlink” transmission, where there is a powerful encoder that exploits the source statistics, and multiple light-weight decoders that work in a slave mode. It is suitable for the traditional server-client model of video communications (e.g. broadcasting or VoD), where a video is to be encoded once, and decoded many times.

However, there are also some application scenarios where this conventional “complex encoding, simple decoding” structure shall be reversed. For example, in wireless sensor networks (WSN) [14] which consist of many tiny sensors with integrated computing and wireless communication capabilities, the energy provisioned for the wireless sensors is not expected to be easily renewable throughout its lifetime. To transmit video over a WSN, it is desired to shift the bulk of computation to the decoder side (which can be a powerful base station of the network) to save the power for computation, but still reduce the data rate as much as possible to save the power for communication. This is the so-called “up-link” video transmission, where “simple encoding, complex decoding” is favored.

Distributed source coding (DSC) is a new coding paradigm that matches this requirement perfectly. DSC encodes multiple correlated sources separately, while decodes them jointly. It is the decoder who bears the responsibility to exploit the source dependency. Theoretical results in the literature show that there can be no rate loss (for lossless DSC and some special lossy DSC cases) or a very small rate loss (for general lossy DSC cases) compared to conventional source coding [97][110][111].

Another application scenario that DSC has advantages over conventional source coding is the compression of encrypted sources. Imaging that a third-party network provider is asked to transmit some encrypted files for a user, while the user wants to keep the files confidential to the network provider. In this case, it is not possible for the encoder to exploit the source dependency because it has been masked by the encryption function. But it is still possible for the decoder to do the job if it holds the secret key and performs joint decoding and decryption. The scheme to compress encrypted sources is a sub-problem of DSC – source coding with side information (in this case the side information is the secret key) at the decoder side. Theoretical analysis shows that under some reasonable assumptions, neither compression performance nor information-theoretic security will be sacrificed to compress the encrypted data [50].

Despite the inspiring theoretical results and the advances of the coding practices for ideal sources (the reader is referred to [118] for a literature review), when it comes to the distributed compression of real-world sources (e.g. distributed video coding (DVC) [40] or compression of encrypted images or videos [92]), there is still a large performance gap when compared to conventional codecs. The problem can be explained as follows: 1) the performance of DSC relies largely on the knowledge of the source statistics; and 2) due to the non-stationarity of real-world sources, it's hard to learn the statistics if the decoder does not have access to the (decoded) sources. This chicken-and-egg dilemma has imposed a significant challenge for the design and deployment of practical DSC schemes.

In this dissertation, significant efforts have been made to improve decoder's learning about the source statistics for efficient, low-complexity and secure communication of multimedia data [63]–[73][121][123]. The major contributions are summarized as follows:

- We address the limitation of decoder-side learning and propose to enable partial access to the current source through progressive decoding.
- For DVC applications, we propose a multi-resolution motion refinement (MRMR) scheme for decoder-side motion learning. That is, the current frame is progressively decoded in the resolution dimension, based on which the motion is refined to facilitate the decoding of the next resolution level. We provide a comprehensive rate-distortion analysis on the efficiency of MRMR, and implement a wavelet-domain DVC codec based on it. It has been shown to have

significant improvement over conventional motion learning methods that employ temporal domain extrapolation.

- Further improvement is made for MRMR by incorporating extensive motion exploration. That is, more detailed motion is exploited at the decoder side, including the use of fractional-pel motion search, the use of smaller block sizes and the use of multiple-hypothesis prediction (note that decoder-side motion estimation does not suffer from the overhead bits in transmitting the motion). With these advanced motion estimation techniques integrated, MRMR has been shown to have comparable prediction performance to H.264/AVC.
- We analyze the limitations of existing approaches on compressing encrypted images, and proposed progressive decoding for better learning about the local statistics and geometric features of the image. Theoretical analysis shows that the proposed scheme can achieve 70% to 90% possible rate saving of an optimum intra coder. Our real-world lossless image codec has achieved both much improved coding efficiency and reduced computational complexity than existing approaches.
- We extend our work to compression of encrypted videos, where partially reconstructed frames are used for both intra-frame and inter-frame prediction. They are adaptively integrated to obtain a hybrid spatial-temporal prediction. Simulation results show that our scheme saves 1.6 bpp more than existing solutions.

- We studied the problem of power-efficient communication of multiple correlated sources over a WSN. Sophisticated rate allocation can be performed at the decoder side to minimize the power consumption of the entire network. We propose a water-filling model for this problem, based on which a greedy, yet optimum rate allocation algorithm is proposed. The algorithm has achieved dramatic complexity reduction over the exhaustive search approach.

The dissertation is organized as follows. Some background knowledge is introduced in Chapter 2. Chapter 3 presents the multi-resolution motion refinement scheme for DVC. Chapter 4 studies compression of encrypted images and videos. Chapter 5 introduces our power-efficient rate allocation algorithm. Conclusions and future works are discussed in Chapter 6.

Chapter 2

From Conventional Video Coding to Distributed Video Coding

Digital video signals are captured and displayed as a sequence of two dimensional (2D) images [80]. Each image is usually called a *frame*. *Frame rate* denotes the temporal sampling frequency of the video. In the spatial domain, a frame is represented as a discrete array of *pixels* (or *pels*). The size of the array characterizes the *spatial resolution* of the video. For colored videos, each pixel is described using the three *components* in a tri-chromatic space, such as the RGB, YUV or YCbCr spaces [105][113]. The amplitude of each component is typically digitized into 8 ~ 16 bits of precision.

The data amount of uncompressed digital videos can be huge. For instance, the CIF (Common Intermediate Format) is specified by ITU-T (International Telecommunications Union - Telecommunications Sector) for video conferencing purposes over Internet. One second of a CIF signal contains 30 frames, each of which has 352×288 pixels. Each pixel is represented in the YCbCr format, with the Cb and Cr components subsampled by one half in both the horizontal and vertical dimensions,[1] and each component is represented with an 8-bit integer. Therefore, without compression, the data rate of a CIF video will be

[1] This is called 4:2:0 sampling.

30×352×288×1.5×8 = 36.5 Mbps (Mega bits per second), which is beyond the bandwidth capability of most commercial networks. For the BT.601 format developed by ITU-R (International Telecommunications Union - Radio Sector) for DVD, 249 Mbps or 124 Mbps of raw data rate is typical (depending on the color representation) [105], which means that a two-hour uncompressed program will need 224 Giga Bytes (GB) or 112 GB for storage. This has exceeded the storage capability of most entertainment devices. Certainly with the development of material science, bandwidth and storage capabilities of future devices will be dramatically improved. However, consumers' anticipation to video quality is also increasing. Today's high definition television (HDTV) aims at providing both enhanced spatial resolution (for example 1920×1280) and temporal resolution (for example 60 fps) [47]. For digital cinema, it is 4096×2048 at 48 fps, with the amplitude precision being 12 bits per component [29]. There are also 3D TV and 3D films on the way. At least in the foreseeable future, the advances of bandwidth and storage have no way to keep up with consumers' explosive demand in video applications. Hence, video compression has been such a hotspot of research over the last two decades.

Compression of video signals can be lossless or lossy. The former requires the reconstructed signal to be perfectly identical to the original, usually for data archiving purposes [35]. The compression ratio of lossless video coding is relatively low, ranging from 1/5 to 1/2. For commercial video applications, lossy compression is more often adopted to intentionally discard some visual information that human eyes can hardly perceive. As pointed out by Shannon [96], lossy compression is a rate-distortion (R-D) problem. A video codec needs to provide the best quality of video within a given rate budget, or compresses the video as much as possible but still maintains a minimum fidelity constraint.

However, R-D is not the only tradeoff that people make in practical video compression. There are also other issues to be jointly considered, such as encoder and decoder's complexity, delay constraint, buffer requirement, scalability to bandwidth variations and robustness against channel errors, etc. Different applications might have different preferences on these issues. For example, in video broadcasting applications, a video stream is to be encoded only once and to be decoded many times. For such a server-client scenario, it is appropriate to design a complex encoder to improve the coding efficiency of the video, while let the decoder be relatively simple, working in a "slave" mode of the encoder. On the other hand, there are also a lot of emerging applications that requires "simple encoding, complex decoding", such as wireless sensor networks (WSN) [14], low-complexity wireless video, video surveillance and camera arrays. In this case people turn to distributed video coding (DVC) [40], a radically new coding paradigm.

In the rest of this chapter, we first give a brief review of conventional video coding techniques in Section 2.2; then in Section 2.3, distributed source coding (DSC) [118] will be introduced as the theoretical foundation of DVC; we shall take a closer look at existing DVC structures in Section 2.4 and discuss its performance loss when compared to conventional video coding.

2.1 Conventional Video Coding

For uncompressed video data, redundancy exists in both the spatial domain and the temporal domain. Video compression is achieved by removing such redundancy. Conventional video coding schemes can be classified into two categories: intra-frame coder – which only deals with the spatial domain redundancy, and inter-frame coder – which seeks to remove both the spatial domain and temporal domain redundancy.

2.1.1 Intra-frame coding

One example of intra-frame video coder is motion-JPEG. The basic idea is to encode each frame independently using the still image coding standard JPEG [1], where each frame is divided into 8×8 blocks, and the blocks are transformed by the Discrete Cosine Transform (DCT) [13]; the resulting DCT coefficients are quantized and the quantization indices are entropy coded using variable length coding (VLC) [28]. Although the coding efficiency of motion-JPEG is low, it is still widely used in video recording devices and video editing systems because of the inherited nature of intra-frame coding: low encoding complexity and random accessibility [103].

In recent years, there are new intra-frame coding standards such as motion-JPEG 2000 [7] and H.264/AVC [8] INTRA mode. The former employs Discrete Wavelet

Transform (DWT) [77] and the latter enables intra-frame prediction [104] plus DCT to exploit spatial domain redundancy. They both provide improved coding efficiency over motion-JPEG, and share the similar characteristics as intra-frame coding.

2.1.2 Hybrid structure

Better coding performance can be achieved if the temporal domain redundancy is also exploited. A successful example of inter frame coding is the so-called hybrid video coding structure.

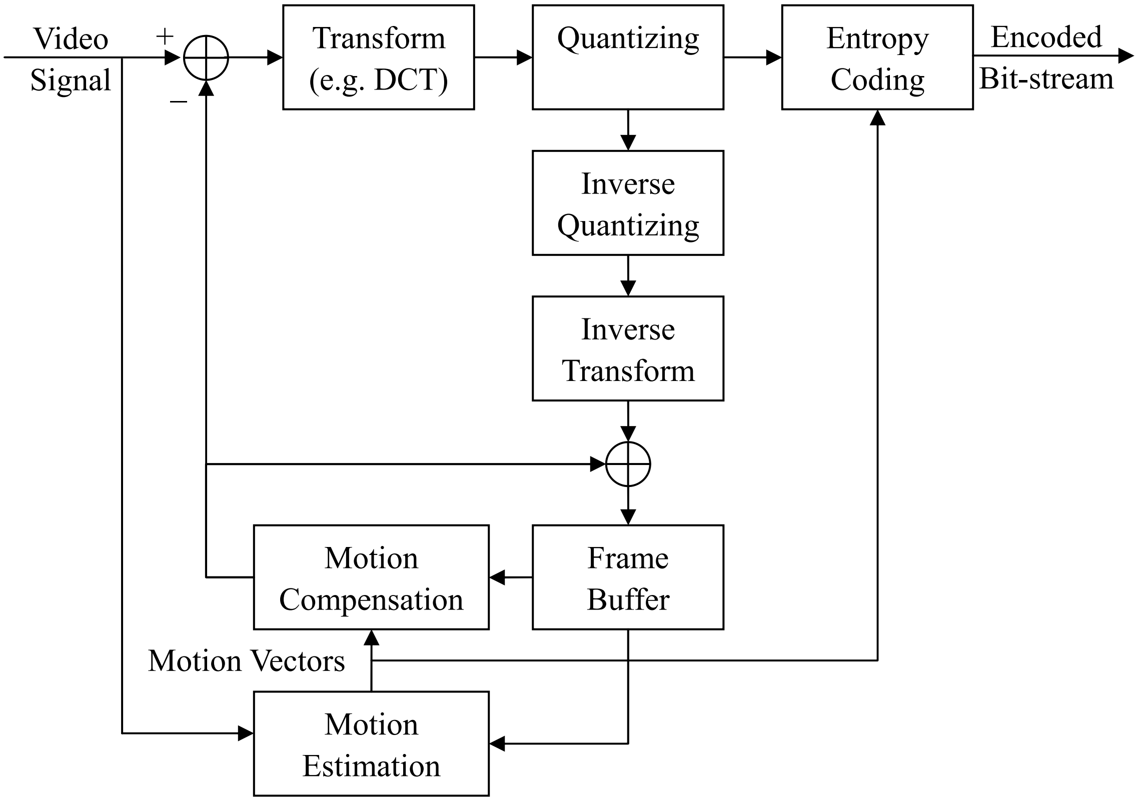


Figure 2.1. General diagram of hybrid video coding.

Empirically, there is usually very little difference between neighboring frames in a video sequence, except that some moving objects have their positions changed – which is called motion. If the motion information is available, the current frame can be well predicted from the reference frame. Based on this idea, one can come up with an efficient video codec as illustrated in Figure 2.1, where the coder first performs motion estimation (ME) and predicts the current frame using motion compensated prediction (MCP) to remove the temporal redundancy; the residual frame is then coded using transform coding to further remove the spatial redundancy. This coding structure is also known as the hybrid coding structure, i.e., inter-frame predictive coding plus intra-frame transform coding.

Hybrid video coding has achieved a tremendous success in the last twenty years. Almost all existing video coding standards (MPEG-1 [3], MPEG-2 [4], MPEG-4 [5], and H.261 [2], H.263 [6], etc.) are based on this structure. MPEG-2 and H.263 are two successful examples: the former is widely used in applications with high bit-rates (4-30 Mbps) such as DVD and HDTV, and the latter is mainly used in real-time or interactive applications with relatively low bit-rates (10-2048 kbps). The new international video coding standard H.264 [8] is another milestone: the compression rate can be twice as much as that of MPEG-2, while keeping the same objective and subjective quality of the video signal. Considering that H.264 essentially shares the same coding structure with MPEG-2/H.263, it is difficult to achieve another factor-of-two improvement. It is also noticed that hybrid video coding has some inherit drawbacks. For example, it relies on a powerful encoder that can afford the heavy computational burden, which is not available in the so-called “uplink” video transmissions, and the predictive nature makes the bit-stream vulnerable to channel errors (because of the error propagation problem).

2.1.3 Block-matching motion estimation

ME plays an important role in video coding. Conceptually, the more accurate the motion information is, the less the residual information needs to be coded, and the higher the coding efficiency is (if the number of bits spent in encoding the motion information are not counted). A theoretical analysis on the relationship between motion accuracy and the efficiency of MCP is found in [37], and is reviewed in Chapter 3 of this thesis.

The motion between two frames can be estimated using optical-flow based methods [17] or block-matching based methods. The former estimates a dense motion field and derives MV for each individual pixel; while the latter estimates the motion field on a block basis, i.e., it searches a displaced block in the reference frame to minimize the prediction error. Due to complexity and bit overhead considerations, block-matching algorithms (BMA) are more often used in video coding.

In the literature, various techniques have been proposed to improve ME accuracy. Without exaggerating, the advances in ME techniques have made the major contribution to the coding efficiency improvement achieved by modern video coding standards. In this subsection, we shall give a brief review of those techniques.

A. Fractional-pel precision motion search

In video capturing, objects are projected from a 3D dynamic scene to the image plane of an imaging device. The movement of an object is not necessarily of integer pels between the time instances that two adjacent frames are captured. So it is physically meaningful to have the motion vectors (MV) taking finer than integer precision.

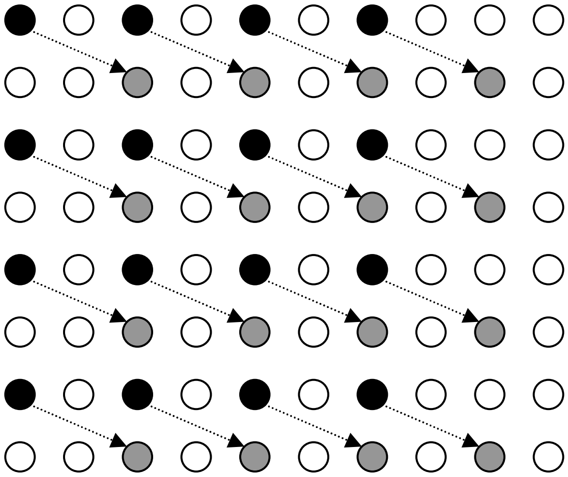


Figure 2.2. An illustration of half-pel motion search. Non-integer samples are pre-interpolated before the motion search.

Fractional-pel motion search is first proposed in [21]. An illustration of half-pel motion search is shown in Figure 2.2, where the black dots form a block in the integer grid, the gray dots are samples interpolated from integer-grid pixels, for a candidate MV (1, 0.5). Finer fractional-pel motion search can be employed to further improve coding efficiency. But both theoretical analysis [38] and coding practices show that efforts beyond quarter-pel precision only provide marginal gain.

In the existing video coding standards, MPEG-1, MPEG-2 and H.263 support half-pel motion search, while MPEG-4 and H.264/AVC support quarter-pel motion search.

B. Multi-frame MCP

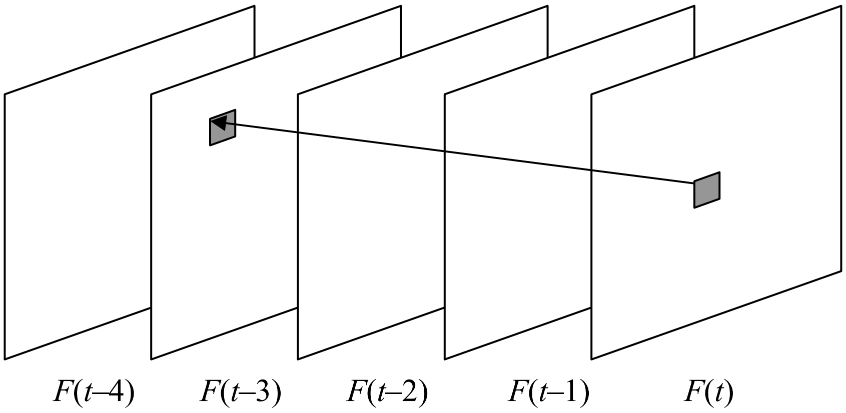


Figure 2.3. An illustration of multi-frame MCP. $F(t)$ is the current frame, which is predicted from one of the four previously reconstructed frames. A temporal tag is needed to indicate which frame the MV is pointing to.

Multi-frame MCP is also known as long-term memory MCP [107]. As illustrated in Figure 2.3, MCP uses more than one previously decoded frame for prediction. This allows the motion estimator to find a good match that may be covered in some of the reference frames.

C. Bipredictive pictures

A bipredictive picture [45] employs both forward and backward prediction. That is, two motion vectors (MV) are searched for an individual block, typically one from the past pictures and the other from the future pictures. The average of the two MCP signals is used as the prediction of the current block. Now we have three frame types: intra-coded (I) frames, predictive-coded (P) frames and bipredictive-coded (B) frames. In Figure 2.4 we show the prediction patterns of these frame types. It is worth noting that an I frame only allows intra-coding; a P frame allows both intra-coding and predictive-coding for each block; while a B frame allows all three types of coding.

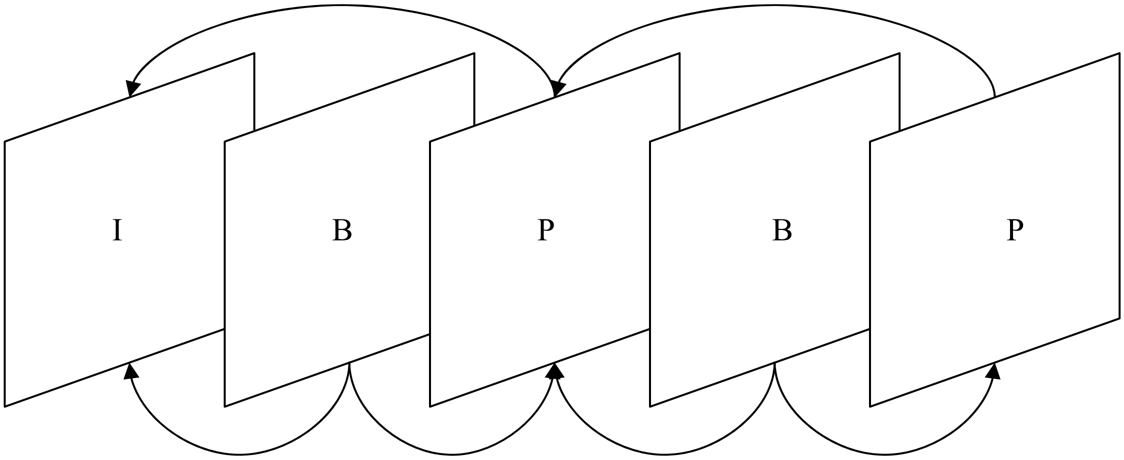


Figure 2.4. An illustration of the predictive patterns for intra-coded frames, predictive-coded frames and bipredictive-coded frames.

Bipredictive MCP was first standardized in MPEG-1. After that, it has been adopted by all other video coding standards. It used to be a rule that a B frame can only be predicted from I frames or P frames. In H.264, hierarchical B extension [93] is defined to allow a B frame to be predicted from other B frames, which provides both higher coding efficiency and temporal scalability.

To decode a B-picture, the decoder has to wait until its reference frames (both past and future references) are decoded. This introduces significant end-to-end delay, especially in real-time or interactive applications.

D. Multiple-hypothesis prediction

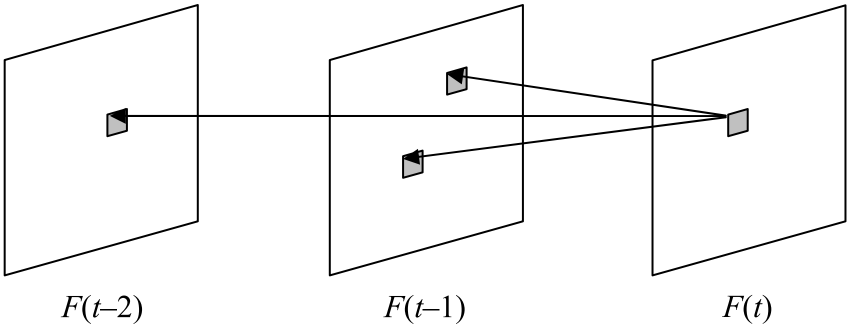


Figure 2.5. An example of 3-hypothesis prediction. *F*(*t*) is the current frame. A block in *F*(*t*) is predicted from three motion-compensated blocks in two previously reconstructed frames.

Multiple-hypothesis prediction (MHP) [32][39][101] is a technique to use the linear combination of multiple MCP signals. B-picture coding is a special case of MHP. In Figure 2.5 we illustrate an example that the current block is predicted from three motion compensated blocks in two past frames. Note that two of them are from the same reference frame. In fact, even the factional-pel motion search can fit into the MHP architecture.

E. Variable block sizes

The intuition of using variable block sizes in MCP [36] is to trade off the motion accuracy with the number of bits needed to represent the motion field [102]. That is, for regions with unified motion, larger block sizes are selected to save the motion bits; while for regions with irregular motion, smaller block sizes can be used to improve the MCP. In H.264/AVC, a 16×16 macroblock (MB) can be partitioned into 16×16, 16×8, 8×16 and 8×8 regions. If 8×8 partitioning is used, each 8×8 sub-MB can be further partitioned into 8×8, 8×4, 4×8 and 4×4 regions (illustrated in Figure 2.6).

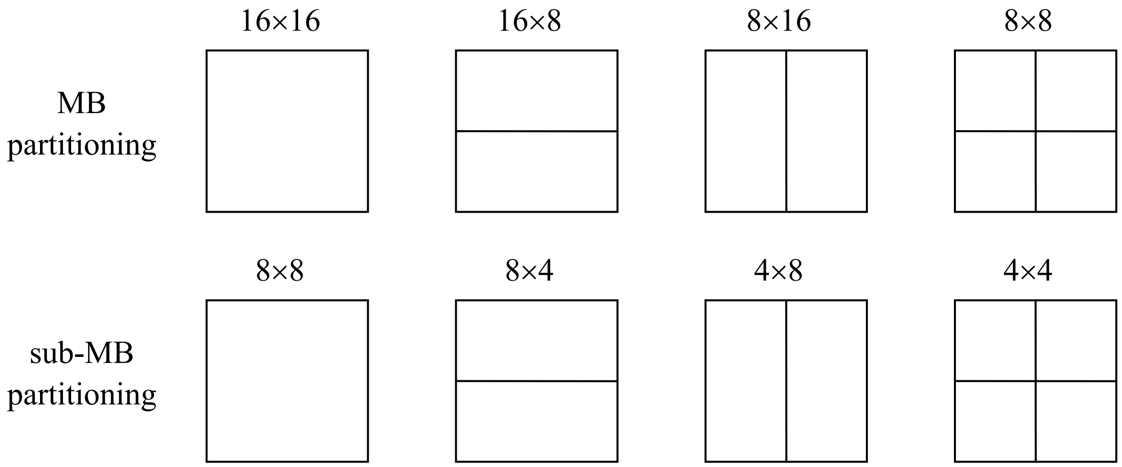


Figure 2.6. MB and sub-MB partitioning modes in H.264/AVC.

F. Overlapped block motion compensation and in-loop filtering

Lossy video compression can introduce some annoying artifacts. A well known artifact is the block artifact. Block artifact not only degrades the visual quality, but also affects the motion search accuracy, since it introduces some non-existing geometric features to the decoded reference frame.

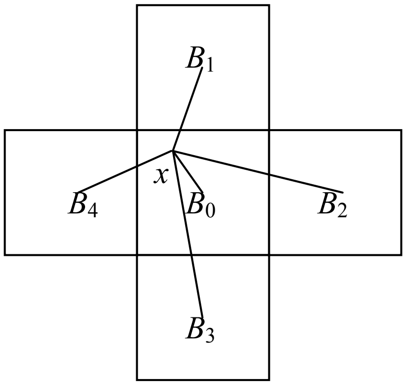


Figure 2.7. An illustration of overlapped block motion compensation using four neighboring blocks. Each block (B_0 ~ B_4) has an MV, then point x has five candidate MCPs. The prediction of x is the weighted average of them. The weighting factors are usually inversely proportional to the distances between x and the centers of the blocks.

With overlapped block motion compensation (OBMC) [83][84], a pixel in a block is not only predicted from the estimated MV that belongs to the block, but also from the MVs of its neighboring blocks. Typically the predictions are weighted inversely proportional to the distances between the pixel and the center of the blocks (see Figure 2.7). In OBMC, the predictions of two neighboring pixels that belong to different blocks are no longer carried out in a totally separated fashion. Hence the block artifact is reduced. Note that OBMC can also be characterized as a form of MHP, with the combination of multiple hypotheses are done in the pixel level.

OBMC is standardized in H.263. It effectively reduces the block artifact produced by MCP, but does not deal with the block edge discontinuity produced by block transform. Hence H.264/AVC instead employs an in-loop filter [61] for deblocking purpose. It is an adaptive filter so the sharpness of true edges is still retained. The in-loop filter achieves 5%~10% bit-rate saving with the same objective quality, and greatly improves the subjective quality [104].

G. Others

Other researches on block-matching motion estimation include the efforts to reduce the computational complexity [48][53][43], to generate a smooth and physically meaningful motion field [46][81][20], to use deformable blocks to adapt to the frame geometric features [94][82][55][56] and frequency-domain ME [54][85]. For a more complete review of ME techniques, the reader is referred to [100].

It is worth noting that although more accurate motion information generates better MCP, it also results in a lot more overhead bits in representing the MVs. In state-of-the-art video coding, such overhead is usually not negligible. Rate-constraint ME [31][108] is recommended to achieve the balance between the rates spent on the prediction residual signal and on MVs.

2.2 Distributed Source Coding

In this section we give a review on distributed source coding (DSC). As opposed to the conventional source coding, which targets to exploits the source redundancy *before* communication, DSC is a technique to exploit the source dependency *at the decoder*. The history of DSC dates back to 1970s. It did not become a hotspot of research until recently, as applications have arisen. For example, data aggregation over wireless sensor networks (WSN) is envisioned for a wide range of applications such as battlefield intelligence, surveillance, reconnaissance, security monitoring, emergency response, disaster rescue, environmental tracking, and tele-medicine. Those applications share a common feature that the encoders are light-weight devices and do not have the capability to exploit source statistics. Therefore it is necessary to shift this job to the decoder side, which fits in with the architecture of DSC very well.

DSC can be classified into two categories: lossless DSC and lossy DSC. Lossless DSC is also called Slepian-Wolf coding (SWC). Lossy DSC is an R-D problem, of which the general bound has not been found. However, one of its sub-problems, lossy source coding with side information (SI) at the decoder (which is also called Wyner-Ziv coding or WZC), has been well studied. We will discuss SWC and WZC in the next two subsections.

2.2.1 Slepian-Wolf coding

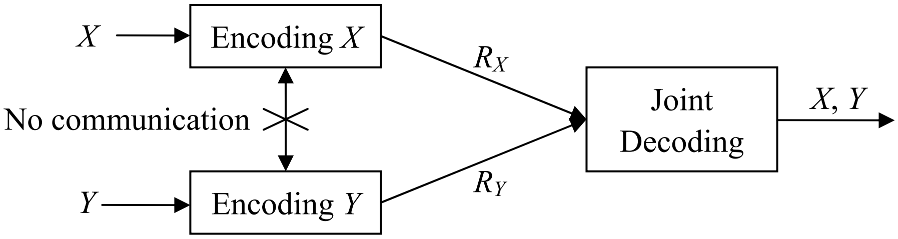


Figure 2.8. An illustration of Slepian-Wolf coding. The two encoders do not communicate with each other. The decoder jointly decodes *X* and *Y*, where the source statistics are exploited.

The problem of lossless DSC of finite-alphabet sources goes back to Slepian and Wolf’s 1973 paper [97] (see Figure 2.8). It is proved that there is no performance loss to compress two statistically dependent sources even if the two encoders do not communicate with each other. More specifically, for two sources *X* and *Y*, as long as the two encoders work at the so-called Slepian-Wolf region:

$$\begin{cases} R_X \geq H\left(X \middle| Y\right) \\ R_Y \geq H\left(Y \middle| X\right) \\ R_X + R_Y \geq H\left(XY\right) \end{cases} \tag{2.1}$$

X and *Y* can be reconstructed at the decoder with no error. Here *H*(·) denotes the Shannon entropy function. An illustration of two-source Slepian-Wolf region is shown in Figure 2.9. Nowadays, SWC has been extended to arbitrary number of discrete sources with finite entropy (the alphabet is not necessarily finite) and similar conclusion can still be drawn. A more general form of Eq. (2.1) can be found in Chapter 5.

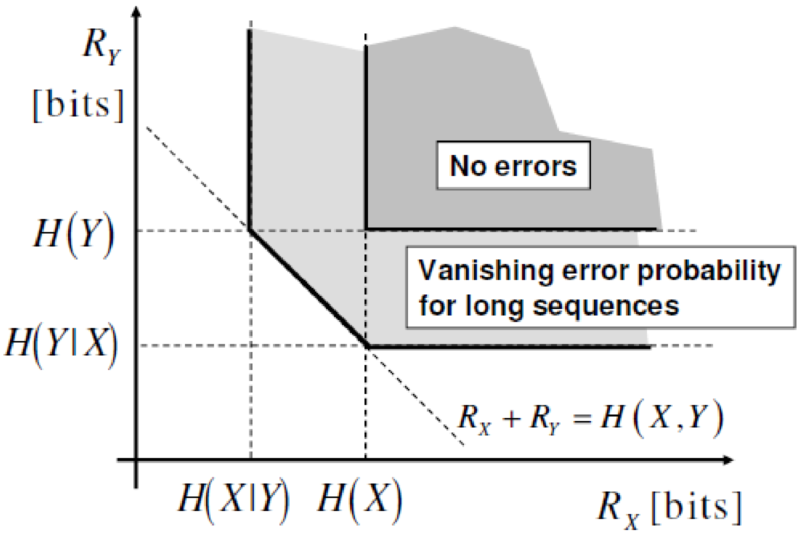


Figure 2.9. Achievable rate region for two-source SWC. From [40] Fig. 2.

As many other theorems in information theory, the proof of SWC itself is not constructive. The first constructive approach of SWC was proposed by Wyner [110] using channel coding, which is roughly described as follows: one encoder compresses *Y* to the rate of *H*(*Y*) using conventional source coding, and the other apply a systematic error correction code (ECC) on *X*, but only transmit the parity bits at the rate of *H*(*X*|*Y*); the decoder treats the systematic bits of *Y* as a noisy version of *X* and tries to correct them using the ECC. This approach is usually referred to as compression with side information at the decoder.

Twenty years after that, practical SWC design became one of the hotspots of research. In 1999, Pradhan and Ramchandram [87] proposed a scheme called Distributed Source Coding Using Syndromes (DISCUS). This scheme divides the codewords of *X* into cosets and only transmits the syndrome of the coset instead of a real codeword. The decoder chooses the codeword from the given coset that is the closest to the known side information *Y* to reconstruct *X*. Since then, state-of-the-art channel coding techniques have been employed in SWC. For example, Zhao and Garcia-Frias [122], Mitran and Bajcsy [78], and Aaron and Girod [9] proposed SWC schemes based on turbo codes [19]. Liveris, Xiong and Georghiades [75] used LDPC codes [33] and Varodayan, Aaron and Girod designed rate-adaptive LDPC accumulative (LDPCA) [114] codes for SWC. Because of the near-capacity performance of modern channel codes, state-of-the-art SWC techniques have achieved a coding efficiency very close to the theoretical Slepian-Wolf bound [118]. Further advances in practical SWC design make it possible to achieve any point inside the Slepian-Wolf region [24][92][99].

2.2.2 Wyner-Ziv coding

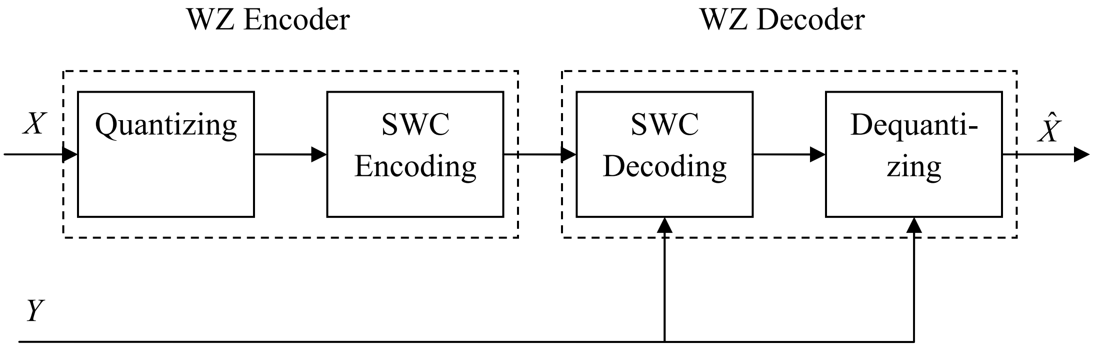


Figure 2.10. A practical Wyner-Ziv coding system.

As we have mentioned, WZC is a sub-problem of lossy DSC. It was first studied by

Wyner and Ziv [111][112]. The R-D bound for WZC, $R_{X|Y}^{WZ}(D)$, is generally larger than the conventional source coding with side information at the encoder $R_{X|Y}(D)$. However, the rate loss $R_{X|Y}^{WZ}(D) - R_{X|Y}(D)$ is zero if X and Y are jointly-Gaussian, and the mean square error (MSE) is used for distortion measure. In this case, the R-D bound for WZC is

$$R_{X|Y}^{WZ}(D) = \begin{cases} \frac{1}{2}\log\left(\frac{\sigma_{X|Y}^2}{D}\right) & (D \le \sigma_{X|Y}^2) \\ 0 & (D > \sigma_{X|Y}^2) \end{cases} \tag{2.2}$$

This result has triggered lots of research interest because most of multimedia data can be modeled as jointly Gaussian.

WZC is actually a source-channel coding problem. To design a Wyner-Ziv coder, one can first quantize the source into a finite alphabet and use a SWC to encode the quantization indices into syndromes (see Figure 2.10). Most of the practical designs of WZC so far are within this framework. Fleming, Zhao and Effros [30] proposed to quantize the source possibly with quantization index reusing. Rebello-Monedero, Zhang and Girod [90] proposed a Lloyd algorithm to design optimal quantizers for WZC. Mitran and Bajcsy [79] proposed a turbo-like WZC scheme by using scalar quantization followed by parallel concatenation of Latin Square based encoders and an iterative turbo decoder. Xiong et al. [119] designed a WZC by using nested lattice quantization [117] followed by an LDPC based SWC, the resulting 1D and 2D Wyner-Ziv coders provide similar R-D performance with conventional source coders [118].

2.3 Distributed Video Coding

With the theoretical results and coding practices in DSC, people are inspired to extend distributed coding to real-world signals such as videos. Most up-to-date DVC schemes are based on the WZC architecture, so they are also called Wyner-Ziv video coding (WZVC). In this section, we will first take a closer look at WZVC. Then some practical applications of DVC are introduced.

Unlike DSC for ideal sources, today's DVC still suffers a non-negligible performance loss when compared to conventional hybrid video coding. We will analyze the rate loss in WZVC, based on which we propose our solution to improve the performance of DVC throughout the thesis.

2.3.1 Wyner-Ziv video coding

In Figure 2.11, a general diagram of WZVC is illustrated. In WZVC, a subset of the frames is compressed as I-frames using conventional intra-frame coding. Other frames, referred to as WZ frames, are decomposed using conditional Karhunen-Loève transform (KLT) [34] (or approximated by DCT or DWT, or even left un-transformed in some low-complexity WZVC), quantized and Slepian-Wolf encoded. The generated syndrome bits are stored in a buffer in the encoder, and are sent to the decoder based on its request. The decoder first decodes the I frames and use them as references. SI is generated for each WZ frame based on the I frames and possibly other previously decoded WZ frames. We will return to the details of SI generation in the next subsection. The SI can be treated as good approximation of the current frame, but contains some “errors”. The Slepian-Wolf decoder serves to “correct” those errors and reconstruct the quantization indices, based on which dequantization and inverse transform are performed, and the whole frame is decoded.

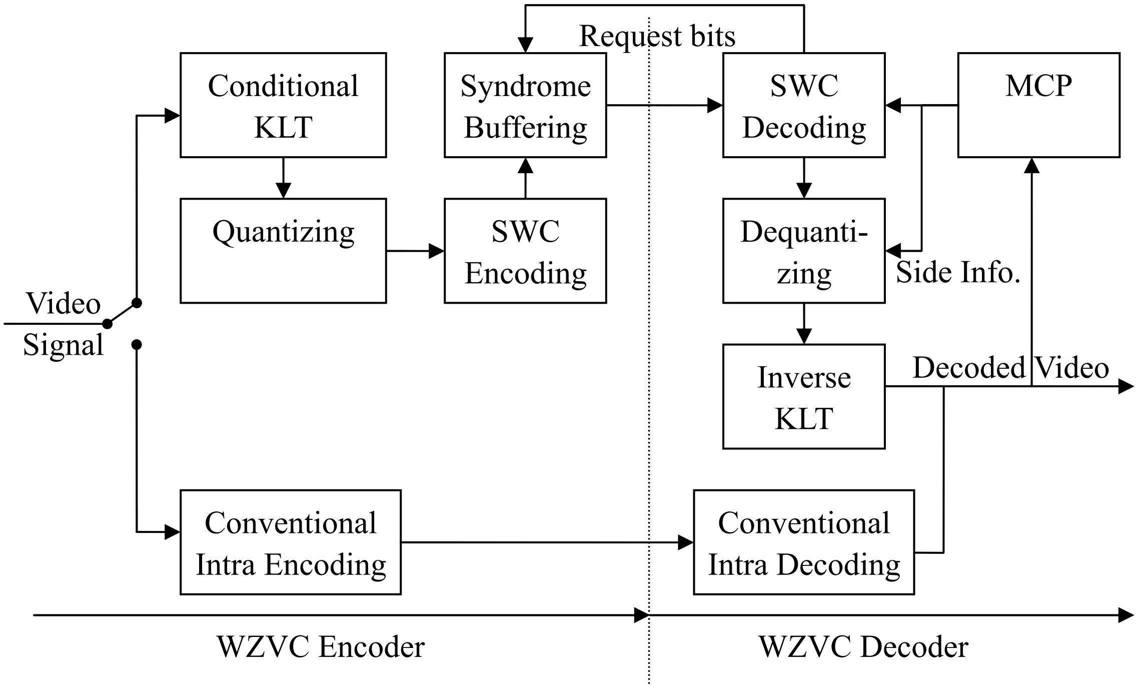


Figure 2.11. Diagram of Wyner-Ziv video coding.

2.3.2 Performance loss in WZVC

Unlike for ideal sources, distributed coding for real-world video signals suffers a large performance gap with respect to conventional video coding. As pointed out in

[57][59], WZVC suffers from three types of performance loss in comparison with conventional hybrid video coding. The first is *system loss*, which is due to the rate loss $R_{X|Y}^{WZ}(D) - R_{X|Y}(D)$ we mentioned in subsection 2.2.2. The second is *source coding loss*, due to the inefficiency of practical SWC to achieve the Shannon bound. The third is *video coding loss*, which is due to the imperfect MCP at the decoder side.

One significant difference between decoder-side ME and encoder-side ME is that the decoder does not have access to the current frame. Then how do we come up with the motion? The simplest method is to assume the motion is zero and directly use the previously decoded frame as the SI, or to use the (weighted) average of several frames [11].

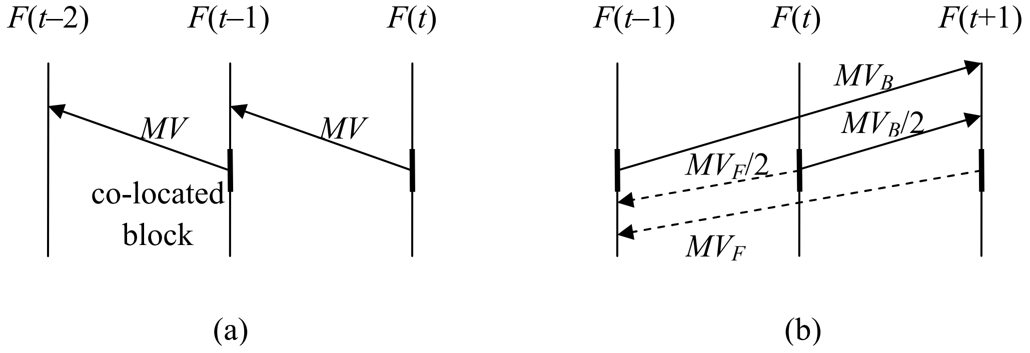


Figure 2.12. 1D illustration of (a) temporal domain motion extrapolation and (b) temporal domain motion interpolation.

More sophisticated approaches employ motion extrapolation or interpolation to reconstruct the motion field of the current frame. In [58], several approaches for decoder-side ME are reviewed and their performances are compared. One typical approach of motion extrapolation is to perform forward ME from $F(t–2)$ to $F(t–1)$; for a block in $F(t)$, the same MV of the co-located block in $F(t–1)$ is used to find its motion-compensated correspondence. A 1-D example of this approach is illustrated in Figure 2.12(a). Motion interpolation is similar, but uses both forward and backward MVs for the prediction (as shown in Figure 2.12(b)).

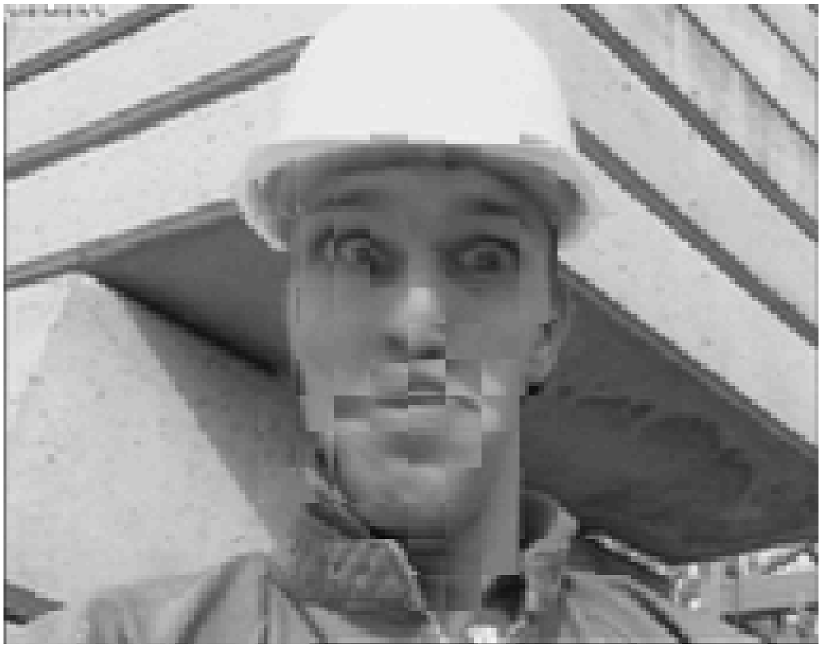


Figure 2.13. MCP result from motion extrapolation.

Without any knowledge about the current frame, motion extrapolation / interpolation approaches basically assume that objects move at a constant speed and the estimated MV’s reflect true motion, which is an over-simplification to the reality. As a result, the estimated side information (SI) is usually not a good approximation to the current frame. An example of the MCP result obtained from motion extrapolation is illustrated in Figure 2.13. We can see that its motion accuracy is pretty bad. Analytical results in [57][59] suggest that WZVC could fall 6dB or more behind conventional video coding in the worst case due to the inaccurate MCP.

Realizing this, researchers have proposed difference approaches to facilitate decoder-side motion search. For example, in [88], the encoder transmits the CRC (cyclic redundancy code) of the quantized symbols to aid the decoder in the determination of the motion through Viterbi decoding. Similar approach is also found in [10], where robust hash codes are used instead of CRC. The drawback of such approaches is that overhead bits are required in the transmission. Backward channel-aware WZVC is proposed in [62], where the decoder feeds back several MVs as candidates for the encoder to choose from. However, real-time interactive communication between the encoder and the decoder might not be available in some WZVC applications.

Approaches involving iterative motion refinement based on partially-decoded results can also be found in the literature. Such approaches can be further characterized into two categories. In the first category [15][57][116], an initial estimation of the SI is first generated by motion extrapolation / interpolation, then the first-pass Wyner-Ziv decoding is performed on the whole frame, leaving whatever decoding error as is in the partially-decoded frame. Then based on this result, inter-frame ME is carried out for the second-pass Wyner-Ziv decoding. The difficulty of such approaches lies in the control of bit-error-rate (BER) of the first-pass decoding. If the BER is too high, the ME result based on the partially-decoded frame might be even worse than the initial SI; on the other hand, if the BER is low, there might not be a significant rate difference when compared to the error-free decoding, since the BER curves for the channel codes employed in WZC are usually very steep.

Another category of approaches is found in [16][76], as well as our own works [67][70]. In our work, we propose to decode the current frame progressively in the resolution or the quality dimension. Once a low-resolution or low-quality version of the current frame is reconstructed, inter-frame ME is performed with respect to previously decoded frames to refine the motion field, and the refined motion information is employed in the Wyner-Ziv decoding of the next resolution or quality level of the current frame. An in-depth analysis on this topic will be presented in Chapter 3.

2.3.3 Applications of DVC

Besides low-complexity video encoding and multi-terminal data aggregation such as multi-view video coding [41], there are many other interesting applications using DVC/DSC. Just to name a few: since DVC encodes a frame itself instead of prediction residual, it can be used to provide better error resiliency over conventional hybrid video coding [12][120][89]; for the similar reason, in SP frame switching [51], one can encode the primary SP frames using WZVC [23][42], without the necessity to encode any secondary SP frames; in addition to improve the coding efficiency of DSC/DVC, people also start to consider power-efficient communications over WSNs [27], including our work [68] to be presented in Chapter 5; there are also security-related applications of

DVC, e.g. tempering proof [60], secure collaboration [44] and compression of encrypted data [50]. Our work on compression of encrypted images and videos [73][121] is going to be presented in Chapter 4.

Chapter 3

Wyner-Ziv Video Coding with Multi-Resolution Motion Refinement

3.1 Introduction

In Chapter 2 we have mentioned that the major performance loss in WZVC is the video coding loss, which is due to the inaccurate motion estimation (ME) at the decoder side. One significant difference between decoder-side ME and encoder-side ME is that the decoder does not have access to the current frame. This hurts the accuracy of the estimated motion vectors (MV), and consequently, more bits (syndromes) are needed to reconstruct the current frame.

Because motion is a highly non-stationary signal, both spatially and temporally, it is always helpful to have a partial observation of the current frame in ME. As reviewed in Chapter 2, all existing approaches on improving decoder-side ME rely on partial access to current frames in different ways. However, most of them are somewhat ad hoc. An in-depth understanding of their theoretical performance is warranted. In this chapter, we will focus on the multi-resolution motion refinement (MRMR) approach, where low-resolution version of the current frame is iteratively decoded and based on which refined motion information is learned. Our theoretical analysis shows that MRMR significantly outperforms motion extrapolation, and falls only about 1.5 dB behind conventional ME.

On the other hand, it is noted that unlike encoder-side ME, decoder-side ME does not suffer from the overhead in transmitting the motion information. Hence, a natural question to ask is: can we improve the side information (SI) quality of MRMR by providing a more detailed description of the motion field? Conventionally, if a block-matching algorithm (BMA) is used for ME, greater details of the motion can be provided by fractional-pel motion search, using smaller block sizes, or using multiple MVs for one block. In this chapter, we also study the performance of MRMR with these advanced ME techniques integrated.

The rest of the chapter is organized as follows. Section 3.2 provides a review on related works on the efficiency of MCP coding and the efficiency of WZVC using motion extrapolation. In Section 3.3, the efficiency of WZVC with MRMR is modeled and compared with conventional inter-frame coding and WZVC with motion extrapolation. In Section 3.4, we provide analysis on the performance of MRMR when combined with advanced ME techniques. Section 3.5 presents a wavelet-domain codec of WZVC with MRMR. Section 3.6 concludes the chapter.

3.2 Related Works

In this section, we first review some results in the rate-distortion theory [18], then introduce the important model proposed by Girod [37][38][39] for the efficiency of motion-compensated prediction, and end up with the use of this model in analyzing the efficiency of decoder-side motion extrapolation [59].

3.2.1 Rate saving of predictive coding

According to the rate-distortion (R-D) theory, if *s* is a 2-D colored signal, which is zero-mean Gaussian and wide-sense stationary, then for a given distortion constraint

$$D(\theta)=\frac{1}{4\pi^2}\iint \min\left[\theta,\Phi_{ss}(\omega)\right]d\omega \tag{3.1}$$

the minimum coding rate for *s* is

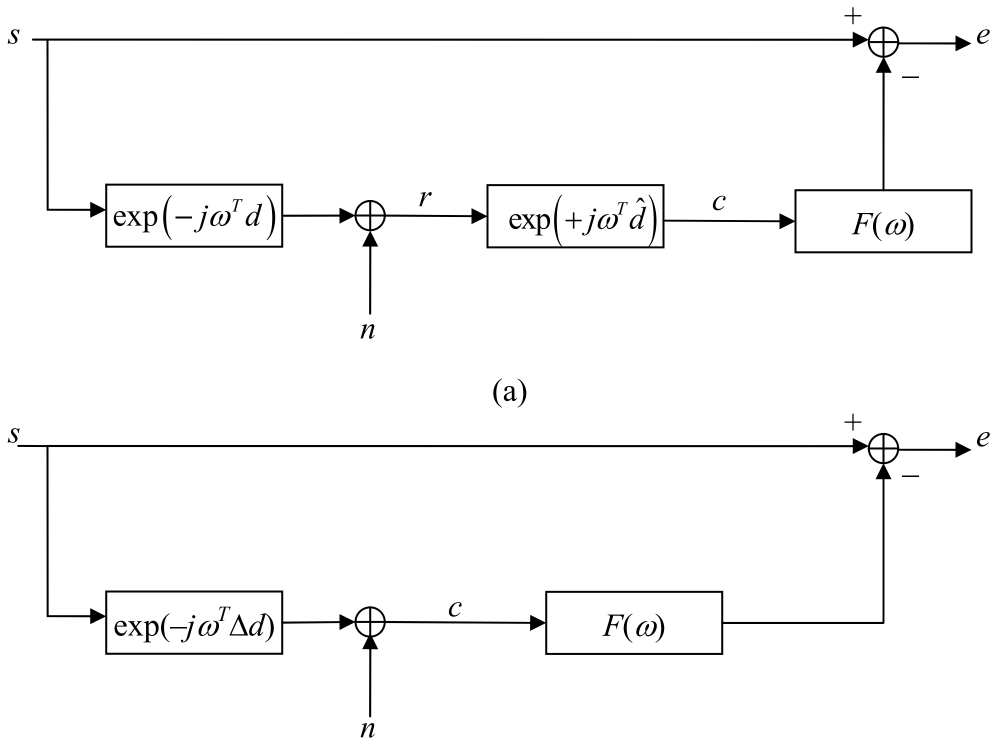
$$R(\theta)=\frac{1}{4\pi^2}\iint \max\left[0,\frac{1}{2}\log_2\frac{\Phi_{ss}(\omega)}{\theta}\right]d\omega \tag{3.2}$$

where Φ_{ss} is the power spectrum density (PSD) of *s*, $\omega = (\omega_x, \omega_y)^T$ is the spatial frequencies in radius (superscript ^{T} denotes the transpose), θ is an arbitrary positive number that generates the R-D bound.

In predictive coding, we encode the prediction residual *e* instead of the original signal. The R-D bound in this case is obtained by replacing Φ_{ss} with Φ_{ee} (the PSD of the residual) in (3.1) and (3.2), respectively. If θ is small, the overall distortion will almost keep unchanged, while the necessary rate will be approximately reduced by

$$\Delta R \approx \frac{1}{8\pi^2} \iint \log_2 \frac{\Phi_{ee}(\omega)}{\Phi_{ss}(\omega)} d\omega \,. \tag{3.3}$$

3.2.2 Efficiency of motion-compensated prediction



(b)

Figure 3.1. (a) Illustration of a typical motion-compensated prediction loop and (b) a simplified version.

Now we consider motion compensated prediction (MCP). In Figure 3.1(a) we illustrate a typical MCP loop. We assume the reference frame *r* is a shifted and noisy version of the current frame, where $d = (d_x, d_y)^T$ is the “true motion” between *s* and *r*, noise *n* is introduced during imaging and the encoding of *r*. In motion compensation, we first perform ME to get an estimate $\hat{d}=\left(\hat{d}_x,\hat{d}_y\right)^T$ of the true motion, and shift the reference frame "back" to get the MCP signal c. The MCP signal is then filtered by an in-loop filter $F(\omega)$ to produce the prediction residual.

A simplified version of the MCP loop is illustrated in Figure 3.1(b). Compared to Figure 3.1(a), we have switched the noise adding module and the motion compensation module,[2] and combined the two shifting operations together, with

$$\Delta d = d - \hat{d} \tag{3.4}$$

denoting the error between the estimated motion and the true motion.

Girod [37] proves that for the MCP loop illustrated in Figure 3.1(b), the following relationship holds:

$$\Phi_{ee}\left(\omega\right)=\Phi_{ss}\left(\omega\right)\left\{1-2\operatorname{Re}\left[F\left(\omega\right)P\left(\omega\right)\right]+\left|F\left(\omega\right)\right|^2\right\}+\Phi_{nn}\left(\omega\right)\left|F\left(\omega\right)\right|^2 \tag{3.5}$$

where Φ_{nn} represents the PSD of the noise, $P(\omega)$ is the characteristic function of Δd, or the Fourier transform of the probability density function (p.d.f.) $p(\Delta d)$, and Re[·] denotes the real part of a complex number.

Substitute (3.5) into (3.3) and suppose $\Phi_{nn} << \Phi_{ss}$, we get the rate saving performance of MCP:

$$\Delta R \approx \frac{1}{8\pi^2}\iint \log_2\left\{1-2\operatorname{Re}\left[F\left(\omega\right)P\left(\omega\right)\right]+\left|F\left(\omega\right)\right|^2\right\}d\omega\,. \tag{3.6}$$

To maximize the rate saving in (3.6), the in-loop filter can be selected to be the Wiener filter:

$$F\left(\omega\right)=P^*\left(\omega\right) \tag{3.7}$$

[2] After the switching, the PSD of the noise does not change.

where the asterisk denotes the conjugate of a complex number. Throughout this chapter, we always assume the in-loop filter takes the form in (3.7). In this case, (3.6) can be simplified as

$$\Delta R = \frac{1}{8\pi^2} \iint \log_2 \left[1 - \left| P(\omega) \right|^2 \right] d\omega . \tag{3.8}$$

In reality, the rate saving performance of MCP depends more on the variance of Δ*d* rather than the exact shape of *p*(Δ*d*). In the literature, a zero-mean isotropic Gaussian distribution of Δ*d* is usually assumed:

$$p(\Delta d) = \frac{1}{2\pi\sigma_{\Delta d}^2} \exp\left(-\frac{\Delta d^T \Delta d}{2\sigma_{\Delta d}^2} \right). \tag{3.9}$$

Thus (3.8) can be further written as

$$\Delta R = \frac{1}{8\pi^2} \iint \log_2 \left[1 - \exp\left(-\omega^T \omega \sigma_{\Delta d}^2 \right) \right] d\omega . \tag{3.10}$$

The fundamental equation in (3.10) suggests that the coding gain of MCP depends exclusively on $\sigma_{\Delta d}^2$, the accuracy of motion estimation. This conclusion applies to both inter-frame video coding and WZVC.

It should be notified that this model, the data rate to encode the MVs has been ignored. The early video coding standards sample the motion field sparsely (e.g., in MPEG-1, only one or two MVs are transmitted for a 16x16 macroblock), thus the overall data rate spent on motion is small. However, in recent video coding standards such as H.264/AVC, with the use of smaller block sizes, multiple-frame motion compensation and fractional-pel motion representation, the MVs occupy a significantly larger portion of the entire bit-stream. In this case, (3.10) becomes less accurate. Moreover, this model only considers object translation and neglects “any other effects like rotation, zoom, covered or uncovered background, illumination changes, etc” [37], which are usually encountered in real-world video coding.

Despite the potential weaknesses, (3.10) captures the essence of MCP with mathematical conciseness, and has proved to be a powerful tool in designing video coding schemes. In [38] and [39], Girod extends the rate-distortion analysis to fractional-pel motion search and multiple-hypothesis prediction (MHP), respectively. In [107], Wiegand et al. design long-term memory MCP and in [32], Flierl et al. propose rate-constrained MHP based on Girod’s theoretical contributions. The model is also used in analyzing the efficiency of scalable video coding [25][86], multiple description video coding and leaky prediction [74], and switching P frames [95]. In [59], Li et al. employ this model to analyze the efficiency of WZVC with motion extrapolation, which will be reviewed in the next subsection.

3.2.3 Efficiency of WZVC with motion-extrapolation

Li et al. [59] exploit an autoregressive model and measure the accuracy of temporal domain motion extrapolation as

$$\sigma^2_{\Delta d-ext} = \left(1-\rho_t^2\right)\sigma_d^2 \tag{3.11}$$

where ρ_t is the correlation between the motion fields in adjacent frames, and σ_d^2 is the variance of the MVs. Substituting (3.11) into (3.10) we get the rate saving of WZVC using motion extrapolated side estimation:

$$\Delta R_{ext} = \frac{1}{8\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}\log_2\left[1-\exp\left(-\omega^T\omega\left(1-\rho_t^2\right)\sigma_d^2\right)\right]d\omega\,. \tag{3.12}$$

The temporal motion correlation ρ_t is estimated in [59] for several QCIF sequences at 30 frames per second (fps), of which the value ranges from 0.31 to 0.85. The results in [59] show that the rate-saving performance of WZVC with motion extrapolation drops quickly as ρ_t decreases. For sequences with low ρ_t, WZVC with motion extrapolation could fall 6 dB or more behind conventional inter-frame coding.

It is also worth noting that (3.11) assumes that the decoder knows the true motion field between the previous two reconstructed frames. In fact, this assumption is not true, because additional motion error will be introduced due to the inefficiency of practical ME algorithms such as block matching (as we will show in the next section). So (3.12) can be seen as an upper bound of the performance of WZVC with motion extrapolation.

3.3 Performance Analysis of Multi-Resolution Motion Refinement

To better use Eq. (3.10), the measure of motion error $\sigma_{\Delta d}^2$ needs to be carefully modeled. A lot of works in the literature simply model $\sigma_{\Delta d}^2$ as a function of the pel-precision in the motion search with

$$\sigma_{\Delta d}^2 = \frac{q^2}{12} \tag{3.13}$$

where q denotes the pel-precision, with q = 1 for integer-pel motion search, q = 1/2 for half-pel motion search, etc.. This model is not very realistic in the sense that in (3.13), $\sigma_{\Delta d}^2$ will approach zero with very fine pel-precision; when it is substituted into (3.10), it turns out that we can achieve arbitrary rate saving by employing fine fractional-pel motion search. However, it is usually not the case in our coding practices. A more accurate model is needed for practical ME methods.

In this section, we consider the motion accuracy of block-matching algorithms (BMA), because they are the most widely-used in today's video codecs. We have built a concise model for estimating the accuracy of decoder-side ME based on Buschmann's work [22]. We also apply this model to analyze the efficiency of the proposed MRMR algorithm, and show its superior performance over motion extrapolation, as well as the gap to the conventional inter-frame ME.

3.3.1 Motion accuracy of block-matching algorithms

Buschmann [22] proposes an excellent model in estimating the accuracy for block-matching based motion search. In this model, a BMA can be described analytically by a series of data processing applied to the true motion field, namely estimation filtering, sampling, quantization and reconstruction filtering. The estimation filtering is introduced because a local neighborhood is usually used for the matching (which is similar to an averaging function). The larger the block size is, the smaller the bandwidth of the low-pass filter is. Sampling and quantization operations account for the fact that the estimated motion field has limited spatial resolution and amplitude precision. The sampling rate is also determined by the block size, and the quantization depends on whether or not fractional-pel accuracy motion search is used. The reconstruction filter is also a low-pass filter, representing the spatial interpolation (e.g. nearest-neighbor interpolation) of the estimated motion field.

In this case, the noise introduced to the original motion field consists of three parts: the low-pass filtering noise, the aliasing noise due to sampling and the quantization noise. For a decoder-side BMA, without the necessity to consider the overhead in transmitting the MVs, the motion field can be very densely sampled (e.g. using overlapped motion compensation) and finely quantized (e.g. using fractional-pel accuracy search), such that *the only operation that contribute to the motion displacement is the low-pass estimation filtering*.

Now we estimate the low-pass filtering noise in a BMA. Similar as in [22], we characterize the autocorrelation function of a true motion field d as isotropic and exponentially-decreasing:

$$R_{dd}\left(\Delta x, \Delta y\right) = \sigma_d^2 \exp\left(-\omega_0 \sqrt{\Delta x^2 + \Delta y^2}\right) \tag{3.14}$$

where ω_0 is a small constant, Δx and Δy are the difference in the coordinates for any two pixels.

The corresponding PSD of the motion field is written as

$$\Phi_{dd}(\omega) = \frac{2\pi\omega_0\sigma_d^2}{\left(\omega_0^2 + \omega_x^2 + \omega_y^2\right)^{3/2}}. \tag{3.15}$$

If we model the estimation filter as an ideal low-pass one, with the frequency response being

$$H_{lp}(\omega)=\begin{cases}0 & \left(\omega\in\Lambda\left(\omega_b\right)\right)\\ 1 & \left(otherwise\right)\end{cases} \tag{3.16}$$

where ω_b is the cut-off frequency of the estimation filter, and $\Lambda(\omega)$ denotes the frequency band

$$\Lambda(\omega)=\left\{\left(\omega_x,\omega_y\right):\max\left(\left|\omega_x\right|,\left|\omega_y\right|\right)>\omega\right\}, \tag{3.17}$$

then the PSD of the low-pass filtering noise is

$$\Phi_{\Delta d\Delta d}(\omega)=\Phi_{dd}(\omega)\left|1-H_{lp}\left(\omega\right)\right|^2=\begin{cases}\Phi_{dd}(\omega) & \left(\omega\in\Lambda\left(\omega_b\right)\right)\\ 0 & \left(otherwise\right)\end{cases}. \tag{3.18}$$

According to the Parseval's relation,

$$\begin{aligned}\sigma^2_{\Delta d-lp}&=\frac{1}{4\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}\Phi_{\Delta d\Delta d}(\omega)d\omega=\frac{1}{4\pi^2}\iint_{\Lambda(\omega_b)}\Phi_{dd}(\omega)d\omega\\ &\approx\frac{1}{4\pi^2}\iint_{\Lambda(\omega_b)}\frac{2\pi\omega_0\sigma_d^2}{\left(\omega_x^2+\omega_y^2\right)^{3/2}}d\omega=\frac{2\sqrt{2}\omega_0}{\pi\omega_b}\sigma_d^2\end{aligned} \tag{3.19}$$

where the approximation is made because for relatively small block sizes, the cut-off frequency ω_b for the estimation filter is much greater than ω_0, hence in the frequency band $\Lambda(\omega_b)$, we have $\max(|\omega_x|, |\omega_y|) >> \omega_0$, and correspondingly, $\omega_x^2+\omega_y^2>>\omega_0^2$.

On the other hand, according to the autocorrelation function in (3.14), the spatial motion correlation between two neighboring pixels is

$$\rho_s=\exp(-\omega_0). \tag{3.20}$$

So we can replace ω_0 with $\ln\rho_s^{-1}$ in (3.19). An estimate of ρ_s is given in [22] that $\rho_s\approx 0.983$ for CIF sequences. As for the cut-off frequency ω_b, we can assume $\omega_b=\pi/B$, where

B is the 1-D block size used for matching. Hence (3.19) can be rewritten as

$$\sigma^2_{\Delta d-lp} = \frac{2\sqrt{2}B\ln\rho_s^{-1}}{\pi^2}\sigma_d^2 = kB\sigma_d^2 \tag{3.21}$$

where $k \approx 0.005$ for CIF sequences, and $k \approx 0.01$ for QCIF sequences.

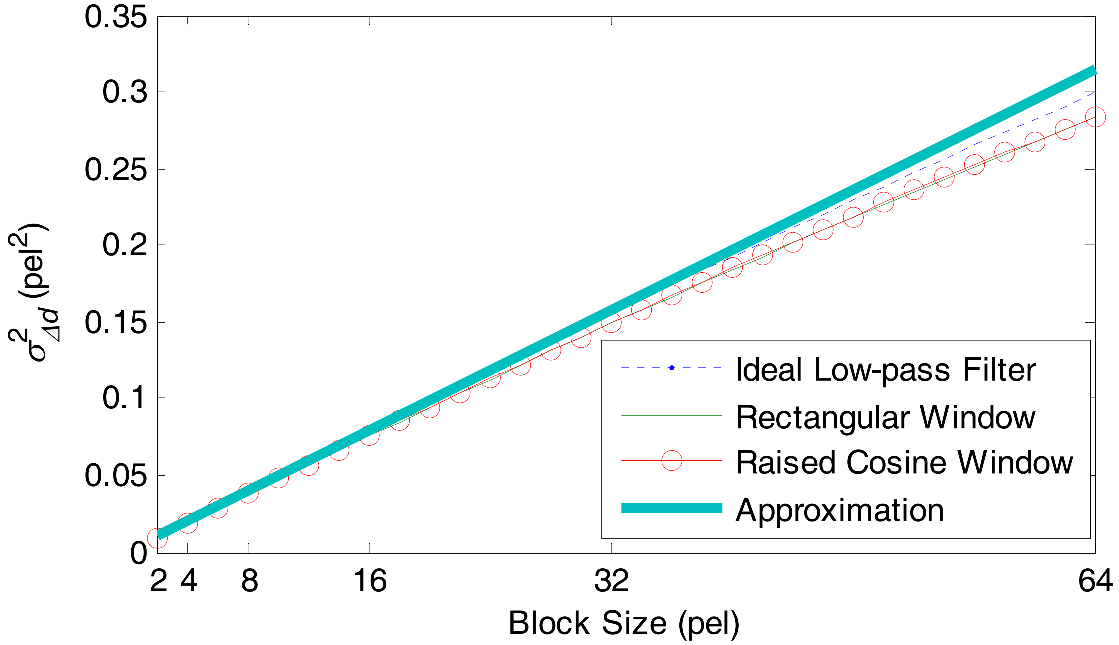


Figure 3.2. Low-pass filtering noise introduced by a BMA with different block sizes. Here we can see (3.21) is a good approximation, especially for smaller block sizes.

It might be an oversimplification to model the estimation filter as ideally low-pass. In fact, we also tested the numerical results when $H_{lp}(\omega)$ is modeled as the Fourier transform of the rectangular window and the raised cosine window. The corresponding curves, together with the ideal low-pass filtering case, are plotted in Figure 3.2. We can see that in either case, $\sigma^2_{\Delta d-lp}$ is nearly linear for a large range of block sizes, but with a slightly different slope. Since for the rectangular window and the raised cosine window,

$\sigma^2_{\Delta d-lp}$ cannot be expressed in a closed form, we will consider (3.21) as a good approximation.

Now we have modeled the motion accuracy of decoder-side BMA as a linear function of the block size. As for encoder-side BMA, there are additional types of motion error (aliasing noise, quantization noise, etc.). However, we can still use (3.21) as a lower bound of the motion error. In this case, the rate saving performance of inter-frame ME is written as

$$\Delta R_{inter} = \frac{1}{8\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}\log_2\left[1-\exp\left(-\omega^T\omega\times kB\sigma_d^2\right)\right]d\omega\,. \tag{3.22}$$

We will use (3.22) for inter-frame ME throughout this section if not specifically mentioned otherwise.

The motion accuracy model we have derived in (3.21) is more realistic than the one in (3.13). For example, in our coding practices, it has been proved that reducing the block size is an efficient way to improve the coding efficiency. The state-of-the-art video coding standard H.264/AVC allows the minimum block size of 4x4. On the other hand, for some sequences with very dynamic scenes, using fractional-pel motion search does not provide much gain. This is because the low-pass filtering noise is large due to the high motion intensity σ_d^2 in (3.21).

3.3.2 Efficiency of decoder-side MRMR

Now we are in the position to derive the efficiency of MRMR. Let's consider the following coding paradigm of WZVC. Suppose the frames are infinitely large and are sampled at the integer grid. The discrete time Fourier transform (DTFT) of the frame creates a periodic frequency spectrum. Due to the periodicity, we only have to transmit the spectrum in one period $|\omega_x| < \pi$, $|\omega_y| < \pi$.

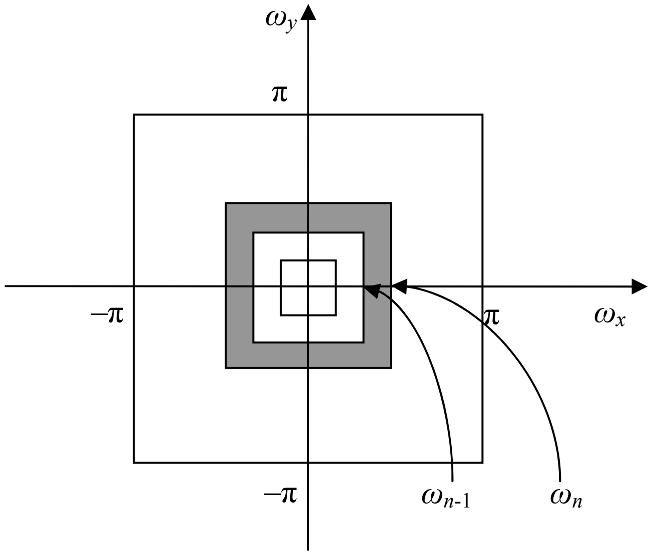


Figure 3.3. Illustration of the filter bank employed in MRMR. The shaded area denotes the frequency band of the n^{th} ideal band-pass filter.

Let $0 = \omega_0 < \omega_1 < \ldots < \omega_N = \pi$ be a series of frequency points, and we construct N ideal band-pass filters based on them, so that the n^{th} filter encompasses the frequency band $\{\Lambda(\omega_{n-1})\backslash\Lambda(\omega_n)\}$ for $n = 1, \ldots, N$, where the definition of $\Lambda(\omega)$ is found in (3.17), and "\" denotes the set difference. An illustration of the filter bank is shown in Figure 3.3.

To encode the current frame, we apply the N filters to the frame and get N band-pass images without overlapping with each other in the frequency domain. The rate to transmit the whole image equals the total rate to transmit the N band-pass images.

At the decoder side, suppose at time n, the first n band-pass images have been received, based on which a low-pass version of the current frame is reconstructed. We assume this partially reconstructed image is critically sampled according to the Nyquist sampling theorem. Thus at a certain time instance, a *low-resolution* version of the current frame is available at the decoder. Now the decoder can use this low-resolution frame to get a refined estimation of the motion field and do the MCP for the next higher band-pass image.[3] We are interested in how much we can benefit from this MRMR approach.

We assume the decoder employs a BMA to refine the motion field. The BMA is carried out using the same parameters (block size, pel-precision, etc.) at different resolution levels. To estimate the efficiency of MRMR, it is necessary to understand the motion search accuracy of a BMA, when there is only a low-resolution version of the current frame available. It is analyzed as follows.

At time n, since the currently-available frequency band is $|\omega_x|<\omega_n$, $|\omega_y|<\omega_n$, the partially reconstructed image can be critically sampled at the rate of ω_n/π. If the decoder applies a BMA with block size B to the low-resolution image, the "effective block size" is π/ω_n times of that in the full-resolution image space (or the bandwidth of the estimation filter is ω_n/π times of the original). Hence we can replace B with $\pi B/\omega_n$ in (3.21) and get the motion search accuracy in the $(n+1)^{\text{th}}$ level in MRMR as

$$\sigma^2_{\Delta d-MRMR(n+1)} = \frac{\pi k B \sigma_d^2}{\omega_n}. \tag{3.23}$$

[3] There is an implicit assumption here that all of the band-pass images share the same motion field

In an ideal case, if the number of resolution levels N approaches infinite and for arbitrary n, ω_n and ω_{n+1} are close enough, for any frequency point $\omega = (\omega_x, \omega_y)^T$ in the $(n+1)^{\text{th}}$ subband, we have $\omega_n \approx \max(|\omega_x|, |\omega_y|)$. Then the motion accuracy in (3.23) can be replaced with:

$$\sigma^2_{\Delta d-MRMR}(\omega) = \frac{\pi k B \sigma_d^2}{\max\left(|\omega_x|, |\omega_y|\right)}. \tag{3.24}$$

Note that unlike the cases in inter-frame ME or motion extrapolation, the motion accuracy in WZVC with MRMR is a function of ω. This can be interpreted as when higher resolution image is used for motion search, the estimated motion is also more accurate. Thus the total rate saving achieved by using MRMR is

$$\Delta R_{MRMR} = \frac{1}{8\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}\log_2\left[1-\exp\left(-\omega^T\omega\times\frac{\pi k B \sigma_d^2}{\max\left(|\omega_x|, |\omega_y|\right)}\right)\right]d\omega. \tag{3.25}$$

3.3.3 Performance comparison

In this subsection, performance will be compared among inter-frame ME, motion extrapolation and MRMR. From the variance of MV displacement in (3.11), (3.21) and (3.24), we can see that all of them are essentially content dependent. The difference is whether the rate saving will depend on spatial motion correlation ρ_s or the temporal motion correlation ρ_t. In general, one can expect better prediction performance from motion extrapolation if the temporal motion correlation is high. However, many factors may result in the reduction of ρ_t: scene change, frame dropping, lens vibration, object moving in / out, and object moving with acceleration. As for the spatial motion correlation, except for the pixel pairs that belong to different moving objects, ρ_s is often more stable and higher, which is good for improving the coding efficiency and for rate control. It is worth noting that typically when the spatial resolution of the sequence gets higher, ρ_s also gets higher.

A. Equivalent transfer function

Eq. (3.5) shows that the effect of MCP can be considered as applying an equivalent spatial transfer function [37] to the original frame. When the in-loop filter is a Wiener filter as in (3.7), the transfer function can be written as

$$H_{MCP}(\omega) = \sqrt{1 - \left|P(\omega)\right|^2} = \sqrt{1 - \exp\left(-\omega^T \omega \times \sigma_{\Delta d}^2\right)}. \tag{3.26}$$

Note that for MRMR, $\sigma_{\Delta d}^2$ is replaced by $\sigma_{\Delta d}^2(\omega)$. A smaller amplitude of $H_{MCP}(\omega)$ means the prediction of the current frame is better because there is less energy remaining in the prediction residual.

We assume B = 4, $\sigma_d^2 = 1$, k = 0.01 and let ρ_t take values from {0.3, 0.6, 0.9}. The frequency responses of the corresponding transfer functions are plotted in 1-D in Figure 3.4. We can see that inter-frame ME always provides the best prediction. MRMR falls behind motion extrapolation in low frequency bands due to the insufficient available information. However, as more and more information is available at the decoder, the performance of MRMR will approach that of the inter-frame ME. This means that in WZVC with MRMR, the coding efficiency of the high frequency components will be closer to inter-frame coding than the low frequency components. This is an attractive property in coding of 2-D signals, because high frequency coefficients considerably outnumber low frequency coefficients.

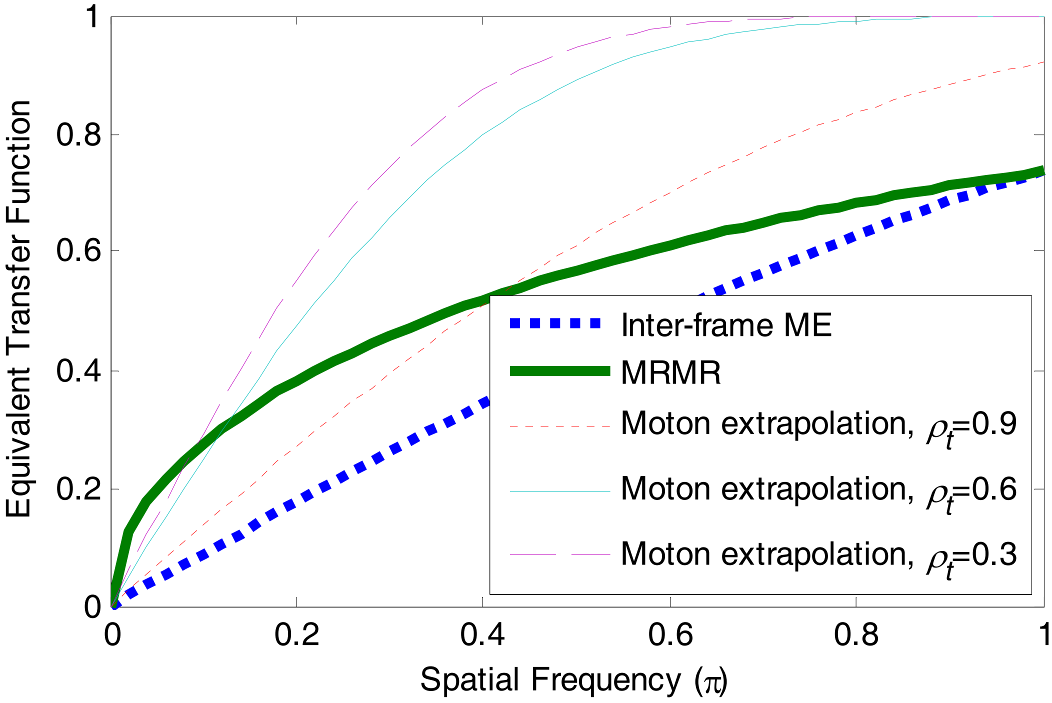


Figure 3.4. 1-D equivalent transfer functions for inter-frame ME, MRMR and motion extrapolation.

B. Critical frequency

It is also observed that for each motion extrapolation curve, there is a "critical frequency" at which it intersects with the curve for MRMR. This critical frequency denotes the time when there is enough information available at the decoder such that refinement of the motion field is worthwhile. A “smart” decoder should be able to switch the prediction mode between MRMR and motion extrapolation based on ω^*. The critical frequency can be calculated from (3.11) and (3.24) as

$$\omega^* = \frac{kB}{1-\rho_t^2}\pi \,. \tag{3.27}$$

The critical frequency ω^* is less than π/4 even when ρ_t is as high as 0.9, meaning that it is usually safe to perform motion refinement when a quarter-resolution image (of QCIF) is decoded. Note that ω^* drops quickly as ρ_t decreases. For sequences with medium or low ρ_t, switching between MRMR and motion extrapolation seems not necessary, since motion extrapolation does not provide significant gain at low-frequency bands.

C. Rate saving performance

Next, comparison will be made on the rate saving performance (over intra-frame coding) of the three approaches. We still assume B = 4, k = 0.01 and let ρ_t take values from {0.3, 0.6, 0.9}. Numerical results are generated for ΔR_{inter}, ΔR_{MRMR} and ΔR_{ext}, and the corresponding curves are plotted in Figure 3.5. A curve with a lower position means better rate saving performance.

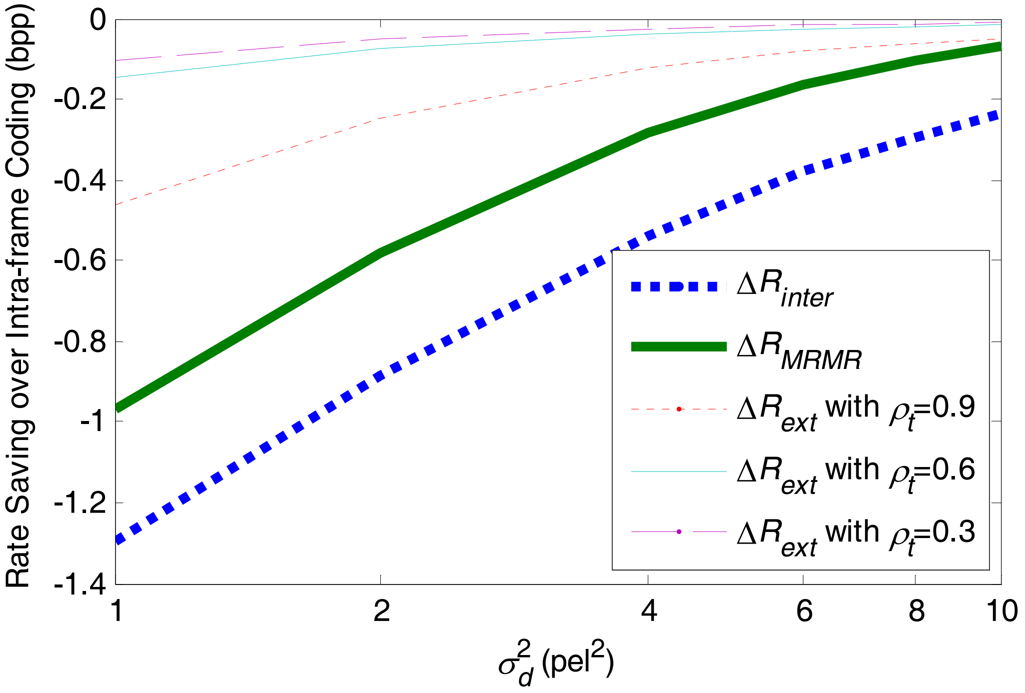


Figure 3.5. Comparison of the rate saving performance among inter-frame ME, motion extrapolation and MRMR.

From Figure 3.5 we can see, for any approach among the three, the rate saved over intra-frame coding is more significant for more static video sequences with smaller σ_d^2. When σ_d^2 varies from 10 to 1, WZVC with MRMR can save 0.07 to 0.97 bits per pixel (bpp) over intra-frame coding. When compared with motion extrapolation, MRMR shows significant improvement. Even when ρ_t is as high as 0.9, WZVC with MRMR can save 0.02 to 0.51 bpp more than WZVC with motion extrapolation. It should be noted that ΔR_{ext} drops quickly as ρ_t decreases. For sequences with medium or low ρ_t, the coding gain of motion extrapolation is marginal. When σ_d^2 is as low as 1, a maximum possible saving of 0.83 bpp is observed. It is well known that for high rate coding, the rate difference at 1 bpp can be translated into 6.02 dB PSNR difference. So it can be concluded that WZVC with MRMR can outperform WZVC with motion extrapolation by up to 5 dB.

The comparison between inter-frame coding and WZVC with MRMR shows an almost constant gap, which is averaged at 0.25 bpp when σ_d^2 varies from1 to 10. The corresponding performance gap in terms of PSNR is 1.5 dB.

3.4 MRMR with Extensive Motion Exploration

In the previous section we have shown that MRMR significant outperforms motion extrapolation, but falls 1.5 dB behind inter-frame ME. We are interested in if it is possible to make up with the 1.5 dB gap.

The analysis in the previous section is based on the assumption that both inter-frame ME and MRMR use the same settings in the BMA. However, since MRMR is performed at the decoder side, it has the advantage that the motion information does not have to be transmitted. Without this overhead, it is possible to exploit more advanced ME techniques to extensively explore the dependency between the current frame and the reference frame(s). In the literature, better MCP can be achieved through finer fractional-pel motion search [38], using smaller block sizes or multiple hypothesis prediction [39][32]. In this section, we will analyze the performance of MRMR with these techniques.

3.4.1 MRMR with fractional-pel motion search

BMAs introduce quantization noise to the true motion field because the reference frame(s) is sampled on a discrete grid. In (3.21) and (3.24), the quantization noise is not considered. In this subsection, we model the quantization error in BMAs as an additive white noise with variance $\sigma^2_{\Delta d-q}$, thus (3.21) and (3.24) are adjusted as

$$\sigma^2_{\Delta d-inter} = kB\sigma_d^2 + \sigma^2_{\Delta d-q} \tag{3.28}$$

and

$$\sigma^2_{\Delta d-MRMR}\left(\omega\right) = \frac{\pi kB\sigma_d^2}{\max\left(\omega_x, \omega_y\right)} + \sigma^2_{\Delta d-q} \tag{3.29}$$

respectively. According to [22], $\sigma^2_{\Delta d-q}$ is derived by applying uniform scalar quantization with the step size q to a random MV with the p.d.f. $p_d(d)$. We use the same settings as in [22], where $p_d(d)$ is assumed to be a generalized Gaussian distribution with the shape factor being 0.3.

Still assume B = 4 and k = 0.01, we plot the rate saving curves in Figure 3.6 for integer-pel, half-pel, quarter-pel accuracy search for both inter-frame ME and MRMR. It can be seen that the rate saving using fractional-pel accuracy search in MRMR is less significant than in inter-frame ME. This can be explained from (3.29) that the impact of low-pass filtering noise is more significant in the MRMR case. We also conclude that in MRMR, it is more effective to perform fractional-pel motion search at the high frequency subbands where the low-pass filtering noise is less dominant.

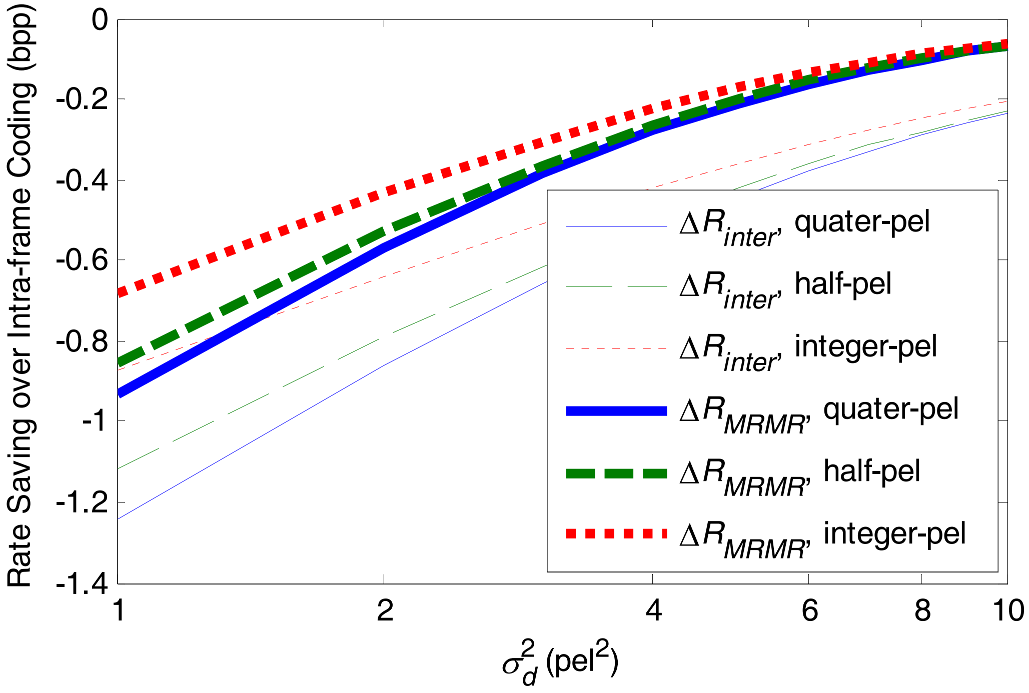


Figure 3.6. Rate saving performance of MRMR and inter-frame ME using different pel-precision search.

It is also observed that when using fractional-pel search in MRMR or inter-frame ME, we can expect more gain on sequences with low motion (smaller σ_d^2). This is explained as follows. $\sigma_{\Delta d-q}^2$ is related to both q and σ_d^2. For larger σ_d^2, $p_d(d)$ becomes flatter and $\sigma_{\Delta d-q}^2$ approaches $q^2/12$, which is a constant and less significant than the low-pass filtering noise. While for a smaller σ_d^2, $p_d(d)$ becomes more impulsive and $\sigma_{\Delta d-q}^2$ will approach σ_d^2, which could be more significant than the low-pass filtering noise. Consequently, we can see the curve for ΔR_{inter} with q = 1 intersects with the curves of ΔR_{MRMR} with q = 0.5 and q = 0.25 when σ_d^2 is around 1.5 and 1, respectively. This means under certain complexity constraints of the encoder, WZVC with MRMR might be able to outperform inter-frame coding.

The results in [59] show that improving the pel-precision does not help much in motion extrapolation. However, it is usually worthwhile to perform fractional-pel motion search in MRMR. The possible rate saving is up to 0.25 bpp if an integer-pel search is substituted by a quarter-pel search, which can be translated into 1.5 dB in PSNR gain. On the other hand, by comparing Figure 3.5 and Figure 3.6 we can see, the extra gain is marginal by employing motion search that is finer than quarter-pel.

3.4.2 MRMR with smaller block sizes

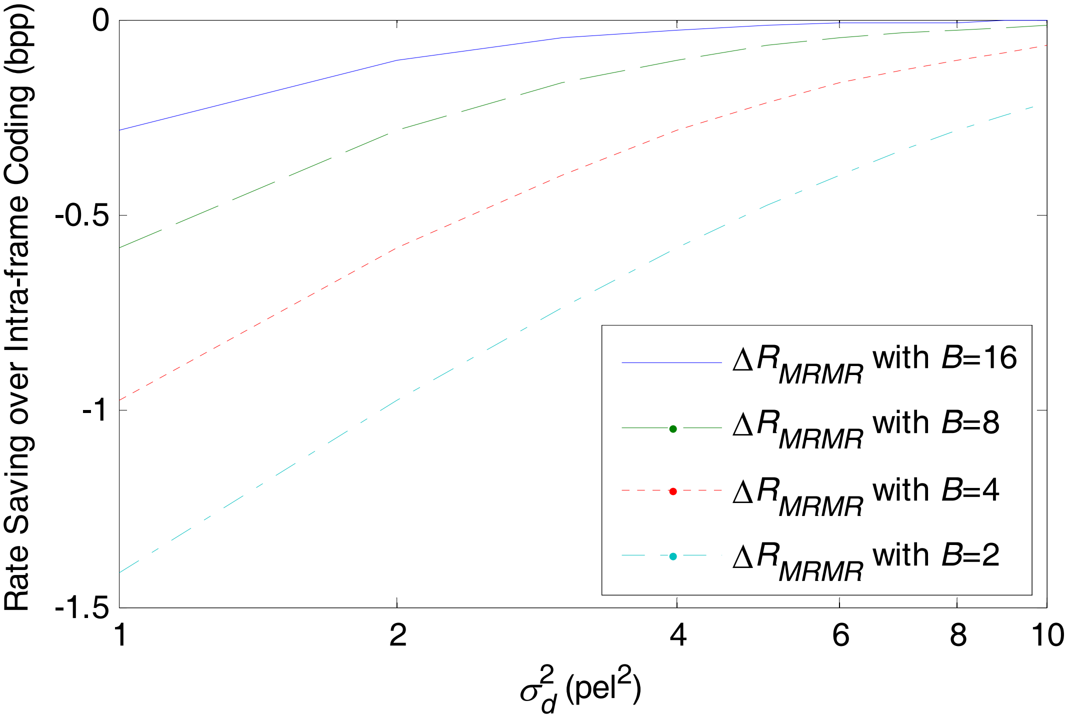


Figure 3.7. Rate saving performance of MRMR with different block sizes.

The motivation to use smaller block sizes in MRMR is natural. By comparing (3.21) and (3.24) we know MRMR is less accurate then inter-frame ME because there is a penalty factor $\frac{\pi}{\max\left(\omega_x,\omega_y\right)}$ in its low-pass filtering noise, and the factor is always greater than 1. To compensate the penalty factor, an efficient way is to use a smaller B in MRMR. In fact, For very accurate ME with small $\sigma_{\Delta d}^2$, using a Taylor series expansion, (3.25) can be approximated as

$$\Delta R \approx \frac{1}{8\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}\log_2\left(\omega^T\omega\sigma_{\Delta d}^2\left(\omega\right)\right)d\omega = C+\frac{1}{2}\log_2 B \tag{3.30}$$

where *C* is independent of *B*. Hence reducing *B* by half means an extra 0.5 bpp saving, or 3 dB gain in PSNR. This encourages the use of smaller block sizes. Certainly this conclusion also applies to encoder-side ME. However, halving the block sizes also means quadrupling the number of MVs, which greatly increases the transmission overhead.

For more practical settings, we assume *k* = 0.01 and substitute different *B* values into (3.25). The corresponding rate-saving curves of MRMR are plotted in Figure 3.7. We can see that reducing the block size in BMA is effective in improving the rate-saving performance of MRMR. The *B* = 2 curve saves 0.26 bpp more than the *B* = 4 curve, which has already made up with the 1.5 dB gap with respect to inter-frame ME.

It is worth noting that block matching with a very small *B* is an ill-posed problem. People usually impose some smoothness constraint to make sure the derived motion field is physically meaningful. This is equivalent to applying additional inter-block low-pass filtering to the motion field, which somewhat limits the gain of reducing the block size.

3.4.3 MRMR with multiple-hypothesis prediction

Another efficient way to improve the prediction performance in BMAs is through multiple-hypothesis prediction (MHP) [39][32]. One typical example is the bi-directional prediction (B pictures), where two motion vectors are assigned to the same block, with one pointing to a previous frame and the other pointing to a future frame. Each of the two motion-compensated blocks is called a hypothesis, and their weighted average is used for the prediction of the current block.

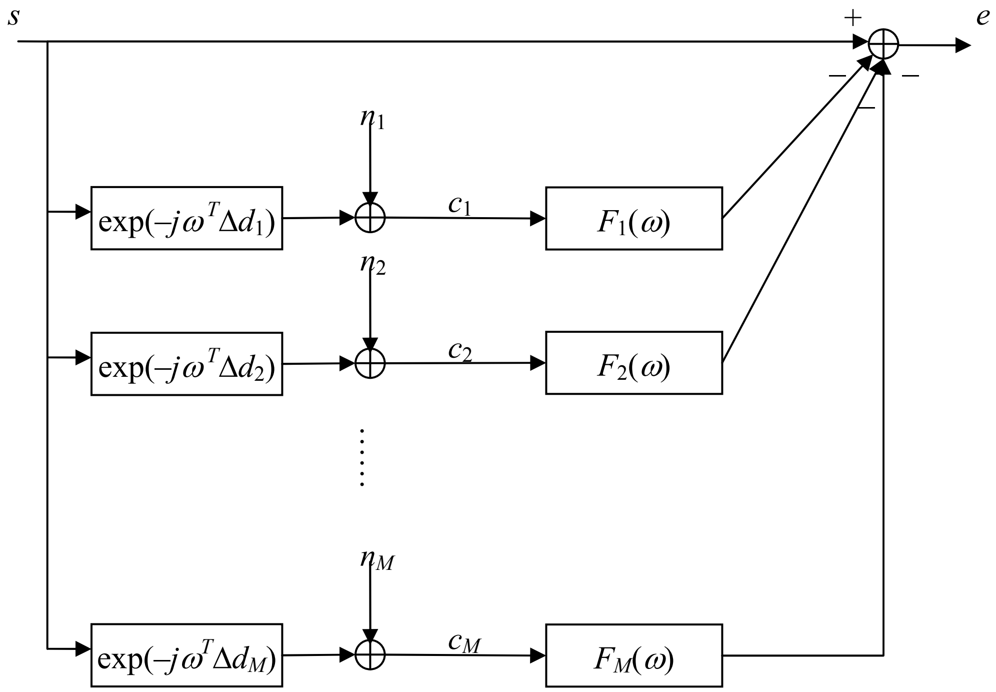


Figure 3.8. Illustration of multiple-hypothesis motion-compensated prediction.

In this chapter, we will not consider B pictures. In stead, the prediction of the current block is the weighted average of M motion-compensated blocks from previously decoded frame(s). It provides several advantages over B-picture coding. First, there is no extra delay in decoding a future picture. Second, more than two MVs can be employed to make the prediction more accurate.

The diagram of multiple-hypothesis motion-compensated prediction is illustrated in Figure 3.8, where there are M hypotheses in total, each hypothesis c_i is a shifted (by the motion error Δd_i) and noisy (by adding n_i) version of the current frame. According to the results in [39], if $\Phi_{n_i n_i} \ll \Phi_{ss}$ for each n_i, and the in-loop filters all take the optimum form, the rate saving using MHP is

$$\Delta R_{MHP} = \frac{1}{8\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \log_2 \left[1 - \mathbf{P}^H E\left(\mathbf{D}\mathbf{D}^H \right)^{-1} \mathbf{P} \right] d\omega \tag{3.31}$$

where

$$\mathbf{D} = \begin{pmatrix} \exp\left(-j\omega^T \Delta d_1\right) \\ \exp\left(-j\omega^T \Delta d_2\right) \\ \vdots \\ \exp\left(-j\omega^T \Delta d_M\right) \end{pmatrix}, \tag{3.32}$$

and

$$\mathbf{P} = E[\mathbf{D}] = \begin{pmatrix} P_1(\omega) \\ P_2(\omega) \\ \vdots \\ P_M(\omega) \end{pmatrix}, \tag{3.33}$$

with $P_i(\omega)$ being the characteristic function of Δd_i, and the superscript ^{H} denoting the transposed conjugate of a complex vector.

Following the same assumptions as in [32], we now let the motion displacements Δd_1, …, Δd_M be jointly Gaussian, each of which is zero-mean and has the same variance

σ_{Δ}^{2}, and the correlation between any two displacements are the same (denoted in ρ_{Δ}).[4] In this case we have

$$\mathbf{P} = P\left(\omega, \sigma_{\Delta}^{2}\right)\mathbf{1} \tag{3.34}$$

and

$$\begin{aligned}
&E\left(\mathbf{D}\mathbf{D}^{H}\right) \\
&= E\begin{bmatrix} \exp\left(-j\omega^{T}\left(\Delta d_{1}-\Delta d_{1}\right)\right) & \exp\left(-j\omega^{T}\left(\Delta d_{1}-\Delta d_{2}\right)\right) & \cdots & \exp\left(-j\omega^{T}\left(\Delta d_{1}-\Delta d_{M}\right)\right) \\ \exp\left(-j\omega^{T}\left(\Delta d_{2}-\Delta d_{1}\right)\right) & \exp\left(-j\omega^{T}\left(\Delta d_{2}-\Delta d_{2}\right)\right) & \cdots & \exp\left(-j\omega^{T}\left(\Delta d_{2}-\Delta d_{M}\right)\right) \\ \vdots & \vdots & \ddots & \vdots \\ \exp\left(-j\omega^{T}\left(\Delta d_{M}-\Delta d_{1}\right)\right) & \exp\left(-j\omega^{T}\left(\Delta d_{M}-\Delta d_{2}\right)\right) & \cdots & \exp\left(-j\omega^{T}\left(\Delta d_{M}-\Delta d_{M}\right)\right) \end{bmatrix} \\
&= \begin{bmatrix} 1 & P\left(\omega, 2(1-\rho_{\Delta})\sigma_{\Delta}^{2}\right) & \cdots & P\left(\omega, 2(1-\rho_{\Delta})\sigma_{\Delta}^{2}\right) \\ P\left(\omega, 2(1-\rho_{\Delta})\sigma_{\Delta}^{2}\right) & 1 & \cdots & P\left(\omega, 2(1-\rho_{\Delta d})\sigma_{\Delta}^{2}\right) \\ \vdots & \vdots & \ddots & \vdots \\ P\left(\omega, 2(1-\rho_{\Delta})\sigma_{\Delta}^{2}\right) & P\left(\omega, 2(1-\rho_{\Delta})\sigma_{\Delta}^{2}\right) & \cdots & 1 \end{bmatrix}
\end{aligned} \tag{3.35}$$

where

$$P\left(\omega, \sigma^{2}\right) = \exp\left(-\frac{\omega^{T}\omega\sigma^{2}}{2}\right) \tag{3.36}$$

is the Fourier transform of a zero-mean Gaussian p.d.f. with variance σ^2, and $\mathbf{1} = [1, \ldots, 1]^T$.

On the other hand, if an $M\times M$ matrix is in the form

$$\mathbf{C} = \begin{bmatrix} a & b & \cdots & b \\ b & a & \cdots & b \\ \vdots & \vdots & \ddots & \vdots \\ b & b & \cdots & a \end{bmatrix}, \tag{3.37}$$

[4] This is true if the N hypotheses are searched using the same method, and there is no preference among them.

then its inverse can be written as

$$\mathbf{C}^{-1} = \frac{1}{(a-b)\left(a-(M-1)b\right)} \begin{bmatrix} a+(M-2)b & -b & \cdots & -b \\ -b & a+(M-2)b & \cdots & -b \\ \vdots & \vdots & \ddots & \vdots \\ -b & -b & \cdots & a+(M-2)b \end{bmatrix}. \qquad (3.38)$$

Substitute *a* with 1, *b* with $P\left(\omega, 2(1-\rho_\Delta)\sigma_\Delta^2\right)$, and insert (3.34) – (3.38) into (3.31) we get

$$\Delta R_{MHP} = \frac{1}{8\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}\log_2\left[1-\frac{M\exp\left(-\omega^T\omega\sigma_\Delta^2\right)}{1+(M-1)\exp\left(-(1-\rho_\Delta)\,\omega^T\omega\sigma_\Delta^2\right)}\right]d\omega. \qquad (3.39)$$

Note that in (3.39), if we set M = 1, same result is obtained as in (3.10). Similar work has been done in [32], under the assumption that the M hypotheses are simply averaged, while (3.39) is the optimum case where the M hypotheses are Wiener filtered. Eq. (3.39) is for inter-frame ME, but we can simply replace σ_Δ^2 with $\sigma_\Delta^2(\omega)$ to extend it to the MRMR case. In the meantime, we would like to point out that using MHP at the encoder side will produce a lot of overhead bits – doubling the number of hypotheses also doubles the number of MVs. Therefore MHP is not widely used in state-of-the-art video coding standards. For example, in H.264/AVC P slices, only one hypothesis is allowed for each inter-predictive block.

For very accurate ME with small σ_Δ^2, (3.39) can be approximated using a Taylor series expansion

$$\Delta R_{MHP} \approx \frac{1}{8\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}\log_2\left[\frac{M-1}{M}\left(\frac{1}{M-1}+\rho_\Delta\right)\omega^T\omega\sigma_\Delta^2\right]d\omega. \qquad (3.40)$$

We can see that for a large number of hypotheses (large M), reducing ρ_Δ is equally important as reducing σ_Δ^2; if the displacements are mutually independent (ρ_Δ=0),[5] doubling the number of hypotheses is equivalent to reducing σ_Δ^2 by half (which means a 3 dB gain in prediction performance); however, if ρ_Δ>0, no significant gain can be obtained by increasing M when M >>1/ρ_Δ.

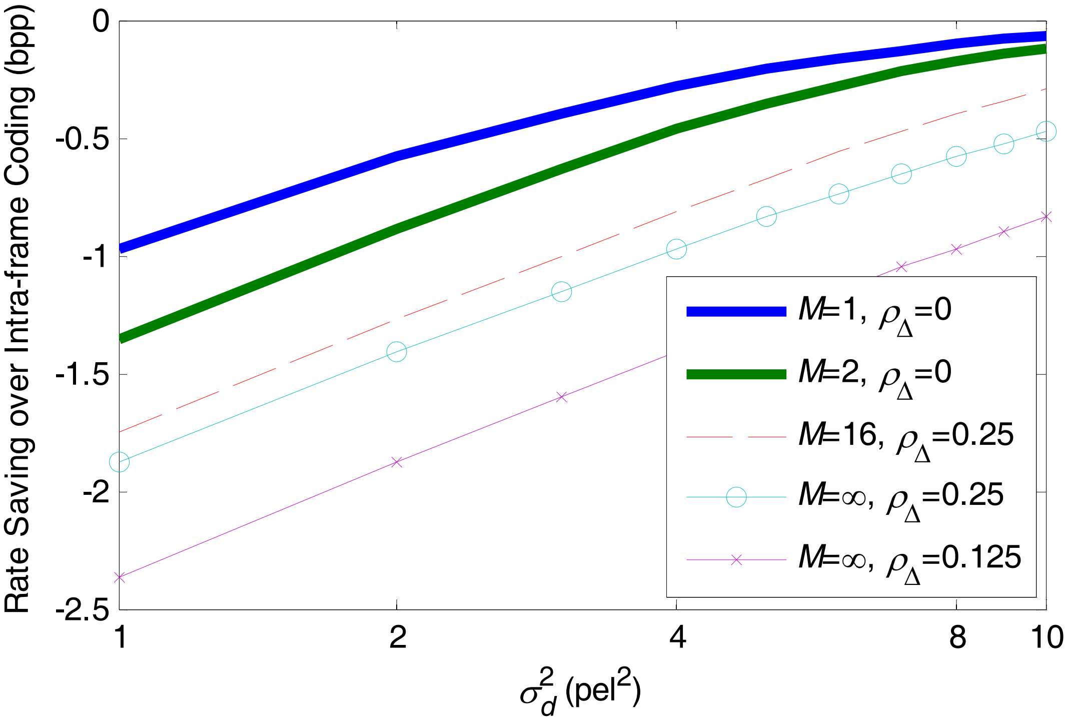


Figure 3.9. Rate saving performance of MRMR with MHP.

Let B=4, k=0.01 and assume there is no quantization noise in the ME. We plot the rate-saving curves for MRMR in Figure 2.10. Our discussions above can be well confirmed.

[5] ρ_Δ can certainly take negative values, but as proved in [32], $\rho_\Delta \geq (1-N)^{-1}$, so here we use 0 as the lower bound of ρ_Δ for large Ns.

3.5 Case Study

From the previous analysis, we have known that MRMR is much better than motion extrapolation, and it is possible to make up the gap between MRMR and inter-frame by extensive motion exploration. In this section, we will integrate the advanced ME techniques into a practical MRMR framework to improve the decoder-side MCP performance, and compare it with motion-extrapolation and inter-frame ME.

3.5.1 System description

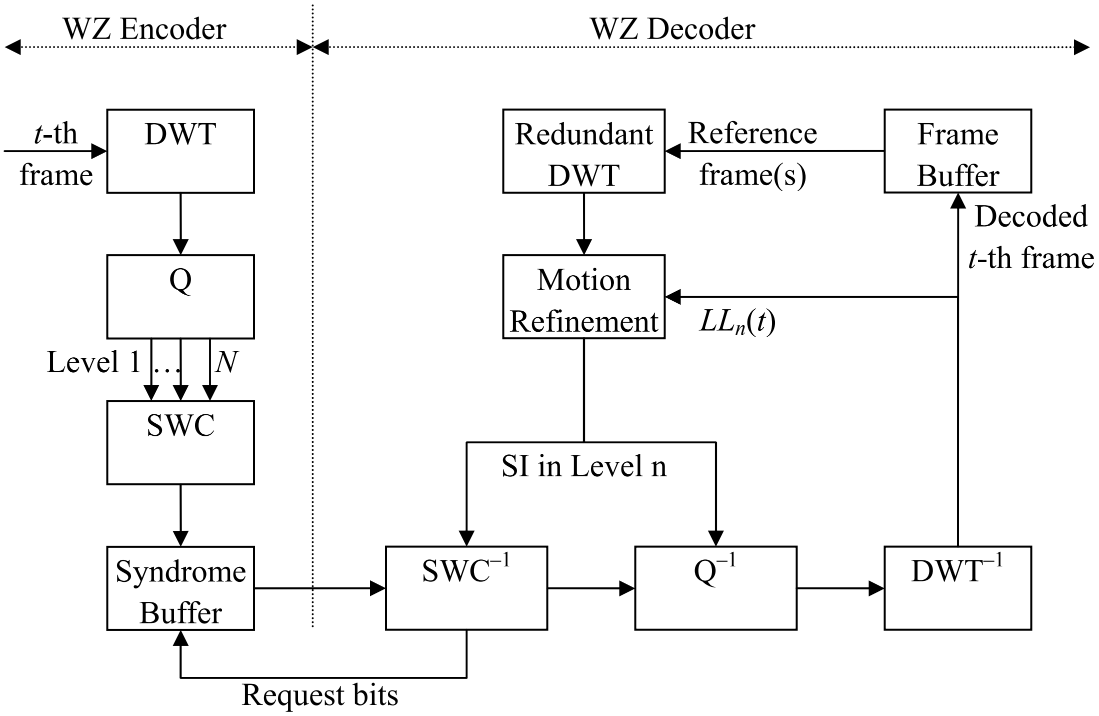


Figure 3.10. Illustration of WZVC with MRMR at the decoder side.

The diagram of a wavelet domain WZVC codec with MRMR is illustrated in Figure 3.10. The encoder applies an N-level wavelet decomposition to each of the frames. For each frame, ($3N$+1) subbands are produced, namely, HL_n, LH_n, HH_n, (n = 1, 2, …, N), and LL_N.[6] For the sake of convenience, let LL_{n-1}denote the reconstructed subband from LL_n, HL_n, LH_n and HH_n, (n = 1, 2, …, N). Each of these subbands is Wyner-Ziv encoded and transmitted to the decoder at the necessary rate. To distinguish the subbands in different frames, we use notations such as $LL_n(t)$, where t is the frame index.

At the decoder side, supposed $F(t–1)$ is available,[7] the decoding process of $F(t)$ can be described as follows:

1) Estimate $LL_N(t)$ from $LL_N(t–1)$ by applying MCP based on an initial motion field. The initial motion field could be the same one as $F(t–1)$, or could be set to zero if it is not available;
2) Reconstruct $LL_N(t)$ by applying Wyner-Ziv decoding;
3) Set $n = N$;
4) Refine the motion field by performing ME between $LL_n(t)$ and $LL_n(t–1)$;
5) Predict $HL_n(t)$, $LH_n(t)$ and $HH_n(t)$ by copying the corresponding motion-compensated coefficients in $HL_n(t–1)$, $LH_n(t–1)$ and $HH_n(t–1)$;
6) Apply Wyner-Ziv decoding to reconstruct $HL_n(t)$, $LH_n(t)$ and $HH_n(t)$;
7) Reconstruct $LL_{n-1}(t)$ based on $LL_n(t)$, $HL_n(t)$, $LH_n(t)$ and $HH_n(t)$;

[6] To accommodate the conventional notation in wavelet decomposition, here LL_N denotes the lowest frequency subband, which is different from what is used in Section 3.3.

[7] Although we only describe the case of using one reference frame, it can be easily extended to multiple-frame prediction.

8) Reduce *n* by 1. If *n* reaches zero, the decoding is finished; otherwise go to step 4).

In the following simulations, WZVC encoder decomposes each frame into 3 levels using the Daubechies 9/7 bi-orthogonal filter bank. Referring to the discussions about the “critical frequency”, 3 levels of decomposition is usually “safe” for most sequences. More decomposition levels only lead to marginal gain. Uniform quantization is employed to quantize the wavelet coefficients.

At the decoder side, to overcome the shift-variance problem in the critically-sampled wavelet domain, $F(t–1)$ needs to be transformed to the over-complete wavelet domain. Therefore, the subbands of $F(t–1)$ are all in an over-complete form, derived from non-subsampled DWT. This approach significantly improves the efficiency of MCP, at the cost of extra memory consumption in the decoder. However this strategy still fits the “simple-encoding, complex-decoding” principle of WZVC.

It should be noted that although iterative motion refinement is employed in our scheme, the computational complexity is even lighter than motion extrapolation, if the same BMA is used in both approaches. For example, when the motion refinement is carried out based on a half-resolution image, the number of MVs for estimation is 1/4 of that of a full-size image. For an *N*-level refinement, the overall complexity is $(1/4 + 1/16 + \ldots + 1/4^N) < 1/3$ of the complexity of the full-resolution motion extrapolation, if the derivation of the initial motion is not considered.

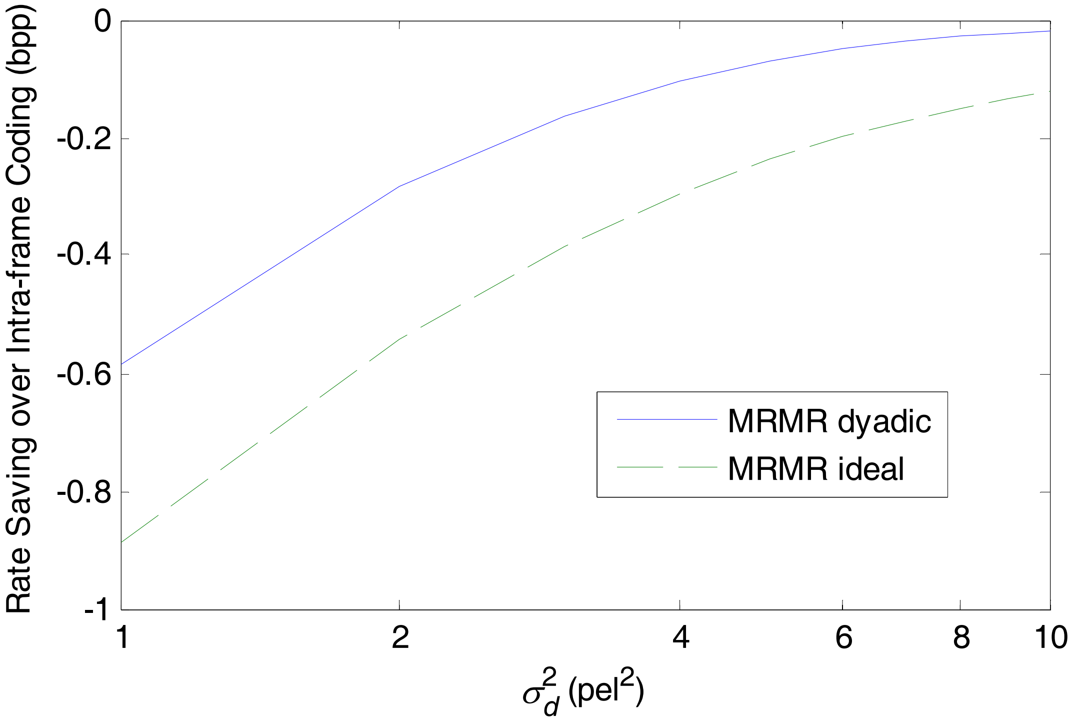


Figure 3.11. Rate loss with dyadic resolution progression.

Meanwhile, it is also worth noting that we actually sacrificed some performance to achieve the complexity reduction. In Section 3.3, the performance is analyzed based on the assumption that infinite levels of motion refinement are carried out. But in this codec we only allow dyadic resolution progression. Numerical results are illustrated in Figure 3.11 and we can see that the additional rate loss is averaged at around 0.17 bpp.

3.5.2 Benchmark performance

In this section, we compare the performance of motion extrapolation, MRMR and inter-frame ME under the basic BMA settings. That is, sequences are encoded in IPPP fashion, each frame is divided into 8×8 blocks, one MV is searched for each block at the integer-pel precision, and only one reference frame is used. Full search is employed and the search range is [–8, 8] × [–8, 8].

In the literature, prediction performance is usually measured by the PSNR of the SI. However, we notice that the motion accuracy in MRMR is progressive (see (3.24)) and the SI quality is higher for high-frequency components. For the same PSNR level, the residual energy is more "compact" in the frequency domain. Therefore we do not directly count on the PSNR values; instead we will take a closer look at the "MSE reduction" performance. More specifically, we treat the residual samples in each subband as i.i.d., and calculate the rate saving of the prediction by

$$\Delta R = \frac{1}{2}\log_2 \frac{MSE_1}{MSE_0}. \tag{3.41}$$

where MSE_1 is the mean-square-error (MSE) of the MCP, MSE_0 is the MSE of all-zero prediction (or if intra-frame coding is employed). Finally, the overall rate saving is calculated from the weighted sum of the rate saving at each resolution levels, where the weighting factors for level 1, 2 and 3 are 3/4, 3/16 and 3/64, respectively, accounting for the number of samples in each level.

Four QCIF sequences at 15 fps are used for testing: Foreman, Carphone, Mother&Daughter and News, each of which is 10 sec in length. We choose very fine quantization step sizes such that the frames are coded at very high quality (PSNR around 60dB). Thus there is no quantization error propagation and the prediction residual is almost purely due to motion mismatch. The results are shown in Table 3.1, from which we have the following observations.

Bits per sample[a]	Foreman				CarPhone			
	Lv1	Lv2	Lv3	Overall	Lv1	Lv2	Lv3	Overall
ΔR_{ext}	–0.04	–0.53	–1.16	–0.18	+0.06	–0.51	–1.20	–0.10
ΔR_{mrmr}	–0.51	–0.87	–1.31	–0.60	–0.55	–1.04	–1.52	–0.68
ΔR_{int}	–0.62	–1.34	–2.22	–0.82	–0.64	–1.47	–2.35	–0.86
Bits per sample	Mother&Daughter				News			
	Lv1	Lv2	Lv3	Overall	Lv1	Lv2	Lv3	Overall
ΔR_{ext}	–1.10	–1.80	–2.58	–1.28	–1.38	–1.67	–2.01	–1.44
ΔR_{mrmr}	–1.23	–1.86	–2.54	–1.39	–1.64	–1.70	–1.98	–1.65
ΔR_{int}	–1.29	–2.09	–2.97	–1.50	–1.72	–2.25	–2.84	–1.85

[a] Negative values with a greater absolute value means better prediction.

Table 3.1. Comparison of prediction performance among motion extrapolation, MRMR and inter-frame ME using basic BMA settings (8×8 blocks, interger-pel search, one hypothesis for each block).

For sequences with irregular motion (small ρ_t), MRMR achieves dramatic improvement over motion extrapolation. For example, 0.42 bpp more rate saving is observed for the Foreman sequence and 0.58 bpp is observed for the CarPhone sequence. While for sequences with more consistent motion (large ρ_t) such as Mother&Daughter and News, the improvement is less significant. But MRMR is still outperforms by 0.11 bpp and 0.21 bpp, respectively. At high rates, the corresponding rate savings obtained by MRMR can be translated into 0.66 to 3.49 dB in PSNR gain over motion extrapolation.

When compared to inter-frame ME, MRMR suffers a small amount of rate loss, which ranges from 0.11 bpp to 0.22 bpp.

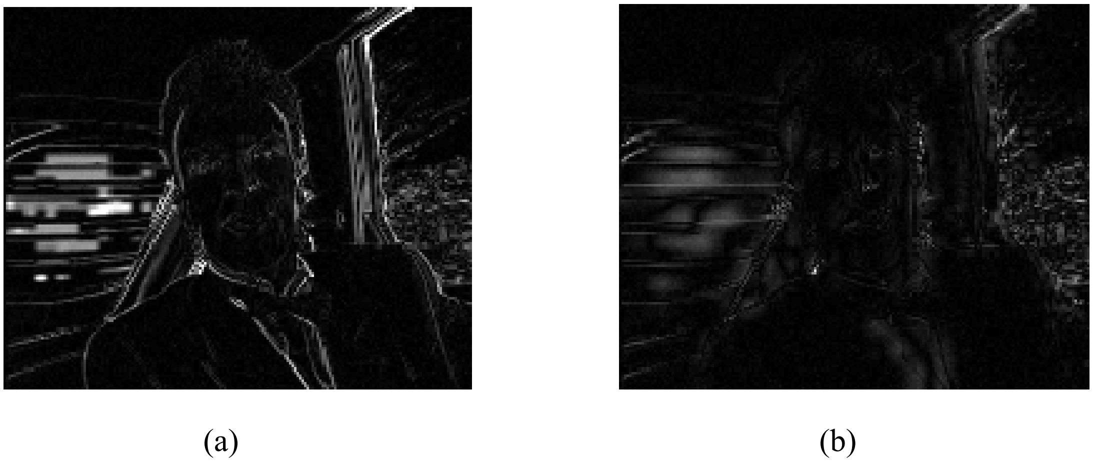
It is also observed that all of the three methods achieve more rate saving for lower frequency components. This can be explained from Figure 3.4, where it is shown that the transfer functions of the three methods are essentially high-pass. They attenuate the amplitudes of low-frequency components more effectively.

However, considering the number of samples in each level, the rate saving at high-frequency bands are more important for the overall prediction performance. Thus MRMR is favorable in the sense that its motion accuracy gets refined at high-frequency bands. From Table 3.1 we can see, even for the Mother&Daughter and News sequences, MRMR is worse than motion extrapolation in level three,[8] it still outperforms in the rest two levels. At the highest frequency bands, MRMR falls only 0.06 to 0.09 bpp behind inter-frame ME.

On the contrary, motion extrapolation is poor in predicting high-frequency components, especially for sequences with irregular motion. For the CarPhone sequence, the MCP results using motion extrapolation is even worse than the original signal in the highest subbands. As for the overall prediction performance, motion extrapolation falls as much as 0.76 bpp behind inter-frame ME, or is 4.6 dB worse in PSNR.

For a visual comparison, sample residual frames from the CarPhone sequence are shown in Figure 3.12. Significant improvement is observed by using MRMR.

[8] Meaning that the critical frequency here is greater than $\pi/8$.



(a) (b)

Figure 3.12. Residual frame of the side information using (a) motion extrapolation and (b) MRMR.

3.5.3 Extensive motion exploration

The previous section considers the prediction performance with basic BMA settings. Now we employ extensive motion exploration in MRMR. Details are discussed in the following.

A. Fractional-pel motion search

We enable quarter-pel search in MRMR. Pixel values at non-integer sample positions are interpolated using the same method as in H.264/AVC. Half-pel samples are derived by a six-tap finite-impulse-response (FIR) filter ([1, –5, 20, 20, –5, 1] / 32), and quarter-pel samples are generated from bilinear interpolation of both integer- and half-sample pixels. Hierarchical motion search is employed to reduce the computation:

full-search is first made on the integer-pel samples, and the best candidate MV is compared with its 8 neighbors in the half-pel samples, then in the quarter-pel samples.

B. Smaller block sizes

We use 2×2 blocks in the BMA. A smoothness constraint should be imposed to the motion field. For this purpose, a smoothing term is added to the matching criteria [105]:

$$E = E_{\mathrm{DFD}} + w_s E_s, \tag{3.42}$$

where DFD means “displaced frame difference”, E_{DFD}, in our case, is the sum of absolute difference (SAD) between the current block and the reference block, w_s is the weighting factor, and the motion smoothness term E_s is defined as

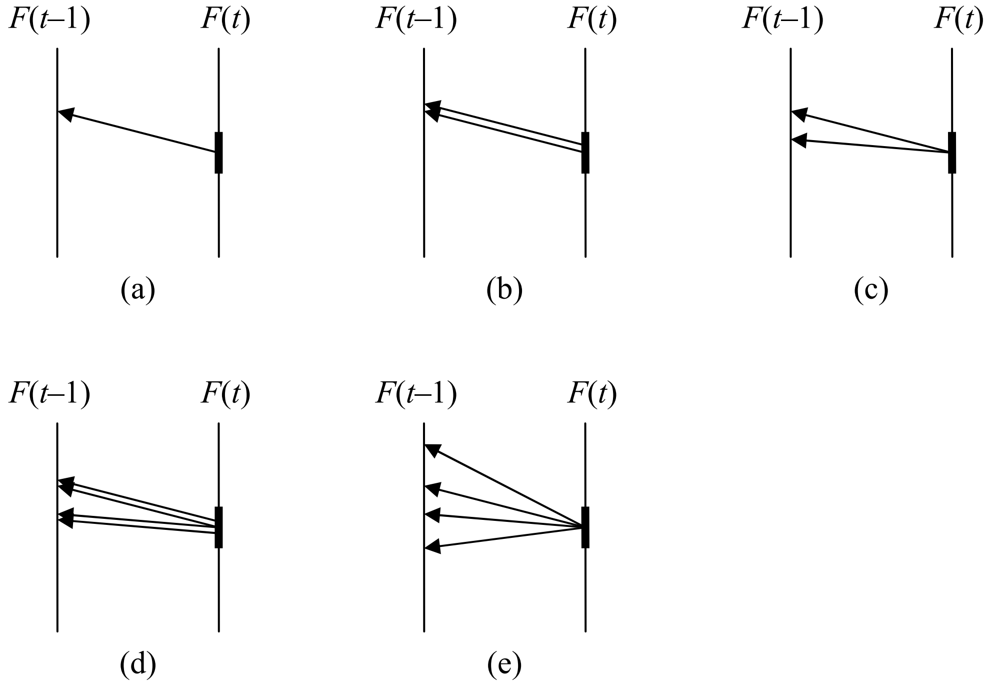
$$E_s = \sum_{n \in N_b} \left[\left| d_x(b) - d_x(n) \right|^p + \left| d_y(b) - d_y(n) \right|^p \right]. \tag{3.43}$$

where b is the current block, n is one of b’s four neighboring blocks, d_x and d_y denote the x and y components of an MV, p can be 1 or 2. Since E_s is calculated based on all of its neighbors, iterative refinement is needed in searching for the final MV.

In our simulations, we let $p = 1$, $w_s = 4$ and use SAD to calculate E_{DFD}. The initial motion field is assumed to be zero. At most 8 iterations are allowed.

C. Multiple-hypothesis prediction

The MHP algorithm is first proposed in [31] and is summarized as follows: The current block first searches for the best matching block in the reference frame. This particular block is used as the initial position of all *N* hypotheses. The joint optimization process fixes (*N*–1) hypotheses and searches for a new position for the remaining one. The criterion is to minimize the SAD between the current block and the average of the *N* hypotheses. After that, joint optimization is repeated on the rest of the hypotheses until the SAD converges (note that in each round the SAD value will be decreased, hence the convergence is guaranteed).



(a) (b) (c) (d) (e)

Figure 3.13. Illustration of the modified MHP algorithm. (a) best matching is searched for the 1st hypothesis; (b) set the initial position of the 2nd hypothesis to the same place of the 1st one; (c) jointly optimize the first two hypotheses; (d) the 3rd and the 4th hypotheses join, initialized to the current positions of the 1st and the 2nd hypotheses, respectively; (e) joint optimization of the first four hypotheses.

According to our analysis in Section 3.4, it is desired to have the multiple hypotheses less correlated to each other. Hence we have modified the algorithm as illustrated in Figure 3.13. We use 8 hypotheses in total, but we do not initialize all of them to the same place. Instead, only two of them are jointly optimize using the algorithm in the first stage. Then two more candidates will join, each of which is initialized at one of the resulting places in the first stage. Next, the second stage of optimization is carried out on the four hypotheses. Finally, the process is repeated with the joining of another four hypotheses. Such modification introduces diversity into the joint optimization process to make the MVs less correlated. Simulation shows that it is about 0.15 bpp better than the original approach.

Another important aspect is how to enable MHP with the smoothness constraint in (3.43) imposed. Eq. (3.43) considers the motion difference between neighboring blocks. However, in MHP, there are multiple MVs assigned to the same block, without any preferences among them. Consequently one might have to consider the motion difference of the current MV with respect to all the hypotheses of neighboring blocks, which is computationally expensive.

For such purpose, we further modify the MHP algorithm as follows. The 1st hypothesis of each block is searched for under the motion smoothness constraint (with respect to the 1st hypothesis of neighboring blocks). The resulting position is stored as the “search center” for the block. Then the multiple hypotheses are only searched around this search center.

D. Results

Saved Rates (bpp) over intra-frame coding	MRMR	H.264/AVC
Akiyo	–2.89	–2.58
Car Phone	–1.70	–1.88
Container	–3.48	–3.04
Football	–0.42	–0.75
Foreman	–1.88	–2.03
Miss America	–1.46	–1.53
Mother & Daughter	–2.09	–2.12
News	–2.80	–2.82
Suzie	–1.10	–1.31
Salesman	–2.44	–2.54
Average	–2.03	–2.06

Table 3.2. Comparison of the prediction performance between MRMR and H.264/AVC

We will compare MRMR with extensive motion exploration with the prediction performance of H.264/AVC. For H.264/AVC, the prediction results are generated by the reference software JM13.2 using the baseline profile (quarter-pel motion search, variable block size with the minimum being 4×4, at most one hypothesis for each block), and are wavelet decomposed for comparison. The first 100 frames of ten QCIF sequences at 30 fps are tested. Results are shown in Table 3.2

We can see that with extensive motion exploration, the prediction performance of MRMR can be very close to that of state-of-the-art inter-frame coding (only 0.03 bpp difference on average), without any overhead in transmitting the motion information. We can also see that typically, if MCP is effective for the sequence, where the achievable rate saving is higher than 2 bpp for both methods, MRMR is as good as, or even better than H.264/AVC; otherwise MRMR still suffers a performance gap (except for the Miss America sequence). This is partly because in MRMR, we simply set the prediction to be zero if no match is found in the ME, while in H.264/AVC, the intra-prediction mode will be triggered in this case, which also provides good prediction.

3.6 Conclusions

The bottleneck in improving the coding efficiency of WZVC is the performance of ME at the decoder side. The ME accuracy can be improved if the decoder has partial access to the current frame. In this chapter, we provide an analytical study on the rate-distortion performance of WZVC in a particular structure, where low-resolution images are progressively decoded and used for the motion refinement. Theoretical analysis shows that MRMR outperforms motion extrapolation dramatically for most practical sequences with medium or low temporal motion correlation. Even for sequences with very high ρ_t, refining the motion field is usually worthwhile when the resolution of the partially reconstructed frame is higher than a particular threshold (determined by the critical frequency). We also show that MRMR falls 1.5 dB behind conventional inter-frame ME, if the same BMA is used for both cases.

Realizing that decoder-side ME can benefit from more detailed motion information, which, if generated and transmitted by the encoder, incurs non-negligible bit overhead, we also provide theoretical analysis of the performance achieved by integrating MRMR with advanced ME techniques, including fractional-pel motion search, ME with smaller block sizes and multiple-hypothesis prediction. Results show that the gap between MRMR and inter-frame ME is not insurmountable.

In addition, we present a wavelet domain practical implementation of WZVC with MRMR. Simulation results show its superior performance over the simple motion extrapolation approaches. When combined with advanced ME techniques, MRMR shows comparable prediction performance as H.264/AVC.

Future researches will be carried out on estimating the efficiency of iterative motion refinement based on partially decoded frames with quality progression. We believe better understanding of the motion accuracy is the key to improve the performance of WZVC, and also has a significant impact on conventional video coding.

Chapter 4

Compression of Encrypted Images and Videos

4.1 Introduction

Conventionally in secure transmission of redundant data, as illustrated in Figure 4.1(a), the data is usually first compressed[9] and then encrypted at the sender side. At the receiver side, decryption is performed prior to decompression to recover the data. However, in some application scenarios, this conventional diagram needs to be revisited.

Let us consider the following case (illustrated in Figure 4.1(b)). Suppose Alice needs to transmit some data to Bob, while Charlie is the network provider. Alice wants to keep the data confidential to Charlie, however the resources that she has is too limited to compress the data. So Alice just gets the data encrypted using simple cipher and forwarded to Charlie. Charlie, as the network provider, always has the interest to reduce the data rate. That is, it is desirable for Charlie to *compress the encrypted data* even if he does not have access to the secret key. Johnson et al. prove in [50] that in this case, if stream cipher [98] is used and the data receiver (Bob) holds the secret key and performs joint decryption and decompression, the overall system performance can be as good as

[9] In this chapter, only lossless compression is considered.

the conventional approach. That is, neither the security nor the compression efficiency will be sacrificed by performing compression on the encrypted data.

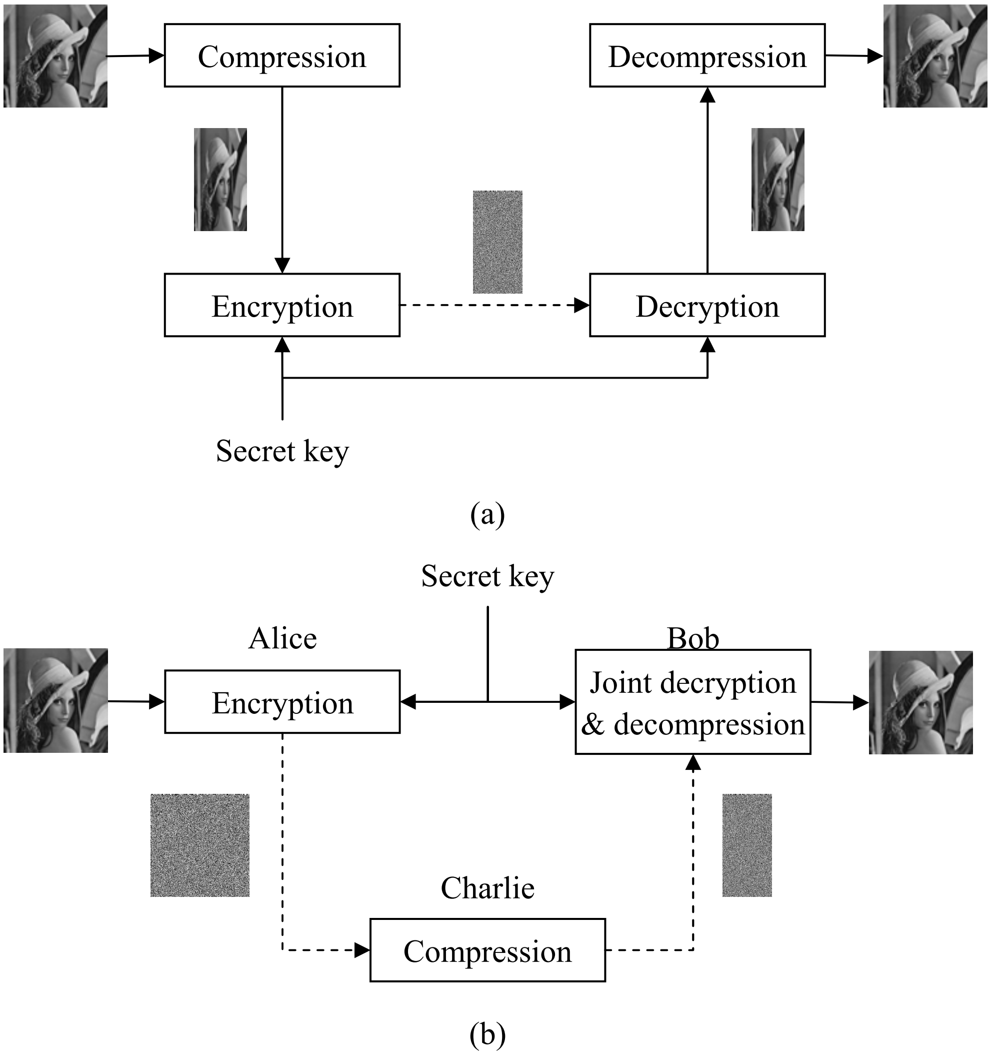


Figure 4.1. (a) Illustration of the conventional approach in which data is first compressed and then encrypted; (b) Illustration of compressing encrypted data. Here solid arrows represent secure channels and dashed arrows denote public channels.

A practical system to compress encrypted data is also proposed in [50]. For example, suppose the plaintext *X* is an i.i.d. source, and Alice uses a stream cipher (e.g. RC4 or DES in CFB mode [98]) as the encryption function:

$$Y = X \oplus K \tag{4.1}$$

where $\oplus$ denotes the bit-wise exclusive OR (XOR) operation, K is the key stream, and Y is the ciphertext.

Charlie gets Y without knowing K. He will encode Y through random binning: each sequence of n samples (denoted in Y^n) is randomly thrown into one of the bins that are indexed as $\{1, 2, \ldots, 2^{nR}\}$. Only the bin indices are transmitted by Charlie. So the actual sending rate for Y is R. Bob, upon receiving the bin index as well as the secret key, will treat K^n as the side information and look for its joint typical sequence $\hat{Y}^n$ inside the given bin. According to the Slepian-Wolf theorem, if and only if

$$R \geq H(Y|K) = H(X \oplus K|K) = H(X) \tag{4.2}$$

the reconstruction of Y^n can be asymptotically error-free. Finally the plaintext is reconstructed as

$$\hat{X} = \hat{Y} \oplus K\,. \tag{4.3}$$

Eq. (4.2) basically suggests that by employing SWC, the compression efficiency of the ciphertext can be just as good as compressing the plaintext. Other applications of compressing encrypted data include a third-party company providing storage to secured contents (online emails, surveillance videos), where the clients usually do not have the motivation to compress the data.

Although it has been shown that theoretically there is no performance loss in compressing encrypted data, we might still face challenges when it comes to practical applications. Considering real-world sources such as images or videos, which are typically highly correlated, a critical issue in improving the coding efficiency is how to exploit the source dependency. Conventional encoder-side decorrelation methods such as transform or prediction are not applicable here because the encryption function has masked the source dependency. In the literature, image or video data is usually modeled as a Markov source and the Markovian property is exploited in the Slepian-Wolf decoder [50][92]. Similar approach is also found in [115] for non-encrypted colored sources. Some good results have been reported for binary images. However, there are some limitations with this approach: first, Markov decoding in a Slepian-Wolf decoder is very expensive, especially in dealing with sources with non-binary alphabets; second, bit-plane based Markov decoding certainly reduces the complexity, but the source dependency that originally defined in the symbol domain is usually not fully utilized when translated to bit-planes (see Figure 4.2 for an example); third, since image or video data is known to be highly non-stationary, a global Markov model cannot describe its local statistics precisely. As reported in [92], for 8-bit grayscale images, only the first 2 most significant bit-planes (MSB) are compressible by employing a bit-plane based 2-D Markov model. How to effectively exploit the correlation of encrypted image data remains a challenging issue.

In this chapter, we propose an efficient way to compress encrypted images and videos through *resolution-progressive compression* (RPC). The encoder starts by sending a downsampled version of the ciphertext. At the decoder, the corresponding low-resolution image is decoded and decrypted, from which a higher-resolution image is obtained by inter- or intra-frame prediction. The predicted image, together with the secret encryption key, is used as the SI to decode the next resolution level. This process is iterated until the whole image is decoded. By doing so, the task of de-correlating the pixels, which is not possible for the encoder, is shifted to the decoder side. In addition, by having access to a low-resolution image, the decoder is able to learn the local statistics, doing much better than "blind" decoding. Moreover, by avoiding exploiting the Markovian property in the SWC, the complexity is significantly reduced.

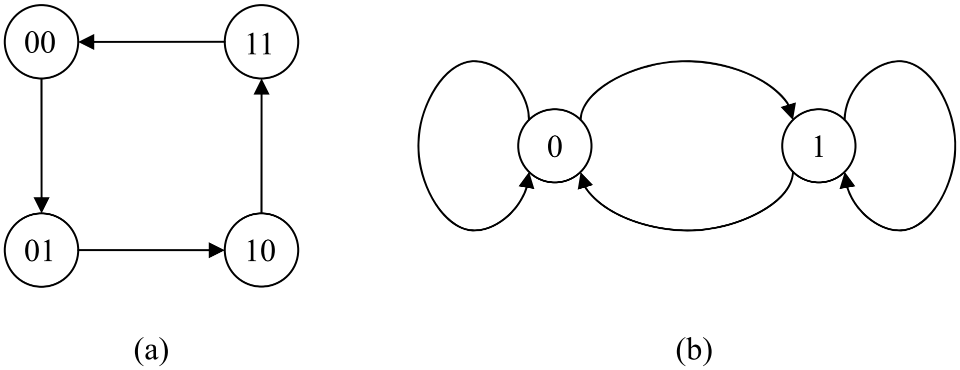


Figure 4.2. (a) Illustration of a four-state Markov process, where we have P(x_{i+1}=x_i+1) = 1, (here "+" is defined in $GF(4)$) hence by exploiting symbol-domain dependency, we have $H(x_{i+1}|x_i)$ = 0 bit. (b) Illustration of the Markov property in the MSB. We have P(b_{i+1}=b_i) = 0.5, hence the necessary bit-rate in the MSB (without looking at the LSB) is $H(b_{i+1}|b_i)$ = 1 bit. This shows that a bit-plane based Markov model cannot fully exploit symbol-domain source dependency.

The rest of the chapter is organized as follows. Compression of encrypted images and videos are discussed in Section 4.2 and 4.3, respectively. Section 4.4 concludes the chapter.

4.2 Compression of Encrypted Images

In this section, we will present the proposed RPC algorithm, analyze its performance and provide simulation results for compression of encrypted still images.

4.2.1 System description

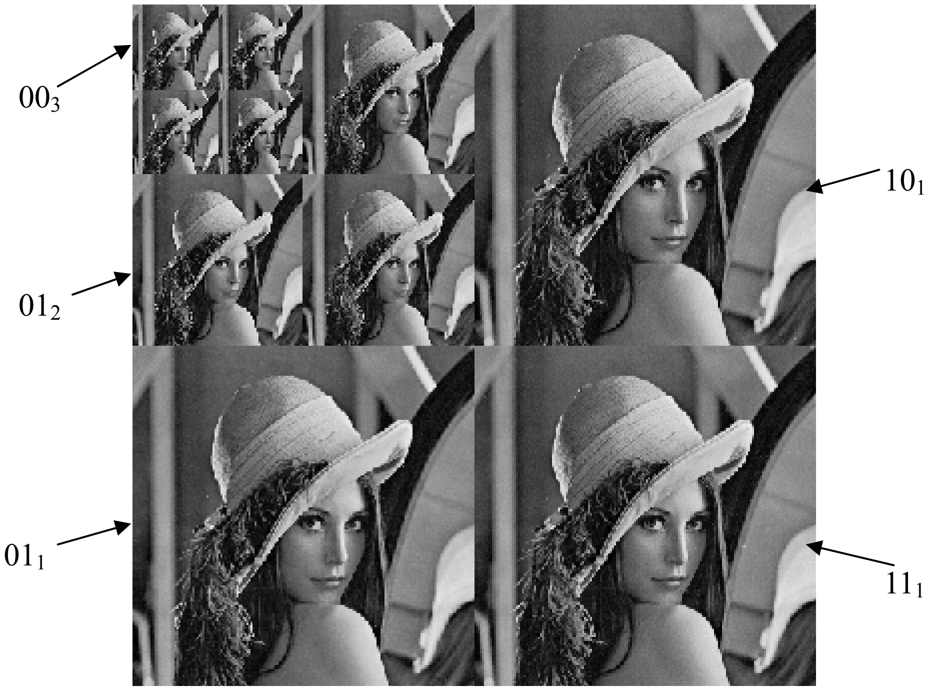


Figure 4.3. Illustration of three-level decomposition of the Lena image.

The encoder gets the ciphertext *Y* and decomposes it into four sub-images, namely, the 00, 01, 10 and 11 sub-images. Each sub-image is a downsampled-by-two version of the encrypted image. The name of a sub-image denotes the horizontal and vertical offsets of the downsampling. The 00 sub-image is further downsampled to create multiple resolution levels. We use 00_n to represent the 00 sub-image in the n-th resolution level. The 00_n sub-image can be losslessly synthesized from the 00_{n+1}, 01_{n+1}, 10_{n+1} and 11_{n+1} sub-images. An example of the decomposition is illustrated in Figure 4.3. Here the image is supposed to be an encrypted one. We use plaintext just for a better illustration. We would like to point out that the stream cipher function in (4.1) only scrambles the pixel values, but not shuffles the pixel locations. This means some geometric information of the pixels is still preserved, which is leveraged by the downsampling operation.

After the downsampling, each sub-image is encoded independently using SWC, and the resulting syndrome bits are transmitted from the lowest resolution to the highest.

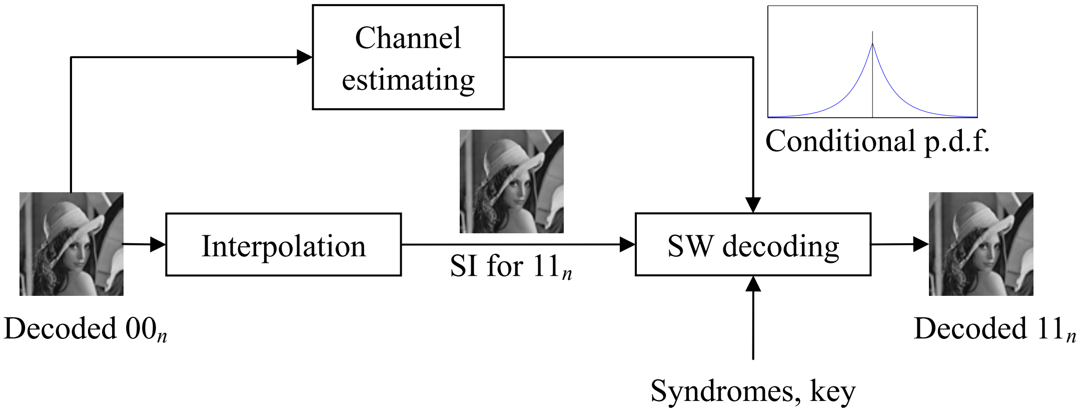


Figure 4.4. The decoder diagram.

Decoding starts from the 00 sub-image of the lowest-resolution level, say, level N. We suggest transmitting the uncompressed 00_N sub-image as the doped bits [92]. Thus the 00_N sub-image can be known without ambiguity, and some knowledge about the local statistics will be derived based on it. Next, other sub-images of the same resolution level are interpolated from 00_N. The interpolation result is used as the SI of the target sub-images. Meanwhile, a channel estimation module is employed to estimate the conditional p.d.f. of the original pixel values, given the SI. The SI, the estimated p.d.f., and the corresponding part of the key stream are fed into the Slepian-Wolf decoding module to decode the target sub-image. The decoder diagram is illustrated in Figure 4.4. When the 01, 10 and 11 sub-images of the same resolution level are all decoded, the 00 sub-image of the higher resolution level can be synthesized, then the decoding iterates until the full-resolution image is reconstructed.

It is worth noting that if the SI is a good approximation of the target sub-image, the pixels in the target sub-image can be considered as conditionally independent of each other (given the SI). In this case it is not necessary for the SW decoder to exploit the Markovian property of the source, which greatly reduces the computational complexity.

A feedback channel is needed for the encoder to know how many bits to transmit for each sub-image, which generally increases the transmission delay. However, this cost is reasonable because the decoder has no idea about the source statistics at the very beginning, and has to learn about it gradually during the decoding process.

4.2.2 Context-adaptive interpolation

The SI generation in our scheme is through interpolation. For the sake of simplicity, for any pixel in the target sub-image, we only use the 4 horizontal and vertical neighbors or the 4 diagonal neighbors in the known sub-image(s) for the interpolation. Intuitively, the SI quality will be better, if the neighbors are closer in geometric locations. Hence we use a two-step interpolation in each resolution level to improve the SI estimation. First, the 11 sub-image is interpolated from 00; after 11 is decoded, we use both 00 and 11 to interpolate 01 and 10. The interpolation pattern is illustrated in Figure 4.5, from which we can see another benefit of the two-step interpolation: the interpolation patterns of the two steps are isomorphic up to a scaling factor of $\sqrt{2}$ and a rotation of $\pi/4$. This simplifies the interpolator design.

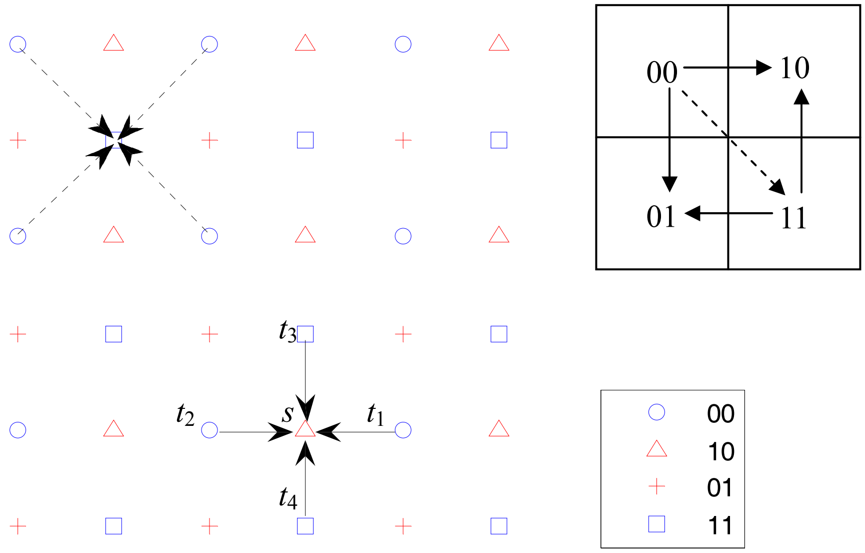


Figure 4.5. Illustration of the two-step interpolation at the decoder side. The dashed arrow denotes the first step interpolation, and the solid arrow denotes the second step.

Real-world image data is highly non-stationary, hence it is desired to have the interpolation adapted to the local context. For example, for a pixel on an edge, it is preferable to interpolate along the edge orientation. Similar efforts can be found in conventional lossless image compression, where the median edge detector (MED) [106] and the gradient adaptive predictor (GAP) [109] are two successful context adaptive predictors. However, they process the pixels in a raster-scanning order, thus cannot be directly applied to our scheme.

In this subsection, a simple, yet effective context adaptive interpolator (CAI) is proposed for our scheme. Due to the isomorphism, we only describe the horizontal-vertical interpolator illustrated in Figure 4.5.

Let s be the pixel value to be interpolated, $\mathbf{t} = [t_1, t_2, t_3, t_4]^T$ be the vector of neighboring pixels. The interpolator classifies the local region into four types: smooth, horizontally-edged, vertically-edged and other. In smooth regions, a mean filter is applied; in edged regions, the interpolation is done along the edge; otherwise we use a median filter. More specifically, the proposed CAI is formulated as

$$\hat{s} = \begin{cases} mean(\mathbf{t}) & \left(\max\left(\mathbf{t}\right) - \min\left(\mathbf{t}\right) \le 20\right) \\ \left(t_1 + t_2\right)/2 & \left(\left|t_3 - t_4\right| - \left|t_1 - t_2\right| > 20\right) \\ \left(t_3 + t_4\right)/2 & \left(\left|t_1 - t_2\right| - \left|t_3 - t_4\right| > 20\right) \\ median(\mathbf{t}) & (otherwise) \end{cases} . \tag{4.4}$$

In (4.4) it can be verified that the first condition contradicts the second (or the third) condition, thus a “smooth” region will never be estimated as “edged” again. The second and the third conditions are adapted from GAP, with an ad hoc threshold. It is also possible that the region is diagonally-edged, but there is no clue about on which side of the edge s lies. Therefore we simply adopt a median filter in this case.

4.2.3 Localized channel estimation

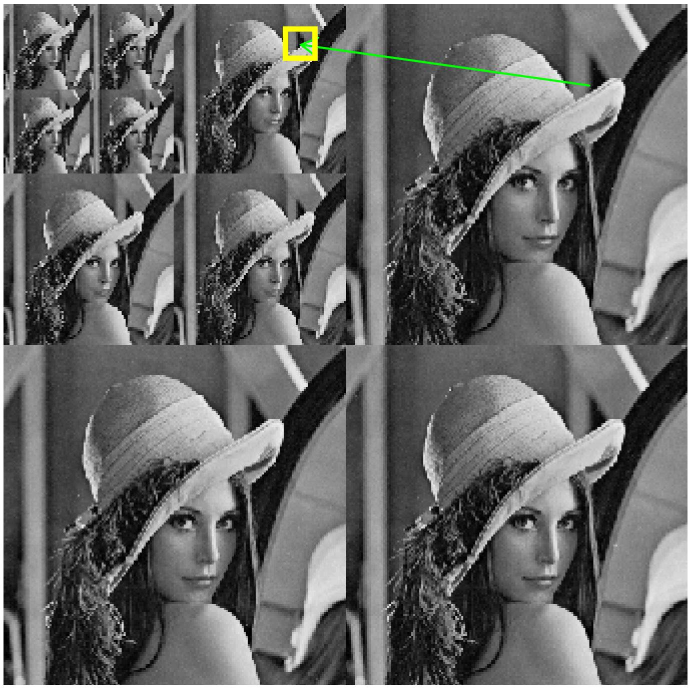


Figure 4.6. Illustration of the localized channel estimator.

SWC treats the SI as a noisy version of the source to be decoded. We can consider there is a virtual channel between the source and the SI [118]. To perform channel decoding, it is also necessary for the SW decoder to estimate the statistics of the virtual channel. In this work, we model the conditional p.d.f. of a pixel s to be Laplacian, centered at the given side information $\hat{s}$:

$$p\left(s\left|\hat{s}\right.\right)=\frac{\alpha}{2}\exp\left(-\alpha\left|s-\hat{s}\right|\right) \tag{4.5}$$

where $\alpha=\sqrt{2/\sigma_{s|\hat{s}}^{2}}$, and $\sigma_{s|\hat{s}}^{2}$ is the variance of $p\left(s\left|\hat{s}\right.\right)$. Hence it is necessary for the channel estimator to estimate $\sigma^2_{s|\hat{s}}$.

Due to the non- stationarity of image data, the accuracy of the SI could vary a lot in different areas. Generally $\sigma^2_{s|\hat{s}}$ at smooth areas will be much smaller than that at textured areas. Hence it is desired to have a localized channel estimator. In this work, $\sigma^2_{s|\hat{s}}$ is estimated from the neighboring prediction (interpolation) residual of the previously decoded level. As illustrated in Figure 4.6, Let s be a pixel in the 10_n sub-image, the channel estimator observes several geometrical neighbors of s in the 10_{n+1} sub-image. The neighborhood is chosen to be a 5×5 window. The mean square error (MSE) of the CAI results for these pixels is *scaled* to be used as $\sigma^2_{s|\hat{s}}$. The scaling is needed because for interpolation at higher-resolution levels, the correlation between the neighboring pixels is higher, which usually means smaller prediction residual. In this work, we adopt an empirical scaling factor of 0.75.

It can be seen that, both the CAI and the localized channel estimation are based on the assumption that the decoder can have an access to a lower-resolution reconstruction of the image. In other words, they are both enabled by the resolution-progressive decoding.

4.2.4 Performance analysis

In this subsection, theoretical analysis will be provided for RPC. Compared to the state-of-the-art conventional lossless image coding schemes such as JPEG-LS [106], RPC may suffer from two types of rate loss. The first type is the source coding loss, which is caused by the inefficiency of channel codes in achieving the Shannon limit. The second type is the image coding loss, caused by the inefficiency in removing the redundancy among the pixels. In this subsection we focus on the second type of loss. We will use an ideal source model to analyze the performance gap. Although real image data are not that simple, the analysis will still provide useful insights and help us understand the performance limit of RPC.

Suppose image X is a zero-mean, wide-sense stationary Gaussian source, with an isotropic autocorrelation function

$$r_{xx}\left(d\right) = \sigma_x^2 \exp\left(-\omega_0 \left|d\right|\right) \tag{4.6}$$

where σ_x is the standard deviation of X, ω_0 is a constant, and $|d|$ represents the Euclidean distance between any pair of pixels. In the following, we also use

$$\rho = \exp(-\omega_0) \tag{4.7}$$

to represent the correlation between any two horizontally or vertically neighboring pixels in the full resolution. Thus the correlation between any pair of pixels that are $|d|$ pixels away from each other is $\rho^{|d|}$.

The power spectrum density of X can be derived as

$$\Phi_{xx}(\omega) = \frac{2\pi\omega_0\sigma_x^2}{\left(\omega_0^2 + \omega^T\omega\right)^{3/2}} \tag{4.8}$$

where ω is the vector of spatial frequency (in radius). At high rates, the optimum encoder reaches the R-D bound:

$$R_{opt}(D) = \frac{1}{8\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}\log_2\frac{\Phi_{xx}(\omega)}{D}d\omega \tag{4.9}$$

where *D* is the distortion level (suppose *D* is small).

On the other hand, if the decoder does not exploit any source dependency of *X*, the memoryless encoding has the R-D function

$$R_{ml}(D) = \frac{1}{2}\log_2\frac{\sigma_x^2}{D}. \tag{4.10}$$

In RPC, the rate saving actually comes from the inter sub-image interpolation, because the variance of the residual is expected to be smaller than the original signal. According to the linear prediction theory [49], the optimum prediction produces the residual variance as

$$E\left\{\left(s-\hat{s}\right)^2\right\} = \sigma_x^2 - \mathbf{r}_{st}^T\mathbf{R}_{tt}^{-1}\mathbf{r}_{st} \tag{4.11}$$

where **t** is the vector of (known) neighbors, $\mathbf{R}_{tt} = E\{\mathbf{tt}^T\}$, $\mathbf{r}_{st} = E\{\mathrm{s}\mathbf{t}^T\}$. In different sub-images, we can estimate the element values of $\mathbf{R}_{tt}$ and $\mathbf{r}_{st}$ from Figure 4.5. Therefore, through some simple calculation, we can get the variances of the interpolation residuals in level *n* as:

$$\begin{cases}\sigma_{rpc}^2(01_n) = \sigma_{rpc}^2(10_n) = F\left(2n-1\right)\sigma_x^2 \\ \sigma_{rpc}^2(11_n) = F\left(2n\right)\sigma_x^2\end{cases}, \tag{4.12}$$

where the facilitating function *F*(*k*) is defined as

$$F(k) = \frac{1+2\rho^{\sqrt{2}^k} - 3\rho^{\sqrt{2}^{k+1}}}{1+2\rho^{\sqrt{2}^k} + \rho^{\sqrt{2}^{k+1}}}. \tag{4.13}$$

The overall R-D function for the proposed scheme can be derived by summing-up the bit rates in all sub-images:

$$
\begin{aligned}
R_{rpc}(D) &= \sum_{n=1}^{\infty} \frac{1}{2^{2n}} \sum_{(i,j)\in\{(0,1)(1,0),(1,1)\}} \frac{1}{2}\log_2 \frac{\sigma_{rpc}^2(ij_n)}{D} \\
&= \frac{1}{2}\log_2 \frac{\sigma_x^2}{D} + \sum_{k=1}^{\infty} \frac{1}{2^k}\log_2 F(k)
\end{aligned} \; . \tag{4.14}
$$

Here the weighting factor $1/2^{2n}$ accounts for the fact that the number of pixels in each sub-images in level n is $1/2^{2n}$ of the full-scale image, and we assume there are infinite resolution levels.

Since in memoryless coding, the source dependency is not exploited, the image pixels are encoded as i.i.d.; while in optimum coding and in RPC, the source dependency is fully or partially exploited. Thus both optimum coding and RPC will provide some rate saving over memoryless coding. From (4.9), (4.10) and (4.14) we obtain the rate saving over the memoryless coding by using RPC

$$
\Delta R_{rpc} = R_{rpc}(D) - R_{ml}(D) = \sum_{k=1}^{\infty} \frac{1}{2^k}\log_2 F(k) \tag{4.15}
$$

and by using optimum coding

$$
\Delta R_{opt} = R_{opt}(D) - R_{ml}(D) = \frac{1}{8\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi} \log_2 \frac{2\pi\omega_0}{\left(\omega_0^2 + \omega^T\omega\right)^{3/2}} d\omega \, . \tag{4.16}
$$

We can see that both ΔR_{opt} and ΔR_{rpc} are functions of ρ (or equivalently, functions of ω_0). In Figure 4.7 ΔR_{opt} and ΔR_{rpc} are compared with respect to different ρ values. We can see that both of them become more significant when ρ increases, and the gap between RPC and the optimum coder is almost a constant at 0.43 bpp for a large range of ρ values. For natural images, the correlation between neighboring pixels is usually high. In Table 4.1, the ρ values are estimated for “Baboon”, “Lena”, “Peppers”, “Boats” and “Goldhill”.

We can see that ρ typically ranges from 0.8 to 0.99. In this case, the proposed scheme saves 1.0 to 3.2 bpp over memoryless coding, which is as much as 70% to 90% of what the optimal conventional encoder can do. This suggests *the inter sub-image interpolator can effectively remove the pixel redundancy*.

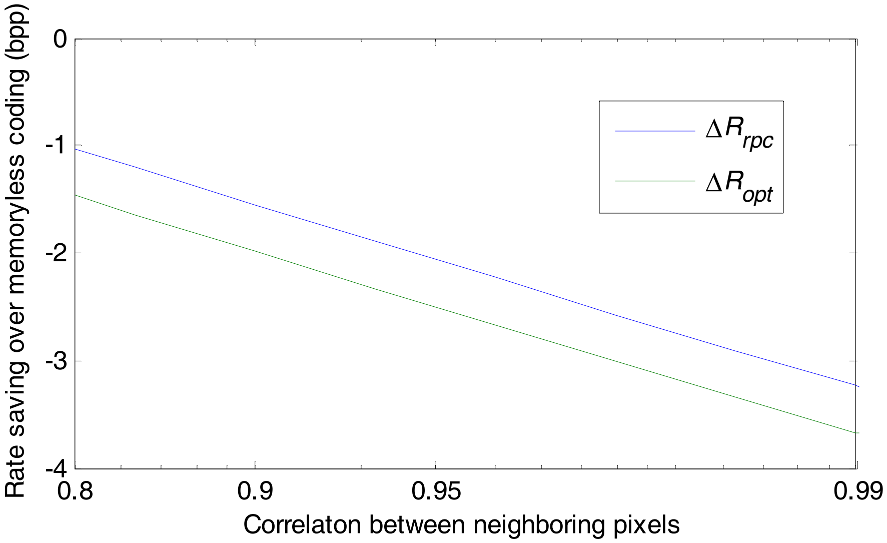


Figure 4.7. Comparison of ΔR_{opt} and ΔR_{rpc}.

	Baboon	Lena	Peppers	Boats	Goldhill
ρ value	0.81	0.97	0.98	0.97	0.99

Table 4.1. Correlation between neighboring pixels, where ρ is calculated as $\rho = (\rho_x+\rho_y)/2$, with ρ_x and ρ_y being the horizontal and vertical pixel correlation, respectively.

It is also observed that on the right hand side of (4.15), the terms to be summed up decrease quickly with respect to k, even for very high ρ values. This suggests we only need a few resolution levels to saturate the decomposition gain. For most images, a 3- or 4-level decomposition is sufficient.

More over, the result in (4.12) helps the design of our channel estimating module. If the wide-sense stationary model is assumed, the scaling factor for estimating $\sigma_{s|\hat{s}}^2$ should be $F(2n)/F(2n+2)$ for the 11_n sub-image, or be $F(2n-1)/F(2n+1)$ for the 10_n and the 01_n sub-images. Numerical results show that $F(k)/F(k+2)$ typically ranges from 0.5 to 1, depending on the ρ value. So we adopt 0.75 as the scaling factor. More sophisticated modeling might further improve the coding performance.

4.2.5 Simulation results

The images listed in Table 4.1 are used for testing. The encoder decomposes each encrypted image into 4 resolution levels. The sub-images in the lowest-resolution level are sent without compression. But the decoder still performs inter sub-image interpolation on them. The results will be used to estimate the p.d.f. of the pixels in the next level. For the other sub-images, we transmit the four least significant bit-planes (LSB) as raw bits, because there is not much gain to employ SWC on them. The four LSBs are sent *prior to* the MSBs, such that the decoder can have better knowledge about the pixels before starting decoding the MSBs. The four MSBs, on the other hand, are Slepian-Wolf encoded using rate-compatible punctured turbo codes [91] in a bit-plane based fashion. The sending rate of each Slepian-Wolf coded bit-plane is determined by the decoder's feedback.

(bpp)	Baboon	Lena	Peppers	Boats	Goldhill	average
MED	6.28	4.90	4.95	4.31	4.72	5.03
GAP	6.22	4.75	4.78	4.29	4.70	4.95
CAI	6.25	4.68	4.60	4.36	4.75	4.93

Table 4.2. Comparison of the residual entropy among MED, GAP and CAI

Firstly, we focus on the performance of the inter sub-image interpolation. The Shannon entropy of the residual is used as the criterion, and the results of MED, GAP and CAI are listed in Table 4.2. For CAI, what we are interested in is the conditional entropy $H\left(s\middle|\hat{s}\right)$. However, we will calculate $H\left(s-\left[\hat{s}\right]\right)$ as an upper bound, where [·] denotes rounding to the nearest integer. From Table 3.2 we can see that CAI provides similar or better performance than MED and GAP.

	Baboon	Lena	Peppers	Boats	Goldhill	average
CALIC	5.88	4.48	4.42	3.83	4.39	4.60
$\mathrm{RPC}_{\mathrm{G}}$	6.65	5.05	5.05	4.91	5.10	5.35
$\mathrm{RPC}_{\mathrm{L}}$	6.55	4.88	4.85	4.74	4.99	5.20

Table 4.3. Comparison of the compression performance among CALIC, resolution-progressive compression using globally estimated variance ($\mathrm{RPC}_{\mathrm{G}}$) and using locally estimated variance ($\mathrm{RPC}_{\mathrm{L}}$).

Secondly, let's study the coding performance of the resolution-progressive compression (RPC) scheme. In $\mathrm{RPC}_{\mathrm{L}}$, we estimate the local variance of prediction residual using the method described in Section 4.2. In $\mathrm{RPC}_{\mathrm{G}}$, we assume the decoder knows the global variance of the prediction residual in each subband (although this is practically not possible). The results are shown in Table 4.3, from which we can see, RPC_L outperforms RPC_G by 0.1 to 0.2 bpp. This means localized channel estimation is more effective than a global one, and this localization is enabled by the resolution progressive compression.

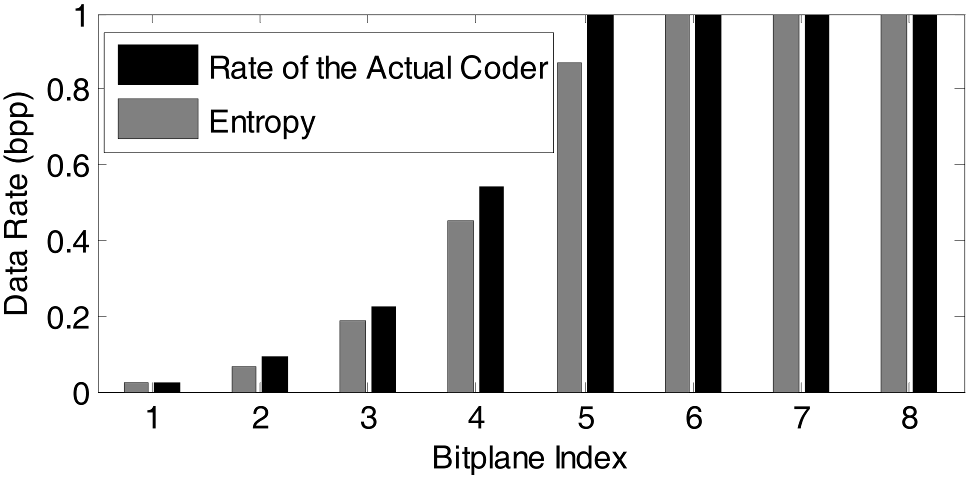


Figure 4.8. Coding result for the "Lena" image. Entropy is calculated from the LSB to the MSB.

There is no numerical results reported in [92] for grayscale images. The authors just mention that only the first two MSBs are compressible. As a comparison, the average code length for each bit-plane (across different sub-images) for "Lena" is plotted in Figure 4.8. Significant compression is observed in the first 4 MSBs, which is superior to the result reported in [92]. In Figure 4.8, the entropy for each bit-plane is also plotted. The difference between the entropy and the actual coding rate is the source coding loss, which is summed up to 0.29 bpp for the Lena image.

For comparison purpose, we also list the actual coding rate of CALIC, a very good conventional lossless image codec. The gap between RPC_L and CALIC is 0.60 bpp on average, which consists of both the source coding loss (0.31 bpp on average) and the image coding loss. The image coding loss is due to that the codec is still not able to remove the redundancy as efficiently as CALIC. For example, CALIC uses bias cancellation and run length coding to further deal with the redundancy among the prediction residual. These techniques can not be directly applied to our scheme.

4.3 Compression of Encrypted Videos

To compress a video signal, the source dependency needs to be exploited both in the temporal domain and the spatial domain. In the previous section, we have discussed how to exploit the spatial domain redundancy at the decoder side. In this section, we will further exploit the temporal domain redundancy in an integrated framework.

4.3.1 System description

We still use stream cipher to encrypt the video, and employ resolution progressive compression. Each frame is encoded independently using the method described in 3.2.1 for still images. In fact, there is not much difference here from the system used in Chapter 3, if we treat the downsampling operation as equivalent to performing a 2-D analysis filter bank on a frame, with $H_0(z)$=1, $H_1(z)$=z. (The corresponding synthesis filter bank is

$G_0(z)=1$, $H_1(z)=z^{-1}$.)

At the decoder side, multi-resolution motion refinement (MRMR) is used to exploit the temporal domain redundancy. The motion compensated prediction (MCP) is treated as the temporal SI (SI$_s$). In the meantime, the intra-frame interpolation result is treated as the spatial SI (SI$_t$). The overall prediction is the linear combination of them:

$$\mathrm{SI} = \alpha \times \mathrm{SI}_t + (1 - \alpha) \times \mathrm{SI}_s \tag{4.17}$$

and the weighting factor α is chosen to be

$$\alpha = \frac{MSE_s}{MSE_s + MSE_t} \tag{4.18}$$

where MSE$_s$ and MSE$_t$ are the spatial and temporal mean square error in the co-located block of the reference frame.

The frames are supposed to be encoded in an “IIPPPP…” structure. That is, except for the first two frames, all others will use previously decoded frames as references; the first two frames are encoded in the intra mode, but MCP is still performed for the second frame for the decoder to learn the source statistics.

4.3.2 Performance analysis

According to [37], the PSD relationship between the original signal s and the MCP residual e is

$$\Phi_{ee}(\omega) = \Phi_{ss}(\omega)\left[1 - \exp\left(-\omega^T \omega \sigma_{\Delta d}^2\right)\right]. \tag{4.19}$$

Details about (4.19) can be found in Section 3.2.

In our algorithm, ME is based on the 00_n sub-image, while MCP is performed for the 01_n, 10_n and 11_n sub-images. If block matching motion search is employed for ME, according to our analysis in Chapter 3, the variance of motion errors will be

$$\sigma_{\Delta d}^2(n) = 2^n kB\sigma_d^2. \tag{4.20}$$

where k = 0.005 for CIF sequences, B is the block size used for matching, and σ_d^2 denotes the motion intensity with respect to the reference frame (see Chapter 3 for more details).

On the other hand, for the 01_n, 10_n and 11_n sub-images, the autocorrelation functions can still be characterized using (4.6), by just replacing ω_0 with $2^n\omega_0$. Therefore, the PSD for the three sub-images is

$$\Phi_{ss}(\omega) = \frac{2^{n+1}\pi\omega_0\sigma_x^2}{\left(2^{2n}\omega_0^2 + \omega^T\omega\right)^{3/2}}. \tag{4.21}$$

Substituting (4.20) and (4.21) into (4.19), and according to Parseval's relation, we derive the variance of the prediction residuals in level n as

$$\sigma_e^2(n) = \frac{1}{4\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}\frac{2^{n+1}\pi\omega_0\sigma_x^2}{\left(2^{2n}\omega_0^2 + \omega^T\omega\right)^{3/2}}\left[1-\exp\left(-\omega^T\omega\times 2^n kB\sigma_d^2\right)\right]d\omega. \tag{4.22}$$

The overall rate for inter-frame predicted RPC is

$$R_{rpc-inter}(D) = \sum_{n=1}^{\infty}\frac{3}{2^{2n}}\times\frac{1}{2}\log_2\frac{\sigma_e^2(n)}{D} \tag{4.23}$$

where $\frac{3}{2^{2n}}$ accounts for the ratio of pixel number in the n-th level with respect to the whole frame.

In Figure 4.9, we plot the rate saving performance of inter-frame RPC over memoryless coding. For comparison purpose, results for intra-frame RPC and optimum intra coding are also plotted. We have assumed k = 0.005 and B = 4, and let σ_d^2 take values from {1, 2, 4, 8}. We can see from the figure that the performance of inter-frame RPC is critically related to the motion intensity. Whenever σ_d^2 doubles, the rate saving is roughly reduced by 0.5 bpp.

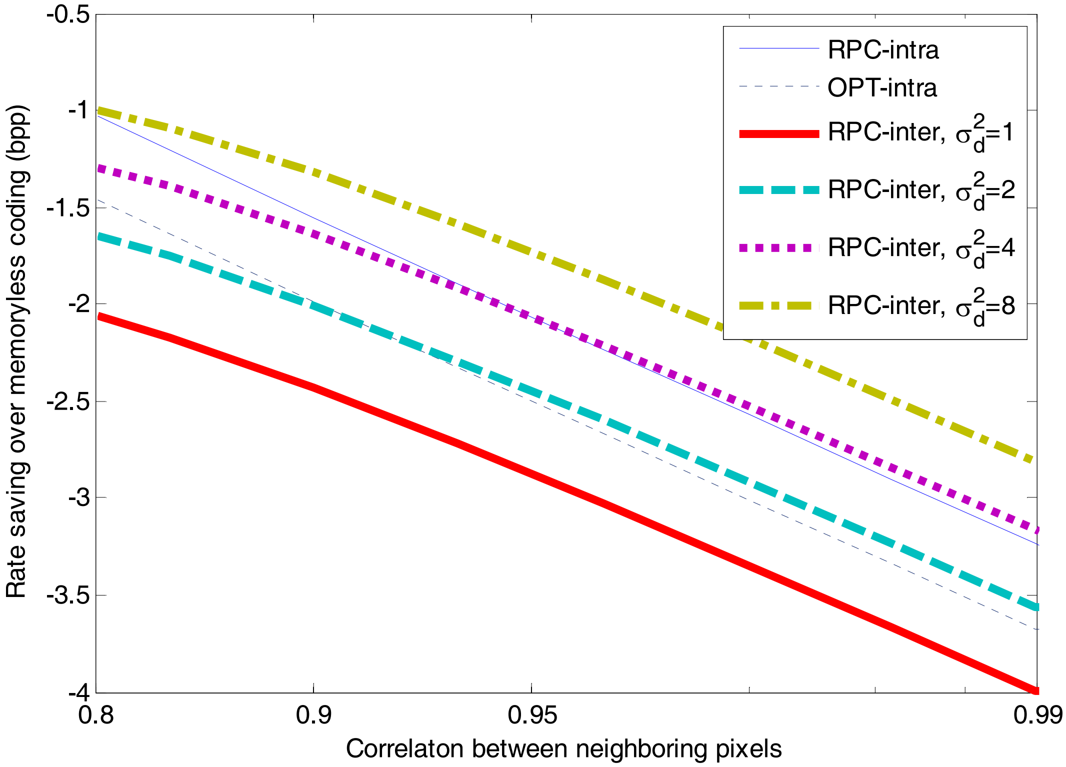


Figure 4.9. Rate saving performance of inter- or intra-frame RPC over memoryless coding.

When compared to intra-frame coding, RPC-inter is better for frames with smaller inter-pixel correlation. That is because RPC-inter does not further exploit the spatial redundancy among the prediction residual, while the corresponding rate loss becomes more significant if the original pixels are highly correlated in the spatial domain.

For low-motion sequences, RPC-inter outperforms RPC-intra a lot; for medium motion sequences, RPC-inter and RPC-intra perform similarly; while for high motion sequences, RPC-inter is less efficient than RPC-intra. So we have integrated the spatial and temporal prediction in (4.17). We call it RPC-hybrid.

4.3.3 Simulation results

We test three CIF sequences: Foreman, Football and FlowerGarden, each of which is at 30 fps. In Table 3.4, we have shown the results of RPC-intra, RPC-inter and RPC-hybrid, and compare them to Berkeley's result in [92].

(bpp)	Foreman	Football	FlowerGarden	average
Berkeley's	5.36	7.43	6.59	6.46
RPC-intra	4.80	5.73	6.12	5.55
RPC-inter	4.46	6.07	5.19	5.24
RPC-hybrid	4.15	5.35	5.09	4.86

Table 4.4. Comparison of the compression performance among Berkeley's approach in [92], resolution-progressive compression using spatial prediction only (RPC-intra), using temporal prediction only (RPC-inter) and using hybrid prediction (RPC-hybrid).

From Table 4.4 we can see that the RPC approach achieves much better results than Berkeley's approach. Even if we do not exploit any temporal domain dependency, RPC-intra saves about 0.9 bpp more. RPC-inter generally outperforms RPC-intra, except for sequences with very irregular motion such as Football. The integration of the spatial and temporal SI further improves the compression efficiency. RPC-hybrid achieves 1.6 bpp more saving over Berkeley's results on average.

4.4 Conclusions

Compression of encrypted sources is another application of DSC. For compression of encrypted real-world sources such as images or videos, efficient exploration of the source dependency is the key to improve the coding performance. Conventional approaches that exploit Markov properties in the SWC decoder do not work well for encrypted grayscale images and videos. We propose resolution progressive compression for this problem, which has been shown to have much better coding efficiency and less computational complexity than existing approaches. The success of RPC is based on enabling partial access to the current source at the decoder side and improving the decoder's learning of the source statistics. Our approach can also be extended to low-complexity pixel-domain DVC without considering encryption, where the DVC encoder does not exploit any spatial/temporal dependency of the video signal. Future research involves improving spatial/temporal prediction (for example, incorporating extensive motion exploration is an effective way to improve the temporal prediction) and better modeling of the virtual channel between the SI and true pixel values.

Chapter 5

Power-optimized Rate-allocation for Slepian-Wolf Coding over Wireless Sensor Networks

5.1 Introduction

We have mentioned that power consumption is the most concerned issue in the designing of communication systems over a WSN. For multi-terminal data aggregation in a WSN, we are interested in power-aware DSC, because the data gathered by the sensor nodes are typically highly correlated, and the "simple encoding, complex decoding" principle of DSC is very suitable for WSNs.

To minimize the power consumption in transmitting the encoded bits and maximize the operational lifetime of the battery-powered sensor nodes, careful rate-allocation (RA) is needed among the sources. In [27], separable cost functions with the linear and the exponential cost models are considered, and the RA problem is solved for the linear model. However, in wireless communications, the exponential model is more appropriate as suggested by Shannon's channel capacity formula. Solution for the exponential cost model has only been given for the two-source case in [27]. In this chapter, we address the problem for a general *N*-source case and propose a fast algorithm to search for the optimal rate point recursively. Compared to the exhaustive search approach, the proposed scheme reduces the computational complexity significantly.

The rest of the chapter is organized as follows. The power-efficient RA problem is formulated in Section 5.2. A water-filling model is established, based on which a fast RA algorithm is proposed with its feasibility and optimality proved in Section 5.3. Simulation results are shown in Section 5.5 and conclusions are drawn in Section 5.4.

5.2 Problem Formulation

Let's consider a set of discrete-time sources X_1, …, X_N, each of which is independent and identically distributed (i.i.d.) over time, takes values from a discrete alphabet and has a finite entropy. The sources are encoded separately in N different source nodes at rates R_1, …, R_N, respectively. The encoded bits are transmitted over a WSN to a sink node, where joint decoding is performed. Lossless reconstruction[10] is possible if and only if the rate point (R_1, …, R_N) lies in the Slepian-Wolf region defined by [26]:

$$R\left(\Phi\right) \geq H\left(X\left(\Phi\right) \middle| X\left(\Phi^c\right)\right), \forall \Phi \subseteq I_N \tag{5.1}$$

where $R\left(\Phi\right)=\sum_{k\in\Phi}R_k$, $X\left(\Phi\right)=\left\{X_k : k\in\Phi\right\}$, I_N = {1, …, N}, and Φ^c denotes the complementary set of Φ (with the universal set being I_N).

As mentioned above, one of the most essential cost metric in a WSN is the power

[10] It is worth noting that practical SWC schemes based on channel coding are not strictly lossless. It only means the decoding error probability can be arbitrarily small. However, we will use "lossless" in this chapter for simplicity.

consumption. During the transmission, power consumption is needed not only at the source nodes but also at the intermediate nodes (see Figure 5.1). We are interested in minimizing the overall cost of the entire WSN.

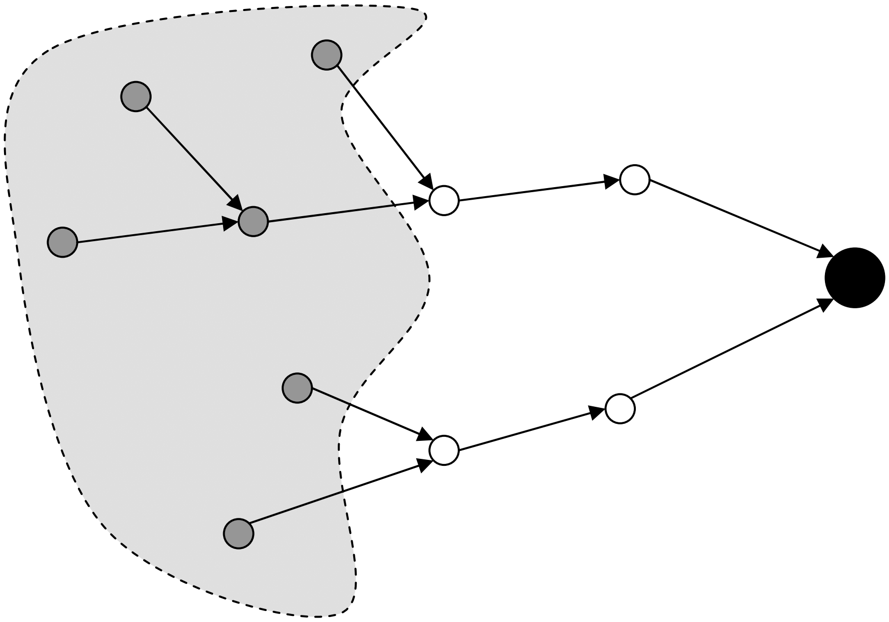


Figure 5.1. Illustration of data gathering in a wireless sensor network. In this scenario the shaded area denotes the region of interest. The gray nodes are sources nodes that sense data from the region. The data might be temperature, moisture, pressure or surveillance video gathered from the region. The data is to be transmitted through some intermediate nodes (white ones) to the sink node (the black one). Note some source nodes can also serve as intermediate nodes. During the transmission, power is needed for both source nodes and the intermediate nodes. We are interested in minimizing the overall power consumption of the network.

In this chapter, we assume the encoded bits are transmitted over a packet switching network using unicast, and packets are routed along the shortest path. That is, a data flow is formed between each sensor and the sink. We also assume that the transmissions of different data flows do not interact with each other. For example, two packets arriving at an intermediate node are neither assembled together, nor subject to any data processing such as network coding. This is typically the case in WSNs as the sensor nodes are not designed to be so powerful in functionality. In this scenario, the cost function can be modeled as the sum of costs to communicate between the sensors and the sink

$$C = \sum_{k=1}^{N} c_k \left(R_k \right) \tag{5.2}$$

where c_k is a topology-dependent cost function for the k-th data flow. It is non-negative and non-decreasing in general.

A good model for the cost functions is established in [27] and summarized as follows. Let w_k>0 be the weighting factor (which, e.g., may reflect the noise level or the fading factor of a wireless link) assigned to the shortest path from X_k to the receiver, then the multiple cost functions are unified in the form of

$$c_k \left(R \right) = w_k \times c\left(R \right), \forall k \in I_N \tag{5.3}$$

where $c(\cdot)$ depends only on the rate value. Two typical examples for $c(R)$ are the linear cost model with $c(R) = R$ for wired networks, and the exponential model with $c(R) = \exp(R)$ for wireless networks. With the linear cost model, the min-cost rate point can be easily found and the result is presented in [27]. However, with the exponential cost model, which is more typical in WSNs, the problem has not been completely solved and will be addressed in this chapter. The problem is formulated as: given the Slepian-Wolf region defined in (5.1), find the rate point R^* in the Slepian-Wolf region that minimizes the overall cost of the WSN:

$$C = \sum_{k=1}^{N} w_k \times \exp\left(R_k\right) \tag{5.4}$$

A better illustration of the model is shown in Figure 5.2

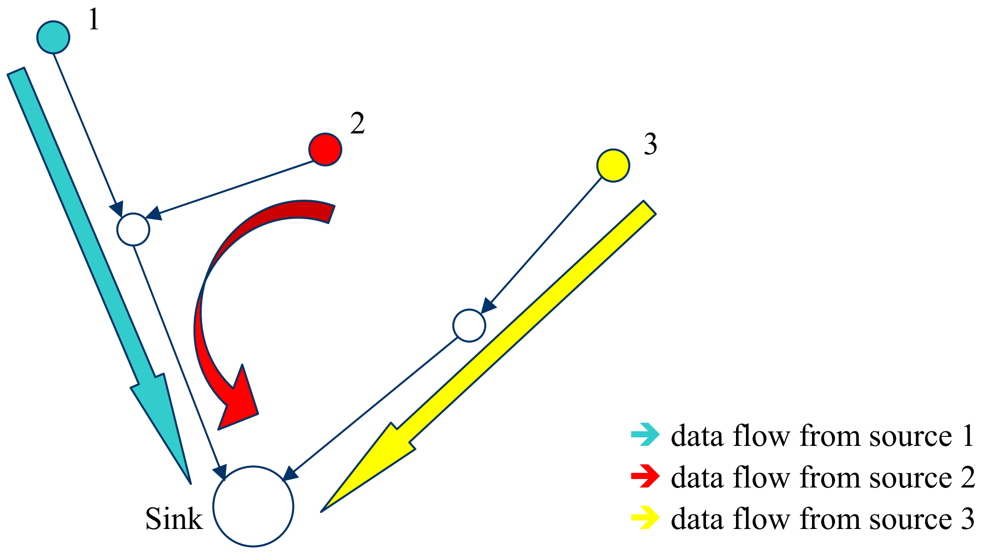


Figure 5.2. Modeling the power consumption of a WSN. Suppose that one data flow is formed for each source. Each data flow is transmitted along the shortest path. For each single link in the data flow, we have the Shannon theorem $R = \frac{1}{2}\log\left(1 + \frac{P_S}{P_N}\right)$, where R is the data rate, P_S is the transmission power, and P_N is the noise power. Then we approximately have $P_S = P_N \times \exp(R)$. For the entire data flow, the total power consumption is $P_S = (P_{N1} + P_{N2} + \ldots) \times \exp(R) = w \times \exp(R)$, where w is the weight of the shortest path. Then the overall power consumption of the network is the summation of all data flows, which leads to (5.4).

When $N = 2$, a closed-form solution is given in [27] using Lagrangian multipliers. Note that the approach in [27] searches the problem space exhaustively. When it comes to the more general case of N sources, the computational complexity increases rapidly (see the beginning of Section 5.3 for more detailed discussions). On the other hand, for many optimization problems, it is a common practice to develop greedy algorithms, which might be much more efficient. In the next section, a greedy approach is proposed to find the min-cost point in a recursive manner, based on a novel water-filling model.

5.3 Low-complexity Rate-allocation

The Slepian-Wolf region defined in (5.1) is the intersection of multiple half-spaces, therefore it is convex (not strictly). On the other hand, (5.4) defines a strictly-concave surface. Thus there is one and only one min-cost rate point R^* in the Slepian-Wolf region, and R^* must lie on the boundary the Slepian-Wolf region. That is, R^* must satisfy at least one of the equalities in (5.1). The main idea of the proposed algorithm is summarized as follows. According to Corollary 1 presented in Subsection 5.3.1, the equality $R(I_N) = H(X(I_N))$ must hold for R^*. If none of the other equalities in (5.1) holds for R^*, it is straightforward to apply Lagrangian multipliers and the complexity is low. However, that is not always the case: R^* might satisfy some other equalities in (5.1). If this happens, R^* can be achieved by first applying SWC to *a subset of the sources* independent of others, then using them as side information to decode other sources. In other words, we can treat the N-source RA problem recursively and reduce the number of sources in each recursion. Now the problem is how to find a suitable subset of the sources while still being able to achieve the minimum cost.

If the receiver does not know how to choose the suitable subset to reduce the problem, it might have to traverse every non-empty subset Φ of I_N, perform the

Lagrangian on $X(\Phi)$, solve the sub-problem of RA for the rest (N–$\|\Phi\|$) sources $X(\Phi^c)$ ($\|\Phi\|$ denotes the cardinality of Φ), combines the result and checks if it satisfies (5.1). Denote the complexity of this approach for n sources as $F(n)$. It is straightforward to derive the recurrence

$$F\left(n\right)=\sum\nolimits_{\Phi}F\left(n-\|\Phi\|\right)=\sum\nolimits_{k=1}^{n}C_n^kF\left(n-k\right), \tag{5.5}$$

using $F(0) = F(1) = 1$ as the initial condition. The first few $F(n)$ values are given in Table 5.1. We can see that $F(n)$ grows rapidly as n increases. A conservative estimate of $F(n)$ is $F(n) > nF(n–1)$ for $n \geq 2$. Thus we conclude $F(n)$ grows as $\Omega(n!)$ – an unacceptable complexity in most applications.

n	*F*(*n*)	*n*	*F*(*n*)
1	1	6	4,683
2	3	7	47,293
3	13	8	545,835
4	75	9	7,087,261
5	541	10	102,247,563

Table 5.1. Complexity growth of the exhaustive search

In the following we will present a fast yet optimal algorithm which makes greedy choices in each recursion to reduce the problem. A water-filling model is introduced for this purpose.

5.3.1 Water-filling model for power-efficient rate-allocation

Water-filling models have been used in conventional source coding of multiple correlated sources [18]. Extensions and modifications are needed to fit the model for our problem.

We use N tubes to represent the rate space of the N sources. We also introduce another (2^N–1–N) virtual tubes, each of which holds the sum of the rates of a certain subset of sources. The total 2^N–1 tubes represent the 2^N–1 inequalities in (5.1). Now we can symbolize each tube by using a subset Φ of I_N, and a lower bound on rate is marked at tube Φ as in (5.1). Besides this lower bound, an upper bound is also defined for tube Φ as

$$R(\Phi) \le H(X(\Phi)). \tag{5.6}$$

If the amount of water in a tube is less than the lower bound, we say there is an underflow; on the other hand if the amount of water in a tube is more than the upper bound, there is an overflow; when the amount of water equals the lower/upper bound, we say the tube is about to be underflowed/overflowed. *A rate point is inside the Slepian-Wolf region if and only if none of the tubes is underflowed.*

With the above water-filling model defined, we have the following proposition:

Proposition 1: if a rate point is inside the Slepian-Wolf region, and two tubes Φ_1 and Φ_2 are about to be underflowed simultaneously, then the two tubes $\Phi_1 \cap \Phi_2$ and $\Phi_1 \cup \Phi_2$ are both about to be underflowed.

Proof: By definition we have

$$R\left(\Phi_1\right)=H\left(X\left(\Phi_1\right)\mid X\left(\Phi_1^c\right)\right),R\left(\Phi_2\right)=H\left(X\left(\Phi_2\right)\mid X\left(\Phi_2^c\right)\right). \tag{5.7}$$

On the other hand, we also have

$$\begin{aligned}
&R\left(\Phi_1\right)+R\left(\Phi_2\right)=\sum\nolimits_{k\in\Phi_1}R_k+\sum\nolimits_{k\in\Phi_2}R_k=\sum\nolimits_{k\in\Phi_1\cup\Phi_2}R_k+\sum\nolimits_{k\in\Phi_1\cap\Phi_2}R_k\\
&=R\left(\Phi_1\cup\Phi_2\right)+R\left(\Phi_1\cap\Phi_2\right)\\
&\geq H\left[X\left(\Phi_1\cup\Phi_2\right)\middle|X\left(\left(\Phi_1\cup\Phi_2\right)^c\right)\right]+H\left[X\left(\Phi_1\cap\Phi_2\right)\middle|X\left(\left(\Phi_1\cap\Phi_2\right)^c\right)\right]\\
&=H\left(X\left(\Phi_1\right)\middle|X\left(\Phi_1^c\right)\right)+H\left[X\left(\Phi_2\setminus\Phi_1\right)\middle|X\left(\left(\Phi_1\cup\Phi_2\right)^c\right)\right]\\
&\qquad+H\left(X\left(\Phi_2\right)\middle|X\left(\Phi_2^c\right)\right)-H\left[X\left(\Phi_2\setminus\Phi_1\right)\middle|X\left(\Phi_2^c\right)\right]\\
&=R\left(\Phi_1\right)+R\left(\Phi_2\right)+H\left[X\left(\Phi_2\setminus\Phi_1\right)\middle|X\left(\left(\Phi_1\cup\Phi_2\right)^c\right)\right]-H\left[X\left(\Phi_2\setminus\Phi_1\right)\middle|X\left(\Phi_2^c\right)\right]\\
&\geq R\left(\Phi_1\right)+R\left(\Phi_2\right)
\end{aligned} \tag{5.8}$$

where in the third line, the inequality is because none of the tubes is underflowed, in the fourth line, the equality is from the chain rule of conditional entropy (where "\" denotes the set difference), in the sixth line, the equality is from (5.7), and in the last line, the inequality is because $(\Phi_1\cup\Phi_2)^c \subseteq (\Phi_1)^c$ (also from the chain rule of conditional entropy).

Based on (5.8), we conclude that the equality always holds in the second line of (5.8). This only happens when $\Phi_1\cap\Phi_2$ and $\Phi_1\cup\Phi_2$ are both about to be underflowed.

Proposition 1 holds for any rate point in the Slepian-Wolf region. Now we switch the discussion to the min-cost rate point R^*. For R^*, we can find all the tubes that are about to be underflowed and derive the union of them, denoted as Φ_u. If $\Phi_u \neq I_N$, we can always find a source from $X(I_N\setminus\Phi_u)$ (so that none of the tubes containing this source is about to be underflowed), decrease its bitrate by a small amount and still keep the rate point inside the Slepian-Wolf region, however the overall cost in (5.4) is reduced. Hence the following two corollaries are in place:

Corollary 1: The min-cost point R^* must have $R(I_N) = H(X(I_N))$.

Proof: As shown above. Note that in this case, the tube I_N is about to be underflowed, and also about to be overflowed.

Corollary 2: The min-cost point R^* must have none of its tubes overflowed.

Proof: If a tube is overflowed, according to Corollary 1, its complementary tube is underflowed, meaning that R^* is out of the Slepian-Wolf region.

Then the dual of Proposition 1 is stated as:

Proposition 2: If R^* is the min-cost rate point, and there are two tubes Φ_1 and Φ_2 that are about to be overflowed simultaneously, then the tubes $\Phi_1 \cap \Phi_2$ and $\Phi_1 \cup \Phi_2$ are both about to be overflowed.

Proof: Similar to that of Proposition 1.

5.3.2 Greedy rate-allocation algorithm

Now that we have established a water-filling model, we are in a position to describe a greedy RA algorithm based on this model. The algorithm will be performed by the receiver node which is assumed to be less constrained by power than the source sensor nodes, and the results of the optimal rate allocation will be fed back to each source node for compression.

According to Corollary 1, the total amount of water (rate budget) is $R(I_N) = H(X(I_N))$.

We fill the water into the tubes as if the bit-rates are allocated. It is desirable that when the water is completely filled into the tubes, the obtained rate point is in the Slepian-Wolf region with the minimum possible cost.

Supposedly, at some point, we have allocated R_i to X_i and R_j to X_j. To increase one of them by an arbitrarily small amount of bit-rate ΔR, one might introduce a cost increment of $w_i\exp(R_i)\Delta R$ or $w_j\exp(R_j)\Delta R$, respectively. The allocation scheme should pick the smaller of them, until the cost increments being the same, i.e., when the following relationship holds:

$$\ln w_i + R_i = \ln w_j + R_j. \tag{5.9}$$

After that, R_i and R_j should be increased evenly until some tube is about to be overflowed.

So the algorithm shall pre-fill the tubes to $-\ln w_1$, …, $-\ln w_N$, respectively (without loss of generality, suppose none of the tubes is overflowed at this time). After that we still have $H\left(X\left(I_N\right)\right)+\sum_{k=1}^{N}\ln w_k$ rate budget in hand. Then we start to fill all the “real” tubes evenly, and we also keep a close watch on both the real and the virtual tubes. The filling continues until an overflow is about to occur in a tube Φ_0 (Φ_0 can always be found because eventually the upper bound of I_N will be reached). Now we can separate the N-source SWC into two phases: $X(\Phi_0)$ are encoded and decoded among themselves first, and then used as side information to decode others (others are encoded as if they knew $X(\Phi_0)$). This essentially reduces the problem to an $(N - \|\Phi_0\|)$-source case. The algorithm can be executed recursively until finally all the sources are coded.

Before we present the algorithm in a more rigorous way, a function η is defined as

$$\eta\left(\Phi,\Phi_1\right)=\frac{H\left[X\left(\Phi\right)\middle|X\left(\Phi_1\right)\right]+\sum_{j\in\Phi}\ln w_j}{\left\|\Phi\right\|} \tag{5.10}$$

for any $\Phi\neq\phi$, $\Phi\cap\Phi_1=\phi$ (ϕ denotes the empty set). Here Φ_1 represents the set of sources that are decoded in advance and used as side information for others. In the right-hand side of (5.10), the numerator represents the remaining capacity of tube Φ, and the denominator is the speed of filling water into tube Φ (assuming unit speed in filling water to each of the real tubes). So $\eta(\Phi, \Phi_1)$ has the physical meaning: with Φ_1 as the side information, the time needed to overflow Φ. We would pick the Φ with the minimum η value in each recursion. Note that this is why the algorithm is "greedy": we make the choice seemingly the best at the moment without looking ahead into any sub-problems generated by picking other Φs.

The algorithm is summarized formally as follows:

1) Let $n = N$, $\Phi_1 = \phi$.
2) Calculate the $2^n - 1$ entropy values $H[X(\Phi)|X(\Phi_1)]$, where Φ is any non-empty subset of $I_N\backslash\Phi_1$.
3) Find Φ with the minimum $\eta(\Phi, \Phi_1)$, denote it as Φ_0. Ties are broken arbitrarily.
4) Set $R_k = \eta(\Phi_0, \Phi_1) - \ln w_k$ to each X_k for all $k\in\Phi_0$.
5) Set $n \leftarrow n-\|\Phi_0\|$, $\Phi_1 \leftarrow \Phi_1\cup\Phi_0$. If $n = 0$, end the program; otherwise go to 2) and continue.

In this algorithm, each recursion runs at most $O(2^n)$ time. In the worst case the size of n is reduced by 1 every time. That gives us a maximum overall complexity of $O(2^N + 2^{N-1} + \ldots + 2^1) = O(2^N)$, which is in the same order of the problem input (we need (2^N-1) arguments to specify an N-source Slepian-Wolf region). Compared to the $\Omega(N!)$ complexity of using exhaustive search, it is a huge saving.

5.3.3 Proof of feasibility

We first show that the algorithm is feasible. That is, by running the algorithm, we eventually come up with a rate point that is inside the Slepian-Wolf region. For this purpose, we will show that at each recursion, the selected sources $X(\Phi_0)$ can be Slepian-Wolf coded using the allocated rates. Or equivalently, we should have:

$$R(\Phi) \geq H\left(X(\Phi) \middle| X(\Phi_0 \setminus \Phi) \cup X(\Phi_1)\right) \tag{5.11}$$

for any $\Phi \subseteq \Phi_0$, $\Phi_0 \cap \Phi_1 = \phi$.

Eq. (5.11) is adapted from (5.1), with $X(\Phi_1)$ as the side information. When $\Phi = \Phi_0$, it is easy to verify that the equality in (5.11) holds; when $\Phi \neq \Phi_0$, we have the following inequality:

$$
\begin{aligned}
& R(\Phi)=\sum\nolimits_{k\in\Phi}R_k=\|\Phi\|\times\eta(\Phi_0,\Phi_1)-\sum\nolimits_{k\in\Phi}\ln w_k \\
&=\|\Phi_0\|\times\eta(\Phi_0,\Phi_1)-\left(\|\Phi_0\|-\|\Phi\|\right)\times\eta(\Phi_0,\Phi_1)-\sum\nolimits_{k\in\Phi}\ln w_k \\
&=H\left[X(\Phi_0)\middle|X(\Phi_1)\right]+\sum\nolimits_{j\in\Phi_0\backslash\Phi}\ln w_j-\left(\|\Phi_0\|-\|\Phi\|\right)\times\eta(\Phi_0,\Phi_1) \\
&\geq H\left[X(\Phi_0)\middle|X(\Phi_1)\right]+\sum\nolimits_{j\in\Phi_0\backslash\Phi}\ln w_j-\left(\|\Phi_0\|-\|\Phi\|\right)\times\eta(\Phi_0\backslash\Phi,\Phi_1) \\
&=H\left[X(\Phi_0)\middle|X(\Phi_1)\right]-H\left[X(\Phi_0\backslash\Phi)\middle|X(\Phi_1)\right] \\
&=H\left(X(\Phi)\middle|X(\Phi_0\backslash\Phi)\cup X(\Phi_1)\right)
\end{aligned}
\tag{5.12}
$$

where in the fourth line of (5.12), the inequality is because Φ is a subset of Φ_0 and Φ_0 is the set with the minimum η value in the current recursion (hence $||\Phi|| \leq ||\Phi_0||$ and $\eta(\Phi_0, \Phi_1) \leq \eta(\Phi_0\backslash\Phi, \Phi_1)$).

Then, the following theorem holds.

Theorem 1 (feasibility): The greedy RA algorithm ends up with a rate point inside the Slepian-Wolf region.

Proof: Immediately from (5.11).

5.3.4 Proof of optimality

Now we will prove the rate point found by the proposed algorithm is actually the min-cost point R^*. For the sake of brevity and without loss of generality, we assume w_1=w_2=…=w_N=1 in this subsection. First we prove the following proposition:

Proposition 3: The min-cost point R^* must have each of its component rates $R_k \geq \eta^*$, where

$$\eta^* = \min_{\substack{\Phi\subseteq I_N \\ \Phi\neq\phi}} \frac{H\left(X\left(\Phi\right)\right)}{\left\|\Phi\right\|}. \tag{5.13}$$

Proof: If this is not true, without loss of generality, let $R_1 \leq \ldots \leq R_M < \eta^* \leq R_{M+1} \leq \ldots \leq R_N$, where $1 \leq M < N$ (the case M=N is ignored because otherwise the tube I_N is underflowed and R^* is outside the Slepian-Wolf region).

Now let's consider the component rate R_{M+1}. If we can reduce R_{M+1} by an arbitrarily small amount while still keeping the rate point inside the Slepian-Wolf region, then R^* cannot be the min-cost point. So some tube(s) containing X_{M+1} is about to be underflowed. Let Φ_{M+1} be the *intersection* of all those tubes. We have the following observations about Φ_{M+1}:

1) $(M+1)\in\Phi_{M+1}$.
2) Φ_{M+1} is about to be underflowed (Proposition 1).
3) $\Phi_{M+1}\cap I_M = \phi$, where $I_M = \{1, \ldots, M\}$. Otherwise, for example, if $1\in\Phi_{M+1}$, let's consider replacing the pair (R_1, R_{M+1}) with $(R_1+\Delta R, R_{M+1}-\Delta R)$, where ΔR is an arbitrarily small positive number. This operation still keeps the rate point inside the Slepian-Wolf region[11], but the overall cost is decreased according to the Jensen's inequality (see (5.14)). This contradicts that R^* is the min-cost point. So none of the sources in I_M is in Φ_{M+1}.

$$\exp(R_1)+\exp(R_{M+1}) > \exp(R_1+\Delta R)+\exp(R_{M+1}-\Delta R). \tag{5.14}$$

In the same way we define $\Phi_{M+2}, \ldots, \Phi_N$. Now we consider the *union* of them: $\Phi =$

[11] Increasing R_1 does not affect any inequality in (5.1); and decreasing R_{M+1} may cause an about-to-underflow tube to be actually underflowed, but those tubes all contain R_1, so the amount of water of each of those tubes is not changed.

$\Phi_{M+1}\cup\ldots\cup\Phi_N$. Firstly, $\Phi\cap I_M = \phi$ because none of $\Phi_{M+1},\ldots,\Phi_N$ has a non-empty intersection with I_M; secondly, $\{M+1, \ldots, N\} \subseteq \Phi$ because $(M+1)\in\Phi_{M+1}$, etc. Combining the above two points we conclude that

$$\Phi = \{M+1, \ldots, N\}. \tag{5.15}$$

According to Proposition 1, Φ is about to be underflowed, so its complementary, $\Phi^c = I_M$, is about to be overflowed. Thus

$$M\eta^* > R\left(I_M\right) = H\left(X\left(I_M\right)\right) \geq \left\|I_M\right\|\eta^* = M\eta^* \tag{5.16}$$

where the first inequality is because $R_1 \leq \ldots \leq R_M < \eta^*$, the first equality is because I_M (or Φ^c) is about to be overflowed, and the second inequality is from (5.13). Now we can see the contradiction, by which Proposition 3 is proved.

Now the optimality of our algorithm is stated in the following Theorem:

Theorem 2 (optimality): The greedy RA algorithm results in the min-cost rate point R^*.

Proof: This can be proved by mathematical induction. If there is only one source, the statement is trivially true. If we assume that it is true for arbitrary n sources where $1 \leq n < N$, then for the N-source case, suppose the min-cost rate point is $R^* = \left(R_1^*, \cdots, R_N^*\right)$, and the rate point found by the greedy algorithm is $R' = \left(R_1', \cdots, R_N'\right)$.

In the greedy algorithm, the first tube to be overflowed, Φ_0, achieves the minimum in (5.13). At this time, for any $k \in \Phi_0$, we have $R_k' = H\left(X\left(\Phi_0\right)\right) / \left\|\Phi_0\right\| = \eta^*$.

The optimal rate point R^* must have each of the component rates $R_k^* = \eta^*$ (Proposition 3). On the other hand, according to Corollary 2, this point should also satisfy

$\sum_{k\in\Phi_0} R_k^* \leq H\left(X\left(\Phi_0\right)\right) = \left\|\Phi_0\right\|\eta^*$. Consequently, R^* must have $R_k = \eta$, for any $k \in \Phi_0$, which is the same as the result in the greedy algorithm.

Now $X(\Phi_0)$ can be Slepian-Wolf coded by themselves. After that the problem is reduced to an $(N - ||\Phi_0||)$-source case. According the induction hypothesis, the greedy algorithm is able to find the optimal rate point for the rest $(N - ||\Phi_0||)$ sources $X(I_N\backslash\Phi_0)$. On the other hand, the corresponding $(N - ||\Phi_0||)$ component rates of R^* also forms a point in the Slepian-Wolf-region for $X(I_N\backslash\Phi_0)$ (given the coded $X(\Phi_0)$ as side information), hence we have $\sum_{k\in\Phi_0^c} \exp\left(R_k'\right) \leq \sum_{k\in\Phi_0^c} \exp\left(R_k^*\right)$ and accordingly, $\sum_{k\in I_N} \exp\left(R_k'\right) \leq \sum_{k\in I_N} \exp\left(R_k^*\right)$. Since R^* is the only min-cost rate point (see the Section 5.3), we safely conclude that $R^{'} = R^*$.

5.4 Simulation

Let $Y = [Y_1, \ldots, Y_N]^T$ be N jointly Gaussian sources, with zero-mean and the covariance matrix Cov. Then its joint p.d.f. is written as

$$f_Y\left(y\right) = \frac{1}{\sqrt{\left(2\pi\right)^N \det\left(\text{Cov}\right)}} \exp\left[-\frac{1}{2} y^T \text{Cov}^{-1} y\right] \qquad (5.17)$$

where $y = [y_1, \ldots, y_N]^T$ is any instance of Y, and det(·) denotes the determinant of a matrix. Suppose discrete sources $X = [X_1, \ldots, X_N]^T$ is generated by performing uniform scalar quantization on Y with a small step size Δ. According to the results in [26], we have

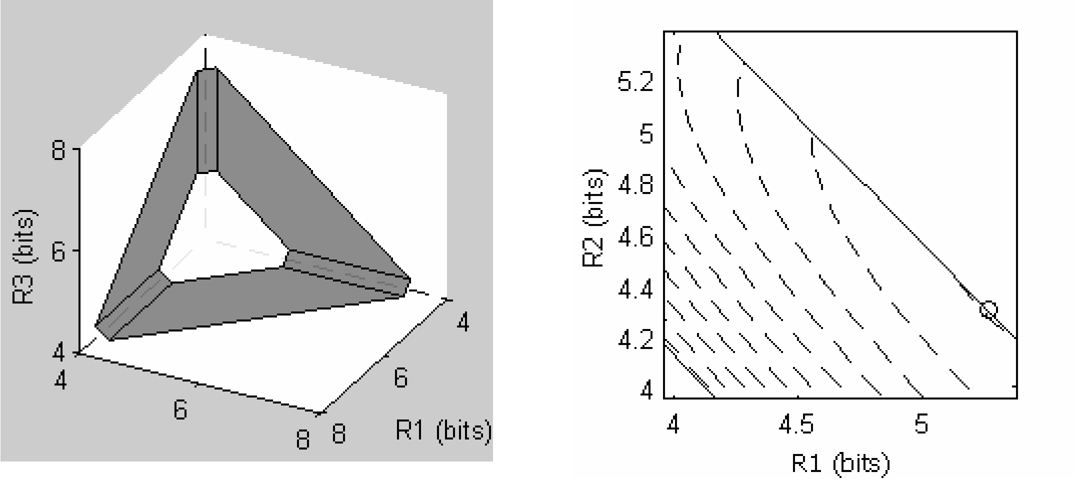
$$H\left(X\left(\Phi\right)\right) = h\left(X\left(\Phi\right)\right) - \left\|\Phi\right\| \log_2 \Delta \qquad (5.18)$$

where $h(\cdot)$ denotes the differential entropy. For jointly Gaussian sources,

$$h\left(X\left(\Phi\right)\right)=\frac{1}{2}\log_2\left[\left(2\pi e\right)^{\|\Phi\|}\det\left(\mathrm{Cov}\left(\Phi\right)\right)\right] \tag{5.19}$$

where Cov(Φ) denotes covariance matrix of $X(\Phi)$. Hence the right hand side of (5.1) can be calculated as

$$\begin{aligned}H\left(X\left(\Phi\right)\middle|X\left(\Phi^c\right)\right)&=H\left(X\right)-H\left(X\left(\Phi^c\right)\right)\\&=h\left(X\right)-h\left(X\left(\Phi^c\right)\right)-\|\Phi\|\log_2\Delta \qquad .\\&=\frac{1}{2}\log_2\left[\left(2\pi e\right)^{\|\Phi\|}\frac{\det\left(X\right)}{\det\left(X\left(\Phi^c\right)\right)}\right]-\|\Phi\|\log_2\Delta\end{aligned} \tag{5.20}$$



(a) (b)

Figure 5.3. (a) The SW rate region of 3 sources. The white hexagon is the minimum sum-of-rate plane; and (b) the hexagon is projected to the R_1R_2 plane, with the cost contours illustrated. The min-cost rate-point is marked with a circle.

We now illustrate the algorithm using a toy example. Let the covariance matrix be:

$$\mathrm{Cov}=\begin{bmatrix}1 & 0.9 & 0.9\\0.9 & 1 & 0.9\\0.9 & 0.9 & 1\end{bmatrix}. \tag{5.21}$$

and the quantization step size is 0.1. The test set contains 10^6 samples. The path weights are assumed to be $w_1 = 1$, $w_2 = \exp(1)$ and $w_3 = \exp(2)$.

We calculate the entropies / conditional entropies using (5.20). Then the 3-D SW region is illustrated in Figure 5.3(a). We can see that the minimum sum-of-rate plane R_1+R_2+R_3 = $H(X_1X_2X_3)$ is a hexagon in the SW region. We are particularly interested in the cost performance of the rate-points in the hexagon given Corollary 1. The hexagon is projected to the R_1R_2 plane in Figure 5.3(b) (the rectangular region sliced by two oblique lines). The cost

$$C = w_1 \exp\left(R_1\right) + w_2 \exp\left(R_2\right) + w_3 \exp\left(H(X_1X_2X_3) - R_1 - R_2\right) \tag{5.22}$$

is densely sampled inside the region and the contour lines are drawn. The min-cost rate point is found numerically at $R_1 = 5.27$, $R_2 = 4.27$ (and $R_3 = 3.96$).

On the other hand, if we apply the water-filling algorithm, the tube ($R_1 + R_2$) is the 1st to be overflowed, then R_1 = ($H(X_1X_2)$ + lnw_1 + lnw_2) / 2 – lnw_1 = 5.27, R_2 = ($H(X_1X_2)$ + lnw_1 + lnw_2) / 2 – lnw_2 = 4.27 are found; the only remaining source X_3 is coded at the rate $R_3 = H(X_3|X_1X_2) = 3.96$. This result matches the numerical result and supports the optimality of our algorithm.

At the optimal rate point, the cost is 7.76×10^2. As a comparison, the mean cost inside the hexagon is 1.08×10^3. This means, instead of randomly picking a point in the SW region with the minimum sum-of-rate constraint, working at the optimal rate point achieves roughly 30% saving in power consumption on average.

5.5 Conclusions

We consider the transmission of multiple Slepian-Wolf coded sources over a WSN. The goal is to minimize the overall transmission power consumption of the entire network. We show that this can be done through careful rate allocation according to the sources statistics and the network topology. The optimum rate allocation algorithm, when the cost metric is modeled as the weighted sum of exp(*rate*) of the multiple sources, is an open problem in the literature.

For SWC of multiple sources, we can first organize a “coding chain” $X(\Phi_1) \rightarrow X(\Phi_2) \rightarrow \ldots$, and allocate rates among $X(\Phi_k)$ using Lagrangian, with $X(\Phi_1 \cup \ldots \cup \Phi_{k-1})$ being the side information, without considering the rest of the sources. The difficulty is how to arrange the coding chain to optimize the power efficiency. Exhaustive search leads to unacceptable complexity. We have constructed a water-filling model for this problem, and proposed a greedy, yet optimum algorithm based on the model. Significant reduction is obtained in computational complexity. Future work will include jointly considering the quantizer design and the rate-allocation to achieve the best cost-distortion tradeoff.

Chapter 6

Conclusions and Future Works

Besides the fundamental rate-distortion tradeoff, there are a lot of other practical constraints to be considered in designing real-world image/video compression codecs. In recent years, with the growing popularity of wireless sensor networks, low-complexity image/video encoding for up-link transmissions has drawn increasing research interests. In this dissertation, we have studied the design of high-efficiency, low-cost and secure multimedia communication systems [63]–[73][121][123]. Our researches have been focused on facilitating the decoder's learning of the source statistics to achieve better coding efficiency and optimizing the power consumption of the system.

6.1 Summarization of the Contributions

- Multi-resolution motion refinement for Wyner-Ziv video coding

The accuracy of motion estimation plays an important role in improving the coding efficiency of Wyner-Ziv video coding. Most existing WZVC schemes perform ME at the decoder. The unavailability of the current frame at the decoder side typically limits the accuracy of ME, which results in the degradation of the coding efficiency of WZVC. To improve the accuracy of ME at the decoder, we propose to progressively decode the current frame in the resolution dimension, and iteratively refine the motion learning based on each partially-decoded frame.

In this dissertation, we presented an analytical model to estimate the potential gain by employing multi-resolution motion refinement (MRMR). Our theoretical results show that at high rates, WZVC with MRMR outperforms WZVC with motion extrapolation by as much as 5 dB, while the gap between MRMR and conventional ME is only about 1.5 dB, if the same motion search method is used.

We also showed that MRMR can benefit from extensive motion exploration, without spending any overhead bits to represent the motion field. We studied the performance of MRMR with fractional-pel motion search, with smaller block sizes and with multiple-hypothesis prediction. Our theoretical results show that the performance of MRMR can be greatly enhanced by incorporating these advanced ME techniques. The practical system we designed achieves prediction performance comparable to H.264/AVC.

- Slepian-Wolf compression of encrypted images and videos

Compression of encrypted data can be achieved by employing Slepian-Wolf coding. However, how to efficiently exploit the source dependency in an encrypted image or video remains a challenging issue. Previous works incorporate 2-D Markov models in the SWC, which is not accurate enough for natural grayscale images; as a result, the compression performance is usually poor.

In this dissertation, we propose to compress encrypted images/videos progressively, such that the decoder can observe a low-resolution version of the image or the current frame, from which local statistics/motion information is learned and used for the decoding of the next resolution level. Another benefit is that we do not have to exploit Markovian properties in the SWC decoder, which greatly reduces the complexity. Our real-world lossless codec has achieved significant coding efficiency improvements over existing approaches for both images and videos.

- Power-optimized rate allocation for SWC over wireless sensor networks.

Power consumption is one of the most critical concerns in communications over wireless sensor networks. Depending on the network topology, the cost to transmit one bit from one source node might be different from another. On the other hand, for multiple correlated sources, to increase the coding rate for one source may reduce the necessary rate(s) for other source(s). Thus it is possible to minimize the overall transmission power consumption of the network through sophisticated rate allocation. However, fast rate-allocation algorithms are only found in the literature for linear cost models. For wireless communications, exponential cost models are more appropriate. In this case, an exhaustive search based approach produces unacceptable complexity for optimum rate allocation.

In this dissertation, we established a novel water-filling model for this rate allocation problem. Based on this model, we proposed a greedy rate allocation algorithm to search for a rate point that allows lossless reconstruction of the sources, while minimizing the power consumption of the entire network. The feasibility and optimality of the proposed solution are mathematically proved. Compared to the exhaustive search based approach, our algorithm dramatically reduces the computational complexity.

6.2 Future works

Future researches can be carried out from the following perspectives.

- For WZVC with decoder-side ME, efforts shall be made to complete the design and analysis of the rate-distortion models. The key is to estimate the motion accuracy when the (partially reconstructed) current frame and the reference frame(s) are in various forms, including the case that the partially reconstructed frame is a low-quality version of the current frame, that the reference frame is several frames away from the current frame, that the reference frame(s) is also a corrupted version. Those are all typical scenarios in video communications over error-prone channels. In particular, we are aiming at constructing a model to estimate the motion accuracy using phase-based motion estimation [54], where the partially reconstructed image is an arbitrary noisy version of the original one. With this model, it is possible for the encoder to packetize the bit-stream in a more efficient way in terms of decoder-side motion estimation accuracy. In other words, we can encode into the base or lower layer(s) the information that is most critical for the motion learning at the decoder side. Thus we can minimize the video coding loss, with almost no overhead in the bitstream. This is a promising approach and is expected to achieve results not possibly achievable by other ad hoc approaches.

- Practical implementations of WZVC with progressive motion refinement will be another important work. We need to incorporate the extensive motion exploration into a practical WZVC codec. The key issue is to model the virtual channel between the MCP results and the true pixel/coefficient values. This is also important to compression of encrypted videos. We also plan to develop a fine-grained scalable (FGS) DVC paradigm, where the video signal is partitioned into a base layer and multiple enhancement layers, with the enhancement layers WZ coded. This way, we keep the nice property of conventional FGS coding, i.e., better error resiliency if information gets lost in the enhancement layers, while the coding efficiency of the enhancement layers is improved because inter-frame correlation can be exploited flexibly by the decoder.
- We also need to study power-efficient multi-terminal image/video compression. The joint correlation of multiple frames needs to be modeled for the scenarios where there is only the base layer information available. Another important issue is the modeling of the cost function. In Chapter 4 we only consider the

power consumption for communication/transmission, but sometimes the power consumption for signal processing is not negligible. We plan to extend our algorithm to fit in arbitrary cost model with potentially multiple-path transmission. Even an approximated algorithm for these challenging problems will be very helpful in practice.

- The decoder-side learning approach can also benefit conventional source coding paradigms. For example, in state-of-the-art video coding standards, the overhead bits for motion/mode information occupy a large portion of the entire bitstream, especially for low bit-rate encoding. If we enable (even very limited) decoder-side learning in conventional video coding, the encoder will be able to skip some overhead information that are already known by the decoder, and save the overall bit-rate.

To conclude, we expect that our work will spur greater research efforts into low-complexity image/video coding. We also hope the methods we developed for decoder-side learning can also benefit conventional encoder-driven image/video codecs.

References

[1] Digital compression and coding of continuous-tone still images: Requirements and guidelines. ISO/IEC and ITU-T, Sept. 1992.

[2] Video codec for audiovisual services at $p \times 64$ kbits/s. ITU-T Recommendation, H.261 Version 2, Mar 1993.

[3] Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbits/s – part 2: Video. ISO/IEC 11172-2 (MPEG-1), Mar. 1993.

[4] MPEG-2, Information technology—Generic coding of moving pictures and associated audio information: Video. ISO/IEC 13818-2, 2d edition, 2000.

[5] Coding of audiovisual objects – part 2: Visual. ISO/IEC 14496-2 (MPEG-4), 2000.

[6] H.263, Video Coding for Low Bit Rate Communication, ITU-T Recommendation, Version 1: November 1995. Version 2: January 1998. Version 3: November 2000.

[7] JPEG 2000 Part III Final Committee Draft Version 1.0 (ISO/IEC 15444-3), ISO/IEC JTC1/SC29/WG 1, Oct. 2002.

[8] Draft ITU-T Recommendation and Final Draft International Standard, Pattaya, Thailand, 2003.

[9] A. Aaron and B. Girod, "Compression with side information using turbo codes," in *Proc. IEEE Data Compression Conference*, Snowbird, UT, Apr. 2002, pp. 252–261.

[10] A. Aaron, S. Rane and B. Girod, "Wyner-Ziv video coding with hash-based motion compensation at the receiver", *Proc. IEEE Int. Conf. Image Processing*, Singapore, Oct. 2004.

[11] A. Aaron, S. Rane, E. Setton and B. Girod, "Transform-domain Wyner-Ziv codec for video", *Proc. Visual Communications and Image Processing , VCIP-2004* , San Jose, CA, January 2004.

[12] A. Aaron, S. Rane, R. Zhang and B. Girod, "Wyner-Ziv coding for video: Applications to compression and error resilience," *Proc. IEEE Data Compression Conference, (DCC)*, Snowbird, UT, March 2003.

[13] N. Ahmed, T. Natarajan, and K. R. Rao, “Discrete Cosine Transform”, *IEEE Trans. Computers*, p.p. 90-93, Jan 1974.

[14] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” *IEEE Communications Magazine*, pp. 102-114, Aug. 2002.

[15] X. Artigas and L. Torres, “Iterative generation of motion-compensated side information for distributed video coding”, *Proc. IEEE Int. Conf. Image Processing*, Genova, Italy, Sep, 2005.

[16] J. Ascenso, C. Brites, and F. Pereira, “Motion compensated refinement for low complexity pixel based distributed video coding”, *Proc. IEEE Int. Conf. Advanced Video and Signal-Based Surveillance*, Como, Italy, Sep. 2005.

[17] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, “Performance of optical flow techniques,” *International Journal of Computer Vision*, 12: pp. 43-77, 1994.

[18] T. Berger, *Rate Distortion Theory: A Mathematical Basis for Data Compression*, Prentice-Hall, Englewood Cliffs, NJ, 1971.

[19] C. Berrou and A. Glavieux, “Near optimum error correcting coding and decoding: turbo-codes,” *IEEE Trans. Communications*, vol. 44, pp. 1261-1271, October 1996.

[20] M. Bierling, “Displacement estimation by hierarchical block matching,” *Proc. SPIE Conference on Visual Commun. Image Processing*, pp. 942-951, Cambridge, MA, Nov. 1988.

[21] S. Brofferio and F. Rocca, “Interframe redundancy reduction of video signals generated by translating objects,” *IEEE Trans. Commun.*, vol. 25, pp. 448–455, Apr. 1977.

[22] R. Buschmann, “Efficiency of displacement estimation techniques”, *Signal Processing: Image Communication*, vol. 10, pp. 43–61, 1997.

[23] N.-M. Cheung and A. Ortega, “Flexible Video Decoding: A Distributed Source Coding Approach”, *IEEE 9th Workshop on Multimedia Signal Processing*, pp. 103-106, Oct. 2007.

[24] T. Coleman, A. Lee, M. Medard, and M. Effros, “On some new approaches to practical Slepian-Wolf compression inspired by channel coding,” *Proc. IEEE Data Compression Conference 2004*, UT, Mar. 2004.

[25] G. Cook, J. Prades-Nebot, Y. Liu, and E. Delp, “Rate-distortion analysis of motion

compensated rate scalable video," *IEEE Trans. Image Proc.*, vol. 15, no. 8, pp. 2170-2190, Aug. 2006.

[26] T. Cover and J. Thomas, *Elements of Information Theory*, 2nd edition, John Wiley & Sons, Hoboken, NJ, 2006.

[27] R. Cristescu, B. Beferull-Lozano and M. Vetterli, "Networked Slepian-Wolf: theory, algorithms, and scaling laws", *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4057–4073, Dec. 2005.

[28] A. Dorgelo and H. Van der Veer, "Variable length coding for increasing traffic capacity in PCM transmission systems", *IEEE Transactions on Communications*, vol. 21, no. 12, pp. 1422-1430, Dec. 1973.

[29] C. Fenimore, "Assessment of resolution and dynamic range for digital cinema", *Proc. SPIE 15th Annual Symp. Elec. Imaging Sci. and Tech.*, Santa Clara, CA, Jan. 2003.

[30] M. Fleming and M. Effros, "Network vector quantization," in *Proc. IEEE Data Compression Conference (DCC)*, Snowbird, UT, Mar. 2001, pp. 13–22.

[31] M. Flierl, T. Wiegand, and B. Girod, "A locally optimal design algorithm for blockbased multi-hypothesis motion-compensated prediction", *Proceedings of the Data Compression Conference*, Snowbird, Utah, Apr. 1998, pp. 239–248.

[32] ——, "Rate-constrained multi-hypothesis prediction for motion compensated video compression", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, pp. 957–969, Nov. 2002.

[33] R. Gallager, *Low Density Parity Check Codes*, MIT Press, 1963.

[34] M. Gastpar, P. L. Dragotti and M. Vetterli, "The distributed Karhunen–Loève transform", *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5177-5196, Dec. 2006.

[35] I. Gilmour and R. Justin Davila, "Lossless Video Compression for Archives: Motion JPEG2k and Other Options", *Media Matters Technical Report*, Jan. 2006.

[36] F. Giorda and A. Racciu, "Bandwidth reduction of video signals via shift vector transmission," *IEEE Trans. Commun.*, vol. 23, no. 9, pp. 1002–1004, Sep. 1975.

[37] B. Girod, "The efficiency of motion-compensating prediction for hybrid coding of video sequences," *IEEE J. Sel. Areas. Commun.*, vol. 5, no. 8, pp. 1140–1154, Aug. 1987.

[38] ——, “Motion-compensating prediction with fractional-pel accuracy,” *IEEE Trans. Commun.*, vol. 41, no. 4, pp. 604–612, Apr. 1993.

[39] ——, “Efficiency analysis of multihypothesis motion-compensated prediction for video coding”, *IEEE Trans. Image Proc.*, vol. 9, no. 2, pp. 173–183, Feb. 2000.

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

[41] C. Guillemot, F. Pereira, L. Torres, T. Ebrahimi. R. Leonardi, J. Ostermann, “Distributed Monoview and Multiview Video Coding: Basics, Problems and Recent Advances”, *IEEE Signal Processing Magazine, Special Issue on Signal Processing for Multiterminal Communication Systems*, September, 2007.

[42] Mei Guo, Yan Lu, Feng Wu, Debin Zhao, Wen Gao, “Wyner-Ziv switching scheme for multiple bit-rate video streaming”, *IEEE transaction on Circuits and Systems for Video Technology*, vol. 18, no. 5, pp 569-581, 2008.

[43] H.-M. Hang, Y.-M. Chou, and S.-Chih. Cheng, “Motion estimation for video coding standards,” *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, pp.113-136, Nov. 1997.

[44] D. He, A. Jagmohan and L. Lu, “Seure collaboration using Slepian-Wolf codes,” *Proc. International Conference on Image Processing (ICIP)*, San Diego, Oct. 2008.

[45] T. Hidaka, “Description of the proposing algorithm and its score for moving image (A part of the proposal package),” Int. Standards Org./Int. Electrotech. Comm. (ISO/IEC) JTC 1, ISO/IEC JTC 1/SC 2/WG 8 MPEG 89/188, Oct. 1989.

[46] B. K. Horn and B. G. Schunck, “Determining optical flow,” *Artificial Intelligence*, vol. 17, 1981.

[47] J. Ive, “Image formats for HDTV,” *EBU Technical Review*, July 2004.

[48] J. R. Jain and A. K. Jain, “Displacement measurement and its application in interframe image coding.” *IEEE Trans. Commun.*, vol.29, pp. 1799-1808, Dec. 1981.

[49] N. Jayant and P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video*, Englewood Cliffs, NJ: Prentice-Hall, 1984.

[50] M. Johnson, P. Ishwar, V. M. Prabhakaran, D. Schonberg, and K. Ramchandran,

"On compressing encrypted data," *IEEE Trans. Signal Processing*, vol. 52, no. 10, Oct. 2004.

[51] M. Karczewicz and R. Kurceren, "The SP- and SI-Frames Design for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, issue 7, pp. 637-644, July, 2003.

[52] K. Karhunen, "Über lineare Methoden in derWahrscheinlichkeitsrechnung," *Ann. Acad. Sci. Fenn., Ser. A.1.: Math.-Phys.*, vol. 37, pp. 3–79, 1947.

[53] T. Koga, K. Iinuma, A. Hirano, Y. Iijima and T. Ishiguro, "Motion-compensated interframe coding for video conferencing," *Proc. Nat. Telecommun. Conf.*, pp. 531-G535, New Orleans, LA, Nov. 1981.

[54] C. Kuglin and D. Hines. "The phase correlation image alignment method," *Proc. IEEE Int. Conf. Cybern. Soc.*, pp. 163-165, 1975.

[55] O. Lee and Y. Wang, "Motion compensated prediction using nodal based deformable block matching," *Journal of Visual Communications and Image Representation*, vol.6, pp. 26-34, Mar. 1995.

[56] H. Li and R. Forchheimer, "A transformed block-based motion compensation technique", *IEEE Transactions on Communications*, vol. 43, pp. 1673–1676, Feb. 1995.

[57] Z. Li, *New methods for motion estimation with applications to low complexity video compression*, Ph.D. dissertation, Purdue Univ., West Lafayette, IN, 2005.

[58] Z. Li and E. J. Delp, "Wyner-Ziv side estimator: conventional motion search methods revisited", *Proc. IEEE Int. Conf. Image Processing*, Genoa, Italy, Sep. 2005.

[59] Z. Li, L. Liu and E. J. Delp, "Rate Distortion Analysis of Motion Side Estimation in Wyner–Ziv Video Coding", *IEEE Trans. Image Processing*, vol. 16, no. 1, pp. 98–113, Jan. 2007.

[60] Y.-C. Lin, D. Varodayan, T. Fink, E. Bellers and B. Girod, "Localization of tampering in contrast and brightness adjusted images using distributed source coding and expectation maximization", *Proc. International Conference on Image Processing (ICIP)*, San Diego, Oct. 2008.

[61] P. List, A. Joch, J. Lainema, G. Bjotegaard, and M. Karczewicz, "Adaptive deblocking filter," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 614-619, Jul. 2003.

[62] L. Liu, Z. Li, and E. Delp, " Backward Channel Aware Wyner-Ziv Video Coding", *IEEE International Conference on Image Processing*, Sept. 2006.

[63] W. Liu, “Data hiding in JPEG 2000 code streams”, *Proc. IEEE International Conference on Image Processing (ICIP)*, Oct. 2004.

[64] W. Liu, L. Dong and W. Zeng, “Optimum detection for image-adaptive watermarking in DCT domain”, *Proc. IEEE International Conference on Image Processing (ICIP)*, Atlanta, GA, Oct. 2006.

[65] ——, “Optimum detection for spread-spectrum watermarking that employs self-masking”, *Proc. IEEE International Conference on Image Processing (ICIP)*, San Antonio, TX, Oct. 2007.

[66] ——, “Optimum detection for spread-spectrum watermarking that employs self-masking”, *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 4, pp. 645-654, Dec. 2007.

[67] ——, “Wyner-Ziv video coding with multi-resolution motion refinement: Theoretical analysis and practical significance”, *(invited paper) Visual Communications and Image Processing (VCIP)*, San Jose, CA, Jan. 2008.

[68] ——, “Power-efficient rate allocation for Slepian-Wolf coding over wireless sensor networks”, *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Las Vegas, NV, Mar. 2008.

[69] ——, “Optimum Power-efficient rate allocation for Slepian-Wolf coding over wireless sensor networks”, *(submitted to) IEEE Transactions on Information Theory*.

[70] ——, “Multi-resolution motion refinement with extensive motion exploration for Wyner-Ziv video coding”, *(submitted to) IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP),* 2009.

[71] W. Liu, Y. Guo and W. Zheng, “Inter-frame image enhancement for motion JPEG 2000”, *Proc. SPIE Conference on Visual Communications and Image Processing (VCIP)*, Jul. 2003.

[72] W. Liu and W. Zeng, “Non-binary distributed source coding using gray codes”, *Proc. IEEE International Workshop on Multimedia Signal Processing (MMSP)*, Shanghai, China, Oct. 2005.

[73] W. Liu, W. Zeng, L. Dong and Q. Yao, “Resolution-progressive compression of encrypted grayscale images”, *Proc. IEEE International Conference on Image*

Processing (ICIP), San Diego, CA, Oct. 2008.

[74] Y. Liu, P. Salama, Z. Li, and E. Delp, “An enhancement of leaky prediction layered video coding,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 11, pp. 1317–1331, Nov. 2005.

[75] A. Liveris, Z. Xiong, and C. Georghiades, “Compression of binary sources with side information at the decoder using LDPC codes,” *IEEE Communications Letters*, vol. 6, no. 10, pp. 440–442, Oct. 2002.

[76] B. Macchiavello, R. L. de Queiroz and D. Mukherjee, "Motion-based side-information generation for a scalable Wyner-Ziv video coder," *IEEE International Conference on Image Processing (ICIP)*, 2007.

[77] S.G. Mallat, “A theory for multiresolution signal decomposition: the wavelet representation”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, July 1989.

[78] P. Mitran and J. Bajcsy, “Near Shannon-limit coding for the Slepian-Wolf problem,” in *Proc. Biennial Symposium on Communications*, Kingston, Ontario, June 2002.

[79] ——, “Coding for the Wyner-Ziv problem with turbo-like codes,” in *Proc. IEEE International Symposium on Information Theory*, Lausanne, Switzerland, June 2002, p. 91.

[80] A. Murat Tekalp, *Digital Video Processing*, Prentice Hall, NJ, 1995.

[81] H. H. Nagel and W. Enklemann, “An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol.8, pp. 565-593, Sept. 1986.

[82] Y. Nakaya and H. Harashima, “Motion compensation based on spatial transformations”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 4, no. 3, pp. 339–356, June 1994.

[83] S. Nogaki and M. Ohta, “An overlapped block motion compensation for high quality motion picture coding”, *Proc. IEEE International Symposium on Circuits and Systems*, May 1992, pp. 184–187.

[84] M. T. Orchard and G. J. Sullivan, “Overlapped block motion compensation: An estimation-theoretic approach,” I*EEE Trans. Image Process.*, vol. 3, no. 5, pp. 693–699, Sep. 1994.

[85] H.-W. Park and H.-S. Kim, “Motion estimation using low-band-shift method for wavelet-based moving-picture coding”, *IEEE Trans. Image Processing*, vol. 9, no. 4, pp. 577–587, 2000.

[86] J. Prades-Nebot, G. Cook, and E. J. Delp, “An analysis of the efficiency of different SNR-scalable strategies for video coders,” *IEEE Trans. Image Process.*, vol. 15, no. 4, pp. 848–864, Apr. 2006.

[87] S. S. Pradhan and K. Ramchandran, “Distributed source coding using syndromes (DISCUS)”, *IEEE Transactions on Information Theory*, vol. 49, no. 3, March 2003.

[88] R. Puri and K. Ramchandran, “Prism: An uplink-friendly multimedia coding paradigm”, *Proc. IEEE Int. Conf. Image Processing*, Barcelona, Spain, Sep. 2003.

[89] S. Rane, P. Baccichet and B. Girod, “Modeling and optimization of a systematic lossy error protection system based on H.264/AVC redundant slices,” *Proc. Picture Coding Symposium*, *PCS-2006,* Beijing, China, April 2006.

[90] D. Rebollo-Monedero, R. Zhang, and B. Girod, “Design of optimal quantizers for distributed source coding,” in *Proc. IEEE Data Compression Conference (DCC)*, Snowbird, UT, Mar. 2003, pp. 13–22.

[91] D. N. Rowitch and L. B. Milstein, “On the performance of hybrid FEC/ARQ systems using rate-compatible punctured turbo (RCPT) codes,” *IEEE Trans. Commun.*, vol. 48, pp. 948–959, June 2000.

[92] D. Schonberg, *Practical distributed source coding and its application to the compression of encrypted data*, Ph.D dissertation, Univ. of California, Berkeley, CA, 2007.

[93] H. Schwarz, D. Marpe, and T. Wiegand, “Comparison of MCTF and closed-loop hierarchical B pictures”, ISO/IEC JTC1/SC29/WG11, Doc. JVT-P059, Poznan, Poland, July 2005.

[94] V. Seferidis and M. Ghanbari, “General approach to block matching motion estimation,” *Optical Engineering*, vol.32, pp. 1464-1474, July 1993.

[95] E. Setton and B. Girod, “Rate-distortion analysis and streaming of SP and SI frames”, *IEEE Trans. Circ. Sys. for Video Tech.*, vol. 16, no. 6, June, 2006.

[96] C. E. Shannon, “Coding theorems for a discrete source with a fidelity criterion,” *IRE Nat. Conv. Rec.*, pt. 4, pp. 142–163, 1959.

[97] J. D. Slepian and J. K. Wolf, “Noiseless coding of correlated information sources,” *IEEE Transactions on Information Theory*, vol. IT-19, pp. 471–480, July 1973.

[98] W. Stallings, *Cryptography and Network Security: Principles and Practice* (3rd Edition), ISBN: 0130914290, Prentice Hall, 2003.

[99] V. Stanković, A. Liveris, Z. Xiong, and C. Georghiades, “Design of Slepian-Wolf Codes by Channel Code Partitioning”, *Proc. IEEE Data Compression Conference 2004*, Snowbird, UT, Mar. 2004.

[100] C. Stiller and J. Konrad, “Estimating motion in image sequences,” *IEEE Signal Processing Magazine*, vol. 16, pp. 70-91, July 1999.

[101] G. J. Sullivan, “Multi-hypothesis motion compensation for low bitrate video coding,” *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, 1993, pp. 437–440.

[102] G. J. Sullivan and R. L. Baker, “Rate-distortion optimized motion compensation for video compression using fixed or variable size blocks,” *Proc. IEEE Global Telecommunications Conf. (GLOBECOM)*, 1991, pp. 85–90.

[103] G. J. Sullivan and T. Wiegand, “Rate-distortion optimization for video compression”, *IEEE. Signal Proc. Magazine*, pp. 74-90, Nov. 1998.

[104] ——, “Video Compression—From Concepts to the H.264/AVC Standard”, *Proc. IEEE*, vol. 93, no. 1, pp. 18-31, Jan. 2005.

[105] Y. Wang, J. Ostermann and Y.-Q. Zhang, *Video Processing and Communications*, Prentice Hall, NJ, 2001.

[106] M. Weinberger, G. Seroussi, and G. Sapiro, “The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS”, *IEEE Trans. Image Proc.*, vol. 9, no. 8, pp. 1309–1324, Aug. 2000.

[107] T. Wiegand, N. Färber, K. Stuhlmüller, and B. Girod, “Error-resilient video transmission using long-term memory motion-compensated prediction,” *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 1050–1062, Jun. 2000.

[108] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, “Rate-constrained coder control and comparison of video coding standards,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 688–703, Jul. 2003.

[109] X. Wu and N. Memon, “Context-based adaptive lossless image coding,” *IEEE*

Trans. Commun., vol. 45, pp. 437–444, Apr. 1997.

[110] A. D. Wyner, “Recent Results in the Shannon Theory,” *IEEE Transactions on Information Theory*, vol. 20, no. 1, pp. 2–10, Jan. 1974.

[111] 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. IT-22, no. 1, pp. 1–10, Jan. 1976.

[112] ——, “The rate-distortion function for source coding with side information at the decoder—II: General sources,” *Information and Control*, vol. 38, no. 1, pp. 60–80, July 1978.

[113] G. Wyszecki and W. S. Stiles, *Color Science,* John Wiley, New York, 1967.

[114] D. Varodayan, A. Aaron, and B. Girod, “Rate-Adaptive Distributed Source Coding using Low-Density Parity-Check Codes”, *Proc. Asilomar Conf. Signals, Syst., Comput*., Pacific Grove, CA, Nov. 2005.

[115] ——, "Exploiting spatial correlation in pixel-domain distributed image compression," *Proc. Picture Coding Symposium*, PCS-2006, Beijing, China, April 2006.

[116] D. Varodayan, D. Chen, M. Flierl and B. Girod, “Wyner-Ziv coding of video with unsupervised motion vector learning,” *EURASIP Signal Processing: Image Communication Journal, Special Issue on Distributed Video Coding*, vol. 23, no. 5, pp. 369-378, June 2008.

[117] R. Zamir, S. Shamai, and U. Erez, “Nested linear/lattice codes for structured multi-terminal binning,” *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1250–1276, June 2002.

[118] Z. Xiong, A. Liveris, and S. Cheng, “Distributed source coding for sensor networks,” *IEEE Signal Processing Magazine*, vol. 21, pp. 80-94, September 2004.

[119] Z. Xiong, A. Liveris, S. Cheng, and Z. Liu, “Nested quantization and Slepian-Wolf coding: A Wyner-Ziv coding paradigm for i.i.d. sources,” in *Proc. IEEE Workshop on Statistical Signal Processing (SSP)*, St. Louis, MO, Sept. 2003.

[120] Q. Xu and Z. Xiong, “Layered Wyner-Ziv video coding,” *IEEE Trans. Image Processing*, vol. 15, pp. 3791-3803, December 2006.

[121] Q. Yao, W. Zeng, and W. Liu, "Multi-resolution based hybrid spatiotemporal compression of encrypted videos," *(submitted to) IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP),* 2009.

[122] Y. Zhao and J. Garcia-Frias, "Joint estimation and data compression of correlated non-binary sources using punctured turbo codes," in *Proc. Conference on Information Sciences and Systems*, Princeton, NJ, Mar. 2002.

[123] Y. Zhu, W. Liu, L. Dong, W. Zeng and H. Yu, "High performance adaptive video services based on bitstream switching for IPTV systems", *(accepted by) IEEE Consumer Communications and Networking Conference (CCNC)*, 2009.

VITA

Wei Liu received his bachelor and master degrees from Tsinghua University, Beijing, China, in 1999 and 2002, respectively, both in Electrical Engineering. From 2004 to 2008, he studied in the Computer Science Department, University of Missouri, under the advice of Prof. Wenjun Zeng, and was granted his Ph.D. degree in 2008.

From 2002 to 2004, he was a research engineer with Panasonic Research and Development Center, Beijing, China. He was a summer intern with the Thomson Corporate Research, Princeton, NJ, in 2006. He was a recipient of the Gilliom Cyber Security Fellowship in 2007 and 2008.

His research interests include multimedia communication and security, image/video coding and watermarking, and distributed source coding.