UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

ADAPTIVE AND SECURE DISTRIBUTED SOURCE CODING FOR VIDEO AND IMAGE COMPRESSION

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

DOCTOR OF PHILOSOPHY

By

LIJUAN CUI
Norman, Oklahoma
2013

ADAPTIVE AND SECURE DISTRIBUTED SOURCE CODING
FOR VIDEO AND IMAGE COMPRESSION

A DISSERTATION APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY

Dr. Samuel Cheng, Chair

Dr. Pramode Verma

Dr. James Sluss

Dr. J. R. Cruz

Dr. William Ray

Acknowledgments

I would like to take the opportunity to thank my advisor, Dr. Samuel Cheng for this enthusiastic guidance and support in my research work and career development. Dr. Cheng is a great advisor. Without his constant encouragement and support, I would not be able to finish this dissertation. I can never thanks Dr. Cheng enough for his help in navigating me towards my educational and career goals. I would also like to take this opportunity to thank Dr. James Sluss, Dr. Pramode Verma, Dr. William Ray and Dr. J.R. Cruz for serving on my committee and providing their suggestions and guidances for this dissertation.

I would express my sincere gratitude to all those whom I have worked with in the past years. I am grateful to Dr. Xiaoqian Jiang from University of California, San Diego, Dr. Lina Stankovic and Dr. Vladimir Stankovic from the University of Strathclyde, Dr. Xianbo Chen from Broadcom Corporation. I have learned a lot from the collaborations with them about data privacy, distributed source coding, firmware development. Moreover, I am also grateful to Dr. Lih-feng Tsaur, Dr. Chikan Kwan, Xin Tian, Arthur Chen, et. al. with whom I worked during the Spring 2013 at Broadcom. Their insightful suggestions help me to work out problems during the intern.

I am grateful to all my Lab mates in the research group of Dr. Samuel Cheng and classmates in OU-Tulsa TCOM, Feng Chen, Mouhammad Al-Akkoumi, Amin, Nafise, Shuang Wang, Wen Gu, et. al.

My deepest gratitude is for my family. I would like to thank my parents, Kaoliang Cui and Xiurong Du. They raised me, loved me and supported me. In addition, I do appreciate my husband, Shuang Wang for his accompanies, helps and supports in all the past years. To them, I dedicate this dissertation.

Table of Contents

Chapter **Page**

List of Tables

Table **Page**

List of Figures

Figure **Page**

Abstract

Distributed Video Coding (DVC) is rapidly gaining popularity as a low cost, robust video coding solution, that reduces video encoding complexity. DVC is built on Distributed Source Coding (DSC) principles where correlation between sources to be compressed is exploited at the decoder side. In the case of DVC, a current frame available only at the encoder is estimated at the decoder with side information (SI) generated from other frames available at the decoder. The inter-frame correlation in DVC is then explored at the decoder based on the received syndromes of Wyner-Ziv (WZ) frame and SI frame. However, the ultimate decoding performances of DVC are based on the assumption that the perfect knowledge of correlation statistic between WZ and SI frames should be available at decoder. Therefore, the ability of obtaining a good statistical correlation estimate is becoming increasingly important in practical DVC implementations.

Generally, the existing correlation estimation methods in DVC can be classified into two main types: online estimation where estimation starts before decoding and on-the-fly (OTF) estimation where estimation can be refined iteratively during decoding. As potential changes between frames might be unpredictable or dynamical, OTF estimation methods usually outperforms online estimation techniques with the cost of increased decoding complexity.

In order to exploit the robustness of DVC code designs, I integrate particle filtering with standard belief propagation decoding for inference on one joint factor graph to estimate correlation among source and side information. Correlation estimation is performed OTF as it is carried out jointly with decoding of the graph-based DSC code. Moreover, I demonstrate our

proposed scheme within state-of-the-art DVC systems, which are transform-domain based with a feedback channel for rate adaptation. Experimental results show that our proposed system gives a significant performance improvement compared to the benchmark state-of-the-art DISCOVER codec (including correlation estimation) and the case without dynamic particle filtering tracking, due to improved knowledge of timely correlation statistics via the combination of joint bit-plane decoding and particle-based BP tracking.

Although sampling (e.g., particle filtering) based OTF correlation advances performances of DVC, it also introduces significant computational overhead and results in the decoding delay of DVC. Therefore, I tackle this difficulty through a low complexity adaptive DVC scheme using the deterministic approximate inference, where correlation estimation is also performed OTF as it is carried out jointly with decoding of the factor graph-based DVC code but with much lower complexity. The proposed adaptive DVC scheme is based on expectation propagation (EP), which generally offers better tradeoff between accuracy and complexity among different deterministic approximate inference methods. Experimental results show that our proposed scheme outperforms the benchmark state-of-the-art DISCOVER codec and other cases without correlation tracking, and achieves comparable decoding performance but with significantly low complexity comparing with sampling method.

Finally, I extend the concept of DVC (i.e., exploring inter-frames correlation at the decoder side) to the compression of biomedical imaging data (e.g., CT sequence) in a lossless setup, where each slide of a CT sequence is analogous to a frame of video sequence. Besides compression efficiency, another important concern of biomedical imaging data is the privacy and security. Ideally, biomedical data should be kept in a secure manner (i.e.

encrypted). An intuitive way is to compress the encrypted biomedical data directly. Unfortunately, traditional compression algorithms (removing redundancy through exploiting the structure of data) fail to handle encrypted data. The reason is that encrypted data appear to be random and lack the structure in the original data. The "best" practice has been compressing the data before encryption, however, this is not appropriate for privacy related scenarios (e.g., biomedical application), where one wants to process data while keeping them encrypted and safe. In this dissertation, I develop a Secure Privacy-presERving Medical Image CompRessiOn (SUPERMICRO) framework based on DSC, which makes the compression of the encrypted data possible without compromising security and compression efficiency. Our approach guarantees the data transmission and storage in a privacy-preserving manner. I tested our proposed framework on two CT image sequences and compared it with the state-of-the-art JPEG 2000 lossless compression. Experimental results demonstrated that the SUPERMICRO framework provides enhanced security and privacy protection, as well as high compression performance.

CHAPTER 1

INTRODUCTION

In modern life, digital videos have been applied in a lot of applications, such as, broadcasting, video streaming, video delivery by mobile telephones. However, digital video will need huge space for storage and wide bandwidth for transmission. In order to maintain the same video quality while reducing the data rate of video signal, video compression technologies provide an important role. Generally, a video coding system includes an encoder and a decoder, which is also referred to as codec. In above one-to-many applications, video is encoded once and decoded million times at the user side, which means that multiple lightweight decoders are necessary for a video coding system. Currently, the most popular video coding standard for above applications is H.264 Advanced Video Codec (AVC), which is relies on the powerful hybrid block-based motion compensation and DCT transform architecture. The main complexity existed at the encoder of H.264 standard is introduced by the motion estimation, which is mainly used to explore the spacial and temporal redundancy in the video sequence. Since adjacent frames are explicitly available at the encoder, the encoder can sufficiently extract the spacial and temporal correlation among frames, which provides a strong guarantee of coding performance.

Recently, several many-to-one setups are becoming more and more popular, such as video surveillance with tiny cameras and cell-to-cell communications, where there are millions of encoders while only a few of decoders. More important, since most of these devices would be powered by batteries, it is extremely crucial to restrict the encoding complexity to extend the limited

battery life. Driven by these emerging applications, the industry is anxious for an entirely new coding paradigm, which could significantly reduce the encoding complexity, even though the expense is to increase decoding complexity. Although the traditional video coding standard (e.g., H.264) is very promising in a centralized setup, it could not be easily tailored to fit such new coding paradigm. Fortunately, distributed video coding (DVC) [1, 2] provides a workaround for these difficulties, where the complexity could be significantly shifted from the encoder side to the decoder side. The DVC technique is based on distributed source coding (DSC) principle brought a paradigm shift from the conventional centralized video coding architecture to a totally distributed manner, where the computationally-expensive motion compensation and correlation extraction procedures will be done at the decoder side.

1.1 Overview of DSC and DVC

From the information theory perspective, DSC refers to separate compression and joint decompression of multiple correlated sources. DSC started as an information-theoretical problem in the renowned 1973 paper of Slepian and Wolf [3]. Slepian and Wolf considered the lossless compression of two physically separated sources, and demonstrated that, roughly speaking, there is no performance loss compared to joint compression as long as joint decompression is performed. In 1976, Wyner and Ziv [4] considered a lossy version (i.e., with a distortion constraint) of the asymmetric Slepian-Wolf (SW) problem known as Wyner-Ziv (WZ) coding, where one source is available at the decoder as side information (SI) (e.g, through entropy coding). Wyner and Ziv showed that for some particular correlation models (e.g. Gaussian, Laplace,

etc.), there is no performance loss due to the absence of SI at the encoder.

DVC exploits WZ coding principles by performing computationally expensive motion compensation at the decoder instead of at the encoder. The beauty of WZ coding (DSC in general) is that there is no need for the encoder to be aware of the SI, which makes it possible to accomplish predictive coding without encoder motion compensation. In a nutshell, a block of pixels/coefficients in a certain video frame (a.k.a., WZ frame) could be efficiently WZ encoded into a stream (i.e., syndromes) without any reference to any other video frames. To recover a WZ frame at the decoder side, a SI frame will be first generated based on the received key frames through motion compensation, where key frames can be decoded independent of other frames. Then the WZ decoder could decompress the WZ frame based on the received syndromes and the generated SI through DSC principle. The state-of-the-art WZ coding designs based on turbo, LDPC, and other graph-based codes have been widely used in DVC studies (see [5–7] and references therein). Since DVC is based on the DSC principle, the next section will introduce some theoretical backgrounds about DSC, especially SW and WZ coding.

1.2 Theoretical Background: Slepian-Wolf and Wyner-Ziv

1.2.1 Slepian-Wolf Coding Theorem for Lossless Compression

Slepian and Wolf proved a very surprising result that generally speaking, it is possible to have no performance loss compared to the case when joint encoding is allowed [3]. For a source pair X_1 and X_2 as shown in Fig. 1.1, let R_X and R_Y be the corresponding compression rates. They show that lossless

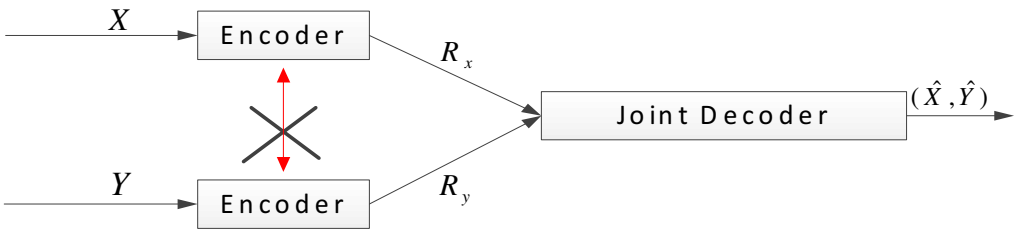


Figure 1.1: Distributed source coding framework.

compression is possible if and only if

$$R_X \geq H(X_1|X_2), R_Y \geq H(X_2|X_1), \text{and} R_X + R_Y \geq H(X_1, X_2). \tag{1.1}$$

For example, in the asymmetric SW setup, the source X_1 could be compressed independently with the code rate at $R_X = H(X_1)$ in theory. According to the Slepian-Wolf Theorem, it is sufficient to have $R_Y = H(X_2|X_1)$ to achieve a lossless compression for source Y, which results a total rate as $R_X + R_Y = H(X_1) + H(X_2|X_1) = H(X_1, X_2)$. This is exactly the same as the rate required even when joint compression is allowed! I usually refer to this no-performance-loss feature as no rate loss.

1.2.2 Wyner-Ziv Coding Theorem for Lossy Compression

WZ coding [4] extends the asymmetric SW setup in which coding of the source X is lossy with respect to a fidelity criterion rather than lossless. WZ coding can be treated as a degenerated case of DSC with two sources X and Y, where source Y is transmitted perfectly to the decoder and source X is quantized first then followed by the SW coding. The rate distortion function for this setup $R_{WZ}(D)$ can be expressed as follows,

$$R_{WZ}(D) = \inf I(U;X) - I(U;Y), \tag{1.2}$$

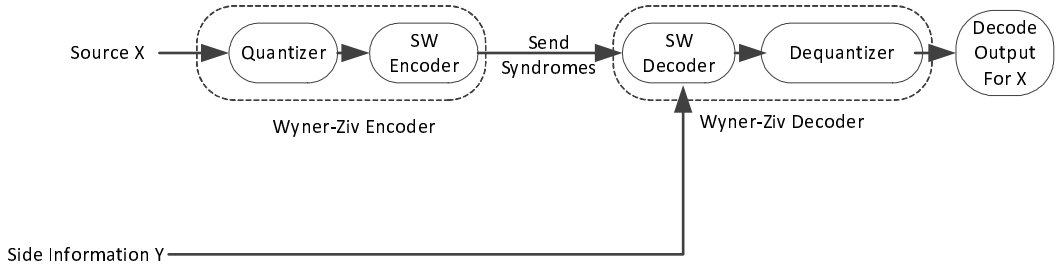


Figure 1.2: Wyner-Ziv coding framework.

where U is an auxiliary random variable satisfying the Markov chain $Y \leftrightarrow X \leftrightarrow U$ and there exists a function $\hat{X} = \hat{X}(U, S)$ satisfying $E\left\{d\left(X, \hat{X}\right)\right\} \leq D$. Here $d(*, *)$ can be any distortion metric. Moreover, when the SI is available at both encoder and decoder, the rate-distortion function is

$$R_{X|Y}(D) = \inf_{\hat{X} \in X : Ed(X,\hat{X}) \leq D} I(X; \hat{X} \mid Y), \tag{1.3}$$

In general, there is a rate loss with WZ coding that $R_{WZ}(D) \geq R_{X|Y}(D)$. However, if the sources are jointly Gaussian and mean square difference is taken as the distortion measure (quadratic Gaussian case), there is no rate loss as in the lossless (Slepian-Wolf) case, that is $R_{WZ}(D) = R_{X|Y}(D)$.

1.3 Wyner-Ziv Video Coding Framework

1.3.1 Practical Wyner-Ziv Coding Design

SW coding was proposed by Slepian and Wolf in 1973 [3] for losslessly compressing two physically separated sources and jointly decompressing them. Wyner is the first one who realized that by taking syndromes as the compressed sources, error-correcting parity check codes can be used to implement SW coding [8]. The approach was rediscovered and popularized by Pradhan *et al.* more than two decades later [9]. Then, numerous channel coding based

SW coding schemes have been proposed in [9–11].

WZ source coding, a.k.a. the lossy version of SW coding, is usually realized by quantization followed by SW coding of the quantized indices based on channel coding [12], as shown in Fig. 1.2. Moreover, quantization can be used to tune rate-distortion performance. At the decoder side, the quantized indices can be recovered using the SW decoder. Then a minimum-distortion dequantizer is used to reconstruct an estimate of the original source [9, 12].

Practical WZ code design based on quantization plus conventional channel codes is possible since correlation between the source and SI is treated as a virtual communication channel which can be expressed in the form $X = Y + N$, where X is the source to be recovered defined as the sum of the SI Y and noise N. As long as this virtual channel can be modeled by some standard communication channel, e.g., Gaussian, channel codes can be effectively employed. Designs [12, 13] based on trellis-coded quantization followed by advanced channel coding, e.g., turbo codes and low-density parity-check (LDPC) codes come very close to the bounds for two jointly Gaussian sources.

1.3.2 Basic Wyner-Ziv video Coding Architecture

WZ video coding is the evolution of SW and WZ coding theories on the video coding applications. For WZ video coding, the frames in a video sequence are firstly divided into key frames and WZ frames. WZ frames corresponding to source X are encoded by the WZ encoder, which includes two key steps, i.e., quantization and SW encoding. Moreover, for transform domain video coding, the codec requires an additional transform module (e.g. DCT transform). Key frames are encoded by traditional video coding technique, e.g.

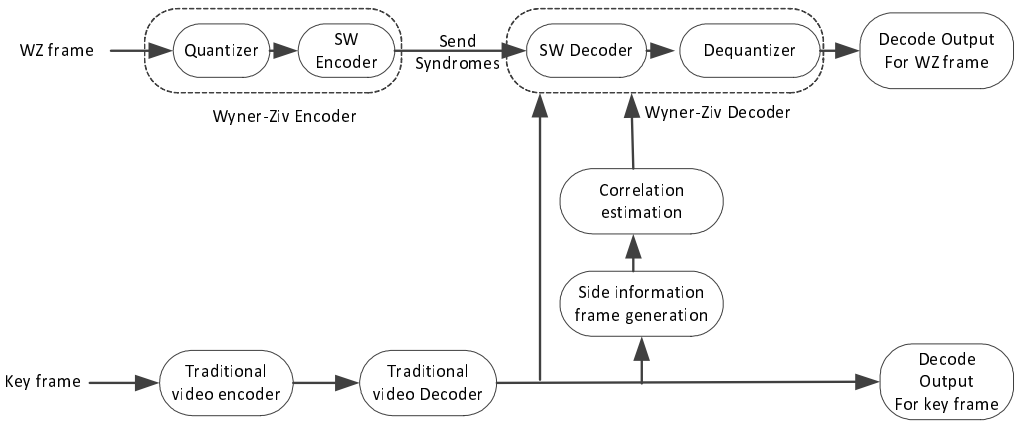


Figure 1.3: Block diagram of WZ video coding.

H.264, and decoded at the decoder side independently, which later will be used to generate SI Y. Unlike ordinary WZ coding, WZ video coding architecture has one more frame interpolation module for generating a high quality SI frame. Moreover, in ordinary WZ coding, the correlation between source and SI is usually assumed as a known priori. However, in practical WZ video coding applications, a robust WZ codec usually requires a correlation estimation module at the decoder side, since the correlation dynamically changes with the scene in the video sequence. The aforementioned two modules (i.e., SI generation and correlation estimation) are two key factors, which have significant impacts on the compression performance of WZ video coding.

The basic WZ video coding procedures illustrated in Fig. 1.3 can be summarized as follows:

1. Each WZ frame is first quantized and represented by quantization indices.

2. The same significant bits in the quantization indices are grouped together as a source vector and encoded by a SW encoder. Then the resulting parity bits are sent to the decoder.

3. Key frames are encoded by the traditional video coding, e.g. H.264, and decoded at the decoder side independently. Decoded key frames can be used to generate an initial estimate or an "errored "version of WZ frame, which is treated as SI Y available at the decoder, through motion estimation and frame interpolation.

4. At the WZ video decoder, the quantized symbol stream is decoded through joint decoding with the aid of generated SI frame Y and the estimated correlation between frames X and Y.

5. Finally, the decoded quantization indices can be reconstructed into WZ frame X through dequantization step.

In addition, for WZ video coding, there are two interesting schemes: pixel-domain and transform domain WZ codecs. Pixel domain WZ codec is the simplest WZ coding scheme, where pixels are treated as the sources directly in the WZ codec. However, to achieve a higher compression performance, WZ codes is usually working on the transform domain (e.g. DCT domain). Transform domain WZ codec is an extension of pixel domain WZ video codec by exploring the redundancy in frequency domain. In transform domain WZ coding, a blockwise DCT is applied on WZ frames and SI frames, respectively, generating different frequency bands. For example, a 4×4 DCT transform will generate 16 frequency bands, where the DC band usually contains the most important information of the WZ frame. In contrast, a higher frequency AC band would contain a less amount of information. In general, only the DC band and the first few AC bands are kept, while other AC bands are discard. Then the coefficients of remaining DCT bands can be treated as distributed source for WZ video coding. The number of remaining DCT

bands provides a trade-off between video quality and compression efficiency.

1.4 The Dissertation Contributions

A key difference between conventional WZ coding and DVC is that the correlation statistics among sources in the former case is usually assumed to be known as a constant at both the encoder and decoder. In DVC, however, such assumption is normally far-fetched, as correlation statistics between WZ and SI frames would be unknown and dynamically change over time and the location of pixels/coefficients, no matter how well SI frame is generated. Indeed, due to the non-stationarity of real scenes, WZ coding in DVC has to deal with varying correlation noise statistics. Therefore, estimating correlation statistics has been identified as one of key challenges in DVC.

Moreover, I extend the concept of DVC (i.e., exploring inter-frames correlation at the decoder side) to the compression of biomedical imaging data (e.g., CT sequence) in a lossless setup, where each slide of a CT sequence is analogous to a frame of video sequence. Besides compression efficiency, another important concern of biomedical imaging data is the privacy and security. Ideally, biomedical data should be kept in a secure manner (i.e. encrypted). In this dissertation, I develop a Secure Privacy-presERving Medical Image CompRessiOn (SUPERMICRO) framework based on DSC, which makes the compression of the encrypted data possible without compromising security and compression efficiency.

This dissertation's contributions can thus be summarized as:

1. Introduce OTF correlation noise estimation within the SW decoding process that takes into account both source and side-information statistics.

2. Construct a factor graph with connected regions incorporating joint bit-plane SW decoding and correlation variable nodes for correlation estimation to adaptively capture correlation statistics for different video sequences.

3. Modify and implement belief propagation inference algorithm applied over the overall graph which works jointly with particle filtering on particles tied to the correlation variable nodes, to successively refine estimation of the correlation noise and source iteratively until convergence is reached. Statistics of the side information is inherently captured.

4. Incorporate the proposed joint bit-plane decoding with correlation estimation design into a transform-domain state-of-the-art DVC.

5. To tackle the complexity issue within the sampling method, the proposed EP based OTF correlation estimation significantly reduced the computational complexity.

6. The proposed SUPERMICRO framework employs a joint bit-plane decoder based on a factor graph which can handle the spatial correlation between adjacent CT image slices at the pixel level, as opposed to previous work [14] performing decompression on each bit-plane (i.e., subframe) separately. As a result, SUPERMICRO preserves the important inter-bit-plane correlation without sacrificing privacy and security.

7. The 2D Markov based symmetric correlation model in [15] can not capture the inter-pixel correlation within each bit-plane very well. The proposed SUPERMICRO framework explores the spatial correlation between adjacent slices at the pixel level using a more realistic Lapla-

cian correlation model [16–18].

8. Since the DSC encoder has no access to the original data, it is short of the knowledge of compression rate for the encrypted data. The assumption of known transmission rate is usually infeasible in practical applications. To tackle this difficulty, I incorporate a Low-Density Parity-Check Accumulate (LDPCA) based SW encoding for rate adaptive decoding with a feedback channel in our proposed framework.

1.5 The Organization of The dissertation

The rest of this dissertation is organized as follows. The SI frame generation and the transitional correlation estimation (i.e., online estimation) schemes in DVC will be presented in Chapter 3. In Chapter 4, I will describe our proposed OTF correlation estimation as an extension of the existing estimation algorithms based on sampling method (i.e., particle filtering (PF)). To reduce the estimation complexity, I introduce EP based OTF estimator in Chapter 5. By extending the concept of DVC (i.g., exploring inter-frame correlations at the decoder side), I propose a secure DSC algorithm for privacy-preserving medical imaging data (e.g., CT sequence) compression in Chapter 6. Finally, I will draw the conclusion in Chapter 7.

Research in the dissertation has been published in several international journals and conferences. In Chapter 4, the work of PF based correlation estimators has been published in IEEE Transactions on Communications [19,20], IEEE Transactions on TCSVT [21], IEEE Transactions on Image Processing [22] and conferences of [23–27]. In Chapter 5, the studies of EP based correlation estimators have been published a part of IEEE Communications Letter [28] and conference papers in IEEE GLOBECOM 2011 [29] and SPIE

2012 [30]. Furthermore, the proposed framework in Chapter 6 was published in IEEE 2012 HISB conference [31].

CHAPTER 2

DVC WITH SI FRAME GENERATION AND CORRELATION ESTIMATION

The state-of-the-art of DVC based on the DSC principle which highly relies on the quality of SI and correlation model. In traditional DSC, SI can be directly obtained from one of the distributed sources. In DVC framework, although one can still use the adjacent frame as SI frame, the motion between temporally adjacent frames significantly degrade the SI quality. Since a better SI quality would yield a higher video coding efficiency, a lot of research works in the literature have been done on the improvement of SI generation. Predicting a frame based on the previous temporally adjacent frame is refereed to as frame extrapolation, while predicting a frame based on both forward and backward frames is called frame interpolation. This chapter will start with the review of frame interpolation techniques.

2.1 Frame Interpolation

In mathematics, interpolation is a method of constructing new data based on some known data points. Frame interpolation is a video processing technique in which intermediate frames are generated between existing ones. For example, frame t could be generated by using its temporally adjacent frames $t-1$ and $t+1$. Here, the number of the temporally adjacent frames used for frame interpolation depends on the group of picture (GOP) size, which provides a trade-off among SI quality, complexity and compression efficiency. In the previous example, the GOP size is equal to two. The validation of frame

interpolation technique relies on the basic assumption of a typical video sequence, where the adjacent frames are similar and changes are due to object or camera motion. Therefore, the accuracy of motion compensation becomes the key factor of frame interpolation, which has significant impact on the quality of generated SI.

There have been many techniques investigated for improving the motion compensation in frame interpolation. In the rest of this chapter, we will briefly review some existing motion compensation algorithms, such as forward motion estimation method and Sub-pixel precision method, which have been widely used in the current DVC applications.

2.1.1 Forward motion estimation

Without loss of generality, let's denote the $t-1$ and $t+1$ frames as backward and forward frames in a video sequence, respectively. Here we suppose these frames are key frames and available at the decoder in advance. Then we would like to generate the SI frame for the t frame (i.e. WZ frame) based on the backward and forward frames at the decoder. The most existing motion compensation algorithms are based on the block matching between backward and forward frames, where the block size in the anchor frame and the search region size in the target frame offer a trade off between the computation complexity and quality. For example, given a current block $X_B(i_B, j_B)$ in the backward frame, we could enumerate all the candidate blocks using a sliding window through a search region in the forward frame. Then the block $X_F(i_F, j_F)$ which minimizes the matching error $D(X_B(i_B, j_B), X_F(i_F, j_F))$ is chosen as the best match block in the forward frame, where (i_B, j_B) and (i_F, j_F) are the coordinate center of current block and best match block,

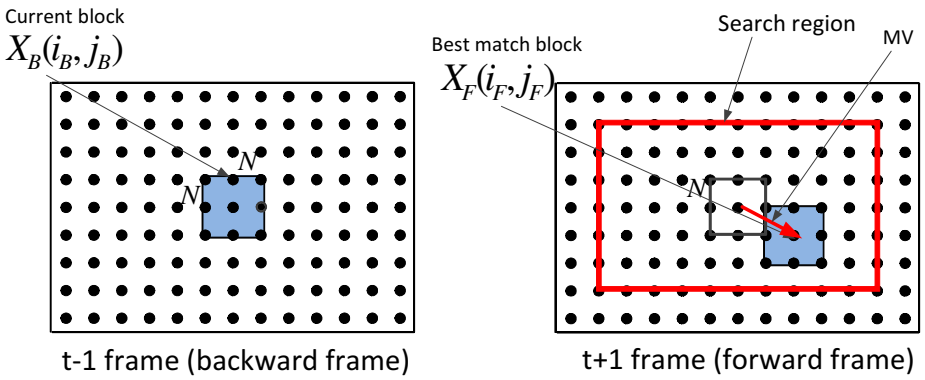


Figure 2.1: Forward motion estimation.

respectively (see Fig. 2.1). Usually, the metric of the mean of absolute difference (MAD) between blocks is used to measure the matching errors, since it is offers a much lower computational complexity compared with the metric of mean squared error.

The MAD between the current block in backward frame and a candidate block in forward frame is given by:

$$MAD(d_i, d_j) = \frac{1}{N^2} \sum_{(i,j)\in B} |X_F(i,j) - X_B(i+d_i, j+d_j)| \tag{2.1}$$

where B denotes block, $N \times N$ is the size of a block, (i, j) represents the pixels' location, and d_i and d_j are the horizontal and vertical displacements, respectively. The values of d_i and d_j are chosen from a range, i.e., so called search region ($d_i = [-M, M]$ and $d_j = [-M, M]$), where a large search region usually provides a better matching with the expense of a higher complexity.

Then the motion vector (MV) between the current block and best match block can be obtained as follows:

$$(v_i, v_j) = \arg\min_{d_i,d_j} MAD(d_i, d_j) \tag{2.2}$$

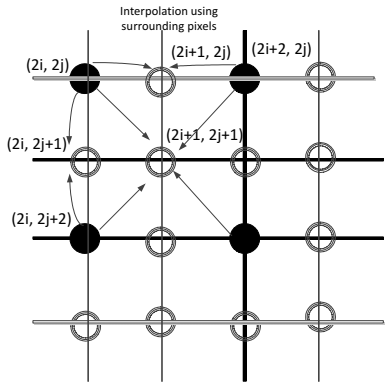


Figure 2.2: Bilinear interpolation, where black point represents the original pixel, while white circle denotes the interpolated pixel.

Actually, the MV (v_i, v_j) can be represented as the offset between the coordinate centers of current block and the best match block $(v_i, v_j) = (i_F - i_B, j_F - j_B)$ (see Fig. 2.1).

Since the interpolated frame is between backward frame and forward frame, the interpolated block $X_I(i_I, j_I)$ with coordinate center (i_I, j_I) can be obtained through linear projection of the best matched block $X_F(i_F, j_F)$ and current block $X_B(i_B, j_B)$. In practice, since multiple motion vectors could pass through an interpolated block $X_I(i_I, j_I)$, it is resealable to choose a best motion vector, whose intersection with the interpolated frame is closest to the coordinate center (i_I, j_I). Then based on the selected best motion vector, we can get the interpolated block as follows:

$$X_I(i,j) = \frac{X_B(i - \frac{v_i}{2}, j - \frac{v_j}{2}) + X_F(i + \frac{v_i}{2}, j + \frac{v_j}{2})}{2} \tag{2.3}$$

2.1.2 Sub-pixel precision method

By default, the precision used in the MV estimation is integer-pixel. However, real MV may not always be multiples of pixel. To allow sub-pixel MV estimation, the search step size must be less than 1 pixel, such as half-pixel

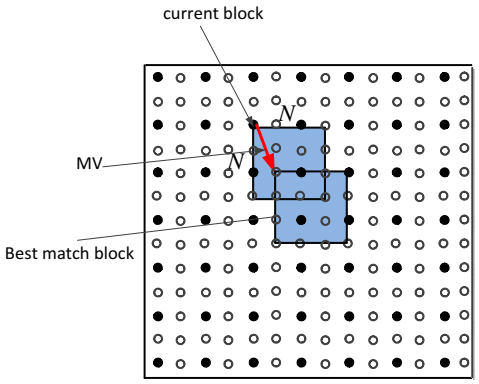


Figure 2.3: Half-pixel forward motion estimation.

and quarter-pixel size. Sub-pixel value can be obtained by pixel interpolation based on the surrounding pixels. Usually, the bilinear interpolation method is used to generate the sub-pixel value as shown in Fig. 2.2. Then following the aforementioned MV estimation procedure but working the sub-pixel precision (see Fig. 2.3), a more precise MV could be found between backward and forward frames.

2.2 Correlation Noise Model in WZ Video Coding

As mentioned in Section 1.3.1, in WZ coding, correlation between the source and SI is modeled as a virtual communication channel which can be expressed in the form $X = Y + N$, where X is the source to be recovered, Y is the SI and N is the the virtual channel noise. Based on experimental observations, most DVC designs so far [16, 32, 33] (with few exceptions) model the correlation noise as Laplace distribution as follows:

$$p[WZ(x,y) - SI(x,y)] = \frac{\alpha}{2}\exp[-\alpha|WZ(x,y) - SI(x,y)|], \tag{2.4}$$

where $WZ(x, y)$ and $SI(x, y)$ are the pixel value at the location (x, y) in WZ and SI frame, respectively, $p(\cdot)$ denotes the probability density function, α is the Laplace distribution parameter defined as

$$\alpha = \sqrt{\frac{2}{\sigma^2}} \tag{2.5}$$

Here, σ^2 is the variance of residuals between WZ and SI frames, where the granularity of residuals can be in the sequence, frame, block and pixel levels. Moreover, the correlation parameter α can vary along both time and space, since the residual errors are usually large, when there are high motions between frames or illumination changes within a frame.

2.3 Background: Correlation Parameter Estimation

Since the capability of correlation parameter estimation has strong impact on the WZ video coding efficiency, many research works have been done for improving correlation estimation in the literature. At the beginning, most of the LDPC and turbo coding-based WZ video coding schemes assume that the correlation noise statistics are stationary along both in time and space [34–36], where the correlation noise statistics could be obtained through training video sequences. However, the above assumption and estimation methods have many limitations, as the correlation statistics strongly depends on the video contents and may vary with time and space. To bridge this gap, non-stationary correlation models (e.g., on the pixel or block level) are studied in [16, 37–40]. In general, the *online estimation* refers to the correlation estimation at the decoder side based on the decoded key frames. In contrast, the *offline estimation* implies the correlation estimation with

original WZ frame available. In practice, *offline estimation* is infeasible due to the requirement in DSC setups. Therefore, *offline estimation* are usually used as benchmark references in experiments. In the rest of this Chapter, we will discuss the *offline correlation noise models* and *online correlation noise models* at the following granularity levels, i.e., sequence, frame, block, and pixel, based on [16].

2.3.1 Offline correlation estimation

Offline correlation estimation requires the original WZ frame and its corresponding SI frame to calculate the correlation parameter. Since the original WZ data is not available at the decoder side, the correlation value estimated through offline correlation estimation scheme can only be treated as benchmark performance. Generally, the correlation is modeled by Laplace distribution as shown in (2.4). In pixel domain (PD) DVC, $WZ(x, y) - SI(x, y)$ corresponds to the residual of pixel values. Moreover, in transform domain (TD) DVC, the residual corresponds to the differences of DCT coefficients. Moreover, since the SI quality varies with time and space, the choice of different granularities in correlation noise models has significant impact on decoding performance. The work in [16] studied a coarse-to-fine strategy to estimate the correlation parameter based on different granularities for both PD and TD DVC. The studies show that a fine grained estimation level usually results a better decoding performance in terms of rate-distortion performance. In the next subsection, we will review the details of offline correlation estimation methods at different granularities.

Pixel domain offline correlation estimation

To obtain the correlation parameter α for PD WZ video coding [16], four granularity levels are analyzed: sequence, frame, block, and pixel.

The general procedures of computing correlation parameter α for PD WZ video coding is summary as follows [16]:

1. Residual frame generation:

$$R(x,y) = WZ(x,y) - SI(x,y), \tag{2.6}$$

   where $R(x,y)$ denotes the residual frame, (x,y) represents the pixel position of a frame.

2. Variance computation: according to the definition, the variance of a random variable is $\sigma_Z^2 = E[Z^2] - (E[Z])^2$, where $E[\cdot]$ is the expectation operator, the variance at different granularity levels can be obtained by averaging the variance on the corresponding levels:

   (a) For sequence level: since a whole video sequence is characterized by the same variance, variance will be averaged for the video sequence.

   First, the variance for each residual frame is given by

$$\begin{aligned} \sigma_R^2 &= E[R(x,y)^2] - (E[R(x,y)])^2 \\ E[R(x,y)] &= \frac{1}{H \times W} \sum_{x=1}^{H} \sum_{y=1}^{W} R(x,y) \\ E[R(x,y)^2] &= \frac{1}{H \times W} \sum_{x=1}^{H} \sum_{y=1}^{W} [R(x,y)]^2 \end{aligned} \tag{2.7}$$

where H and W are the height and width of a frame.

Then averaged frame variance at sequence level is:

$$\sigma_s^2 = \frac{\sum_{\text{F frames}} E_R[R^2]}{F} - \left(\frac{\sum_{\text{F frames}} E_R[R]}{F}\right)^2 \tag{2.8}$$

where F is the total number of WZ frame in the coded video sequence.

(b) For frame level: all the samples in one frame are characterized by the same variance. The average variance σ_R^2 of the residual frame is

$$\sigma_R^2 = E[R(x,y)^2] - (E[R(x,y)])^2 \tag{2.9}$$

(c) For block level: all the samples in each $m \times m$ block of a frame have the same variance.

First, the k-th block in the residual frame can be written as

$$R_k(x,y) = WZ_k(x,y) - SI_k(x,y). \tag{2.10}$$

Then the averaged variance of k-th block in R frame can be calculated as

$$\sigma_{R_k}^2 = E_{R_k}[R_k(x,y)^2] - (E_{R_k}[R_k(x,y)])^2 \tag{2.11}$$

where the expectations take over all pixels with in the k-th block.

(d) For pixel level: the variance of each residual pixel is

$$\sigma_p^2 = (R(x,y))^2 = (WZ(x,y) - SI(x,y))^2 . \tag{2.12}$$

3. Correlation parameter α computation: according to the relationship between parameter α and variance in (2.5), the correlation parameter of different granularity levels can be written as

   (a) For sequence level: $\alpha_s = \sqrt{\frac{2}{\sigma_s^2}}$

   (b) For frame level: $\alpha_R = \sqrt{\frac{2}{\sigma_R^2}}$

   (c) For block level: $\alpha_{R_k} = \left\{ \begin{array}{c} \sqrt{2}, \sigma_{R_k}^2 \leq 1 \\ \sqrt{\frac{2}{\sigma_{R_k}^2}}, \sigma_{R_k}^2 > 1. \end{array} \right\}$

   (d) For pixel level: $\alpha_p = \left\{ \begin{array}{c} \sqrt{2}, |R(x,y)| \leq 1 \\ \sqrt{\frac{2}{R(x,y)^2}}, |R(x,y)| > 1. \end{array} \right\}$

   Here, since the variances obtained from different granularity levels are based on the average of samples in different precisions, a variance value calculated in a high precision (e.g. pixel, block levels) is more likely to be close to zero compared with low precision estimations (e.g. frame and sequence levels). In order to avoid numerical errors and maintain a reliable estimation, the maximum estimated correlation parameter is bounded by $\sqrt{2}$. Finally, the obtained correlation parameter of different granularities can be used to help the decoding of WZ frame with different performance gains. In general, a higher estimation precision will results a better decoding performance.

Transform domain offline correlation estimation

In contrast to PD offline correlation estimation, the estimation takes place on the DCT coefficients instead of pixel in TD DVC. For the offline correlation estimation in TD DVC [16], three granularity levels are studied, i.e., DCT band/sequence, DCT band/frame, and coefficient/frame.

The procedures of computing correlation parameter α for TD WZ video coding is similar to the procedures of PD video coding, which can be summarized as follows [16]:

1. Residual frame generation: Residual frame $R(x, y)$ is generated as describe in (2.6).

2. Residual frame DCT transform: applying 4×4 block-based discrete cosine transform on the residual frame $R(x, y)$ will generate the DCT coefficients frame T, which consists of 16 DCT bands

$$T(u, v) = DCT[R(x, y)] \tag{2.13}$$

   where (u, v) denotes the DCT coefficient position of T frame.

3. Variance computation: according to the definition of variance for a random variable as $\sigma_Z^2 = E[Z^2] - (E[Z])^2$, where $E[\cdot]$ is the expectation operator, the variance at different granularity levels can be obtained by averaging the variance on the corresponding levels:

   (a) For DCT band/sequence level: it means that the same DCT band in a whole video sequence has the same variance, which can be obtained by averaging the corresponding DCT bands in whole video sequence.

First, let T_b denotes a set of coefficients at the b-th DCT band of transformed frame T, where the length J of T_b is the ratio between the frame size and the number of total DCT bands, i.e., $\frac{H \times W}{4 \times 4}$ in our setups. Then the variance of each DCT band within a given frame is given by

$$\begin{aligned} \sigma_b^2 &= E_b[T_b^2] - (E_b[T_b])^2 \\ E_b[T_b] &= \frac{1}{J} \sum_{x=1}^{J} T_b(j) \\ E_b[T_b^2] &= \frac{1}{J} \sum_{x=1}^{J} [T_b(j)]^2 \end{aligned} \tag{2.14}$$

Then average variance at sequence level is:

$$\sigma_{b,s}^2 = \frac{\sum_{\text{F frames}} E_b[T_b^2]}{F} - \left(\frac{\sum_{\text{F frames}} E_b[T_b]}{F} \right)^2 \tag{2.15}$$

where F is the total number of WZ frame in the coded video sequence.

(b) For DCT band/frame level: all the samples in each DCT band of a transformed residual frame are characterized by the same variance. The average variance σ_b^2 of the DCT band b for a certain transformed residual frame is

$$\sigma_b^2 = E_b[T_b^2] - (E_b[T_b])^2 \tag{2.16}$$

(c) For coefficient/frame level: the variance of each DCT coefficient

has different value, which can be written as

$$\sigma_c^2 = \left(T(u,c)\right)^2 . \tag{2.17}$$

4. Correlation parameter α computation: according to the relationship between parameter α and variance (2.5), the correlation parameter of different granularity levels can be written as

   (a) For DCT band/sequence level: $\alpha_{b,s} = \sqrt{\frac{2}{\sigma_{b,s}^2}}$

   (b) For DCT band/frame level: $\alpha_b = \sqrt{\frac{2}{\sigma_b^2}}$

   (c) For coefficient/frame level: $\alpha_c = \left\{ \begin{array}{c} \sqrt{2}, |T(u,v)| \leq 1 \\ \sqrt{\frac{2}{T(u,v)^2}}, |T(u,v)| > 1. \end{array} \right\}$

   Here, the maximum the correlation parameter α is also bounded by $\sqrt{2}$ in case (c) for the same reasons discussed in the PD DVC.

2.3.2 Online correlation estimation

Due to the requirements of DSC setup, where each WZ frame needs to be encoded independently, the offline correlation parameter estimation is infeasible in the practical DVC applications. As the original WZ frame is not available at the decoder, we should resort to other strategies to perform online correlation estimation at the decoder side. In [16], the online correlation estimation methods with threes different granularities, i.e., frame, block, pixel levels in PD and TD DVC, are studied. The strategies of online correlation parameter estimation at the decoder is similar to the offline case except that the residual is estimated in a different way, as the original WZ frame is not explicitly available before decoding. The following sections review the *online*

correlation estimation at different granularity levels in terms of PD and TD DVC.

Pixel domain online correlation estimation

The procedures of computing correlation parameter α for PD DVC are summarized as follows [16]:

1. Residual frame generation: since original WZ frame is not available at the decoder for the generation of residual frame, the residual frame $R(x, y)$ will be approximated through the difference between motion compensated backward and forward frames X_B and X_F, which is given as follows:

$$R(x, y) = \frac{X_F(x + dx_f, y + dy_f) - X_B(x + dx_b, y + dy_b)}{2} \tag{2.18}$$

   where $X_F(x+dx_f, y+dy_f)$ and $X_B(x+dx_b, y+dy_b)$ denotes the forward and backward motion compensated frames, respectively, (x, y) represents the pixel position of a residual frame, (dx_f, dy_f) and (dx_b, dy_b) represents the motion vectors for the X_F and X_B frames, respectively.

2. Variance computation: according to the variance definition of a random variable $\sigma_Z^2 = E[Z^2] - (E[Z])^2$, where $E[\cdot]$ is the expectation operator, the variance at different granularity levels can be obtained by averaging the variance on the corresponding levels:

   (a) For frame level: all the samples in each frame are characterized by the same variance. The estimated variance $\hat{\sigma}_R^2$ of the residual

frame is

$$\hat{\sigma}_R^2 = E[R(x,y)^2] - (E[R(x,y)])^2 \tag{2.19}$$

(b) For block level: all the samples in each $m \times m$ block of R frame have the same variance.

First, the residual frame k-th block can be obtained as

$$R_k(x,y) = \frac{X_{F_k}(x+dx_f, y+dy_f) - X_{B_k}(x+dx_b, y+dy_b)}{2} \tag{2.20}$$

Then the averaged variance of R frame k-th block is

$$\begin{aligned} \hat{\sigma}_{R_k}^2 &= E_{R_k}[R_k(x,y)^2] - (E_{R_k}[R_k(x,y)])^2 \\ E_{R_k}[R_k(x,y)] &= \frac{1}{m \times m} \sum_{x=1}^{m} \sum_{y=1}^{m} R_k(x,y) \\ E_{R_k}[R_k(x,y)^2] &= \frac{1}{m \times m} \sum_{x=1}^{m} \sum_{y=1}^{m} [R_k(x,y)]^2 \end{aligned} \tag{2.21}$$

(c) For pixel level: the variance of each residual pixel is

$$\hat{\sigma}_p^2 = (R(x,y))^2 \,. \tag{2.22}$$

3. Correlation parameter α computation: according to the relationship between parameter α and variance (2.5), the correlation parameters of different granularity levels can be written as

(a) For frame level:

$$\hat{\alpha}_R = \sqrt{\frac{2}{\hat{\sigma}_R^2}} \tag{2.23}$$

(b) For block level:

$$\hat{\alpha}_{R_k} = \begin{cases} \hat{\alpha}_R, & \hat{\sigma}_{R_k}^2 \leq \hat{\sigma}_R^2 \\ \sqrt{\frac{2}{\hat{\sigma}_{R_k}^2}}, & \sigma_{R_k}^2 > \hat{\sigma}_R^2, \end{cases} \tag{2.24}$$

In the above equation, for the case $\hat{\sigma}_{R_k}^2 \leq \hat{\sigma}_R^2$, we argue that the interpolated block has a higher quality than that averaged on the interpolated frame. However, to maintain the uncertainty between WZ frame and SI frame, the maximum value of $\hat{\alpha}_{R_k}$ is bounded by $\hat{\alpha}_R$, which can be obtained according to (2.19). When $\sigma_{R_k}^2 > \hat{\sigma}_R^2$, it means that the interpolated block may have a low quality, which corresponds to a high residual error in the offline case. As a higher residual error implies a lower confidence about the similarity between SI frame and WZ frame, we will choose a smaller $\hat{\alpha}_{R_k}$ as $\sqrt{\frac{2}{\hat{\sigma}_{R_k}^2}}$.

(c) For pixel level:

$$\hat{\alpha}_p = \begin{cases} \hat{\alpha}_R, & \sigma_{R_k}^2 \leq \hat{\sigma}_R^2 \\ \hat{\alpha}_{R_k}, & (\sigma_{R_k}^2 > \hat{\sigma}_R^2) \wedge (D_{R_k} \leq \hat{\sigma}_R^2) \\ \hat{\alpha}_{R_k}, & (\sigma_{R_k}^2 > \hat{\sigma}_R^2) \wedge (D_{R_k} > \hat{\sigma}_R^2) \wedge [R(x,y)]^2 \leq \hat{\sigma}_{R_k}^2 \\ \sqrt{\frac{2}{[R(x,y)]^2}}, & (\sigma_{R_k}^2 > \hat{\sigma}_R^2) \wedge (D_{R_k} > \hat{\sigma}_R^2) \wedge [R(x,y)]^2 > \hat{\sigma}_{R_k}^2 \end{cases} \tag{2.25}$$

where D_{R_k} is the distance between the average value of R frame k-

th block and average value of R frame, which is written as $D_{R_k} = (E_{R_k}[R(x,y)] - E_R[R(x,y)])^2$. For case 1) $\sigma^2_{R_k} \leq \hat{\sigma}^2_R$, since the interpolated block has a high quality, all the pixels of the k-th block have the same confidence information as that on the frame level. For case 2) $D_{R_k} \leq \hat{\sigma}^2_R$ means that the k-th block has similar proprieties as that in the frame level, but with a low confidence, as $\sigma^2_{R_k} > \hat{\sigma}^2_R$. Similarly, for case 3), although both the k-th block variance and block distance are higher than these in the frame level (i.e., $(\sigma^2_{R_k} > \hat{\sigma}^2_R) \wedge (D_{R_k} > \hat{\sigma}^2_R)$), the interpolated pixel itself has a good quality as $[R(x,y)]^2 \leq \hat{\sigma}^2_{R_k}$. Thus, such pixel shares the correlation parameters as the k-th block. Finally, for case 4), since the pixel is not well interpolated, it deserves a lower α value than that in block level.

In summary, the above proposed heuristic rules are based on the following guidelines:

1. If a finer grained estimation level (e.g., block or pixel levels) has equal or higher confidence of SI quality than that of a coarser estimation level (e.g., frame or block levels), which is also equivalent to $\sigma^2_{block} \leq \sigma^2_{frame}$ or $\sigma^2_{pixel} \leq \sigma^2_{block}$, the correlation parameter from the coarser level will be chosen as its correlation parameter. This scenario maintains a sufficient uncertainty between WZ frame and SI frame.

2. If a finer grained estimation level has a lower confidence of SI quality than that of a coarser estimation level, a smaller correlation parameter α will be calculated according to the formula (2.5).

Transform domain online correlation estimation

1. Residual frame generation: Residual frame $R(x, y)$ is generated as describe in (2.18).

2. Residual frame DCT transform: DCT transformed coefficients frame T is generated as in (2.13).

3. $|T|$ frame generation: take the absolute value of each element of T frame.

4. Variance computation: according to the variance definition of a random variable $\sigma_Z^2 = E[Z^2] - (E[Z])^2$, where $E[\cdot]$ is the expectation operator, the variance at different granularity levels can be obtained by averaging the variance on the corresponding levels:

   (a) For $|T|$ frame DCT band/frame level: all the samples in the same DCT band of a transformed residual frame are characterized by the same variance. The average variance σ_b^2 of the DCT band b for a certain transformed residual frame is

   $$\hat{\sigma}_b^2 = E_b[|T|_b^2] - (E_b[|T|_b])^2 \tag{2.26}$$

   (b) For coefficient/frame level: the distance between a coefficient ($|T|_b(u, v)$) of $|T|$ frame at band b, and the average value $\hat{\mu}_b$ of $|T|_b$ is computed as follows

   $$D_b(u, v) = |T|_b(u, v) - \hat{\mu}_b. \tag{2.27}$$

5. Correlation parameter α computation: According to the relationship between parameter α and variance (2.5), the correlation parameter of

different granularity levels can be written as

(a) For DCT band/frame level: $\hat{\alpha}_b = \sqrt{\frac{2}{\hat{\sigma}_b^2}}$

(b) For coefficient/frame level:

$$\hat{\alpha}_c = \begin{cases} \hat{\alpha}_b, & [D_b(u,v)]^2 \leq \hat{\sigma}_b^2 \\ \sqrt{\frac{2}{[D_b(u,v)]^2}}, & [D_b(u,v)]^2 > \hat{\sigma}_b^2 \end{cases} \tag{2.28}$$

The heuristic rules introduced in the above equation follows the same guidelines as discussed in the PD DVC section. To avoid redundancy, we would like to direct readers to the previous section for details.

CHAPTER 3

CORRELATION ESTIMATION IN DVC WITH PARTICLE FILTERING

As mentioned in the introduction and Chapter 2, since the decoding performance of DSC principle relies on the knowledge of correlation very much, the design of correlation estimation scheme becomes a significant task both in theoretical studies and practical applications (e.g. DVC). In addition, the existing correlation estimation methods in DVC are usually twofold: online estimation and on-the-fly (OTF) estimation. The previous Chapter has reviewed the online estimation schemes proposed in [16], which are mainly based on the adjacent motion compensated frames before decoding and empirical rules. However, it does not take into account the information from the received syndromes. Since the received syndromes contain important information of the original WZ frame, such additional information could potentially improve the estimation performance. In chapter, I will propose an OTF correlation estimation scheme, which can extract syndrome information for improving correlation estimation during the LDPC decoding. The proposed OTF estimation scheme is based on the Factor graph and Bayesian approximate inference. Therefore, I first introduce some background knowledges before I walk through the proposed model.

3.1 Theoretical Background

3.1.1 Factor Graph

Factor graph, as a bipartite graph, can be used to represent the factorization of a function. Factor graph is characterized by nodes (usually depicted by a circle) for every variable x_s in the function, additional nodes (depicted by small squares) for each factor $f_s(x_s)$ of the function and undirected links connecting each factor node to all of the variables nodes on which that factor depends. Given a factorization of a function over a set of random variables

$$g(x_1, x_2, \cdots, x_n) = \prod_s f_s(\mathbf{x}_s), \tag{3.1}$$

where $(x_1, x_2, \cdots, x_n)$ is a set of variables in the function, $\mathbf{x}_s$ is a subset of variables and $f_s(\mathbf{x}_s)$ is a factor function of all variables in $\mathbf{x}_s$.

Consider a simple function that factorized as follows:

$$g(x_1, x_2, x_3) = f_1(x_1, x_2) f_2(x_2, x_3) f_3(x_1, x_3) \tag{3.2}$$

Then the factor graph of the factorization of $g(x_1, x_2, x_3)$ is shown in Fig. 3.1

Moreover, a factor graph is a particular type of graphical model that enables efficient computation of marginal distributions through the sum-product algorithm, commonly referred to as belief propagation (see the following Section 3.1.2).

3.1.2 Belief Propagation Algorithm

The BP algorithm is an approximate technique for computing marginal probabilities by exchanging the message between neighboring nodes on a factor

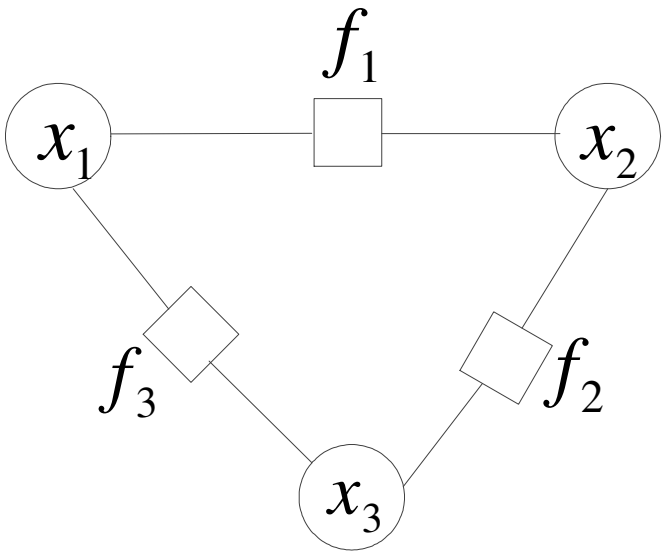


Figure 3.1: Factor graph.

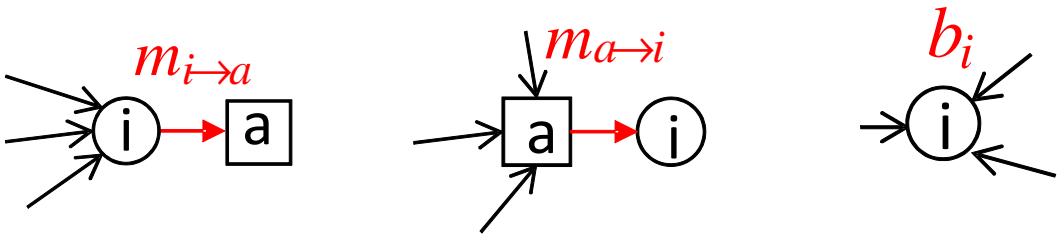


Figure 3.2: Belief propagation algorithm.

graph. Denote $m_{a\to i}(x_i)$ as the message sent from a factor node a to a variable node i, and $m_{i\to a}(x_i)$ as the message sent from a variable node i to a factor node a. Loosely speaking, $m_{a\to i}(x_i)$ and $m_{i\to a}(x_i)$ can be interpreted as the beliefs of node i taking the value x_i transmitting from node a to i and from node i to a, respectively. The message updating rules can be expressed as follows (see Fig. 3.2):

$$m_{i\to a}(x_i) \propto \prod_{c\in N(i)\setminus a} m_{c\to i}(x_i) \tag{3.3}$$

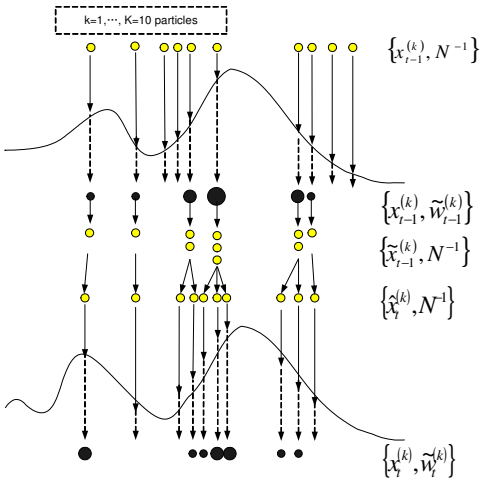


Figure 3.3: Particle filtering algorithm.

and

$$m_{a\to i}(x_i) \propto \sum_{\mathbf{x}_a \backslash x_i} \left(f_a(\mathbf{x}_a) \prod_{j \in N(a)\backslash i} m_{j\to a}(x_j) \right), \tag{3.4}$$

where $N(i)\backslash a$ denotes the set of all neighbors of node i excluding node a; f_a is the factor function for factor node a; $\sum_{\mathbf{x}_a \backslash x_i}$ denotes a sum over all the variables in $\mathbf{x}_a$ that are arguments of f_a except x_i. Moreover, the BP algorithm approximates the belief of node i taking x_i as

$$b_i(x_i) \propto \prod_{a \in N(i)} m_{a\to i}(x_i). \tag{3.5}$$

3.1.3 Particle Filtering

Particle filters, also known as Sequential Monte Carlo methods, are sophisticated techniques for optimal numerical estimation when exact solutions cannot be analytically derived [41]. It is used to estimate posterior probability distributions of the unknown object states through a list of particles by estimating the state at one time from available measurements up to next time [41].

The procedure of particle filtering algorithm is shown as follows:

1. First, genertate K number particles $x_{t-1}^{(k)}$ with weight $\frac{1}{N}$.

2. Then K new samples, $\tilde{x}_{t-1}^{(1)}, \cdots, \tilde{x}_{t-1}^{(K)}$, will be drawn with probabilities sampled from a weight distribution using systematic resampling [42]. As a result, some $x_{t-1}^{(k)}$ that have small probabilities will be likely to be discarded whereas those with high probability will be repeatedly drawn.

3. To maintain the diversity of the particles, the particle locations will be perturbed by an Metropolis-Histing (MH) [43] based Gaussian random walk, which consists of two basic stages. First, let the proposed new K particles at each iteration be $\hat{x}_t^{(k)} = \tilde{x}_{t-1}^{(k)} + Z_r$, that is the current value plus a Gaussian random variable $Z_r \sim N(0, \sigma_r^2)$. Second, decide whether the proposed values of new particles are rejected or retained by computing the acceptance probability $a\{\hat{x}_t^{(k)}, \tilde{x}_{t-1}^{(k)}\} = min\{1, \frac{p\left(\hat{x}_t^{(k)}\right)}{p\left(\tilde{x}_{t-1}^{(k)}\right)}\}$, where $\frac{p\left(\hat{x}_t^{(k)}\right)}{p\left(\tilde{x}_{t-1}^{(k)}\right)}$ is the ratio between the proposed particle value and the previous particle value. When the proposed value has a higher posterior probability than the current value $\tilde{x}_{t-1}^{(k)}$, it is always accepted; otherwise, it is accepted with probability a.

4. Update weight by resetting to a uniform weight $\frac{1}{K}$ for each particle.

5. Iterate steps 2 to 4 unless the maximum number of iterations is reached or other exit condition is satisfied.

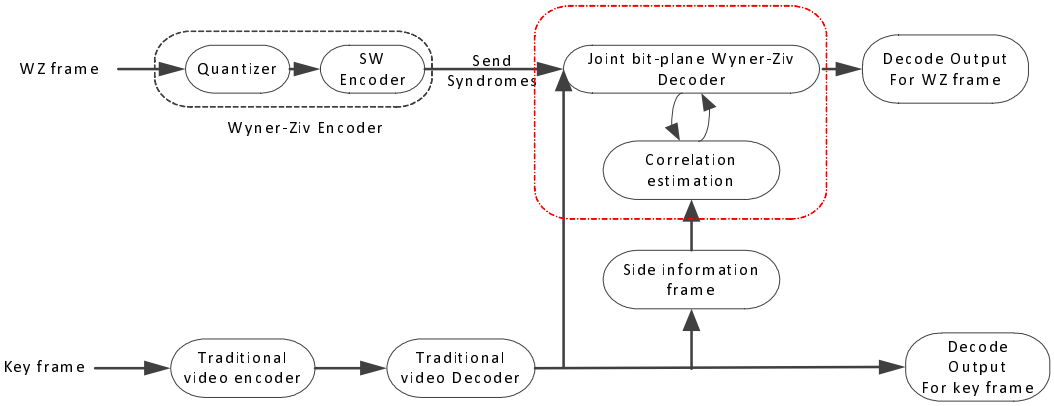


Figure 3.4: Work flow of the proposed WZ decoder with OTF correlation estimation.

3.2 Correlation Estimation in DVC With Particle Filtering

In this section, I will talk about our proposed OTF correlation estimation model, where the work flow is shown in Fig. 3.4. Here, I first describe how a factor graph is designed for our problem, and then I explain the concept of adaptive graph-based decoding incorporating particle filtering for the OTF correlation estimation.

3.2.1 Factor Graph Construction

The construction of a factor graph capturing and connecting SW coding and correlation tracking is described after identification of appropriate variable nodes and factor nodes. Variables nodes denote unknown variables such as coded bits and correlation variance and factor nodes represent the connection among multiple variable nodes. I model the correlation between source and side information as Gaussian or Laplacian, and build a 3D graph to capture joint bit-plane decoding, striving for efficient compression, accurate correlation modeling, and good performance.

For WZ coding, I carry out joint bit-plane coding by first quantizing an N-

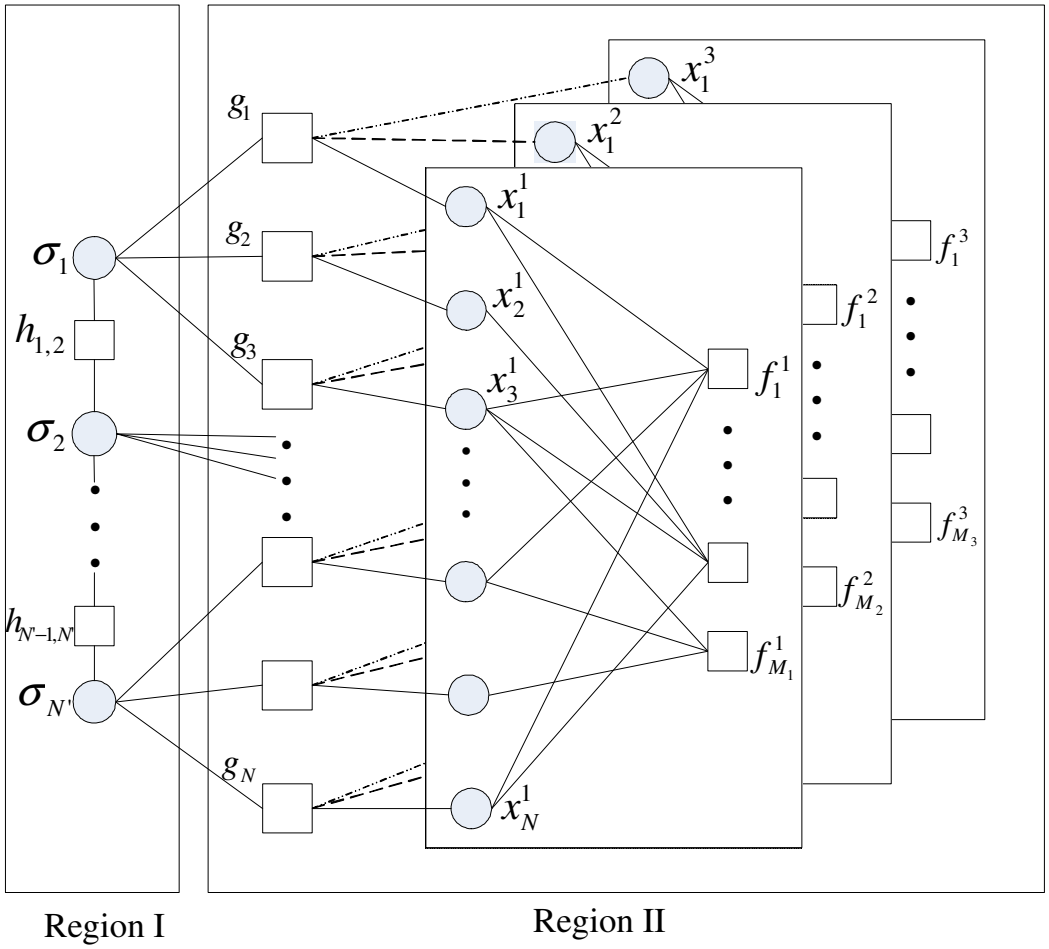


Figure 3.5: The proposed WZ decoder with OTF correlation estimation.

bit source x_i $(i = 1, \cdots, N)$ into $Q[x_i]$, using 2^q level Lloyd-Max quantization. I denote $x_i^1, x_i^2, \cdots, x_i^q$ as the binary format of the index $Q[x_i]$, and denote $\mathbf{B}_j = x_1^j, x_2^j, \cdots, x_N^j$ as the j-th significant bit-plane. Each bit-plane is SW encoded independently by using q LDPC codes. Thus the j-th SW encoder compresses $\mathbf{B}_j$ $(j = 1, \cdots, q)$ and computes the parity / syndrome bits $\mathbf{S}_j = s_1^j, s_2^j, \cdots, s_M^j$. This results in an $N : M_j$ SW compression ratio, where M_j denotes the number of output syndrome bits for j-th bit-plane. I thus define SW coding rate as $R = \frac{\sum_{j=1}^{q} M_j}{Nq}$, where $R = 1$ represents no compression.

The parity factor nodes $f_1^j, f_2^j, \cdots, f_{M_j}^j$ of the SW LDPC code take into

account constraints imposed by the received parity bits. The variable nodes and factor nodes f_a^j, $a = 1, \cdots, q$ are represented as circle nodes and square nodes, respectively, in Region II of the factor graph illustrated in Fig. 3.5, where $q = 3$. The q planes capturing SW parity factor nodes and source variable nodes represent the third dimension of the graph.

Let y_i $(i = 1, , N)$ be the N-bit side information available at the decoder. To take into account the remaining correlation between quantized source $Q[x_i]$ and side information y_i, I additionally define correlation factor nodes g_i, $i = 1, 2, \cdots, N$ as in (3.6), where $P(\bullet)$ is the value of quantization partition at index "$\bullet$"; i.e., if a sampled source x_i satisfies $P(\bullet) \leq x_i < P(\bullet + 1)$, the quantization index $Q[x_i]$ of source x_i is equal to "$\bullet$". σ_z is the standard deviation of the correlation noise between x_i and y_i assuming an additive correlation model $x_i = y_i + z_i$, where z_i is Gaussian noise independent of y_i. Alternatively, the correlation factor node g_i can be expressed as (5.4) when I consider Laplacian noise.

$$f_i(Q[x_i], y_i, \sigma_z) = \int_{P(Q[x_i])}^{P(Q[x_i]+1)} \frac{1}{\sqrt{2\pi\sigma_z^2}} e^{-\left(\frac{x-y_i}{\sqrt{2}\sigma_z}\right)^2} dx$$
$$= \frac{1}{2}\text{erfc}(-\frac{P(Q[x_i]+1) - y_i}{\sqrt{2}\sigma_z}) - \frac{1}{2}\text{erfc}(-\frac{P(Q[x_i]) - y_i}{\sqrt{2}\sigma_z}) \tag{3.6}$$

$$g_i(Q[x_i], y_i, \alpha) = \int_{P(Q[x_i])}^{P(Q[x_i]+1)} \frac{\alpha}{2} e^{\alpha|x-y_i|} dx, \tag{3.7}$$

where α is the scale parameter of the Laplacian distribution.

While in standard WZ decoding, σ_z is assumed to be constant and known a priori, in practical applications such as DVC this is rarely the case. I assume that σ_z is unknown and varies slowly over time, typical for correlated frames in a video sequence. I now add extra correlation variable nodes $\sigma_1, \sigma_2, \cdots, \sigma_{N'}$

shown as circle nodes in Region I of our constructed factor graph in Fig. 3.5. Each factor node g_i in Region II is connected to an additional variable node corresponding to σ_l, $l = 1, 2, \cdots, N'$, where N' is the number of correlation variable nodes in Region I. The factor function $g_i(Q[x_i], y_i, \sigma_z)$ of g_i is the same as (3.6) or (5.4). The number of factor nodes g_i that each variable σ_l is connected to is defined as the connection ratio, which is three in the example shown in Fig. 3.5. Since our assumption is that correlation variance varies slowly temporally and spatially, it is expected that adjacent variable nodes σ_l will not differ much in value. This is represented in the graph by additional factor nodes $h_{l,l+1}$ that connect adjacent variable nodes σ_l and σ_{l+1}, as in (3.8), where λ is a hyper-prior. That is,

$$h_{l,l+1}(\sigma_l, \sigma_{l+1}) = \exp(-\frac{(\sigma_{l+1} - \sigma_l)^2}{\lambda}), \tag{3.8}$$

The final factor graph (Fig. 3.5) comprises Region II, with a standard Tanner graph for bit-plane LDPC decoding, and Region I, with a bipartite graph capturing correlation variance σ_z^2, and factor nodes g_i , defined in (3.6) and (5.4) for Gaussian and Laplacian correlation, respectively, connecting the two regions.

3.2.2 Correlation Estimation in DVC With Particle Filtering

In this section, the message passing algorithm for efficient SW decoding and correlation estimation via the belief propagation (BP) algorithm operating jointly with the particle filtering algorithm is described.

BP operates on the factor graph described in Section 3.1.2. Messages are passed iteratively between connected variable nodes and factor nodes

in both regions of the graph until the algorithm converges or until a fixed number of iterations is reached. These messages (inferences or beliefs on source bits and correlation) will represent the influence that one variable has on another. I group these types of messages into the two connected regions identified previously, thus generalizing BP. Hence, our correlation estimation exploits variations in side information in each bit-plane, and hence pixel, and dynamically tracks spatial and temporal variations in correlation between source and side information.

Standard BP (the sum-product algorithm), generally used for SW decoding, can handle only discrete variables. The correlation variance, however, is not a discrete variable, since it varies continuously over time. I therefore resort to particle filtering [44], which is integrated within the standard BP algorithm in order to handle continuous variables. Particle filtering (PF) estimates the a posteriori probability distribution of the correlation variable node σ_l by sampling a list of random particles (tied to each correlation variable node) with associated weights.

Each variable node σ_l is modelled with N_p particles . Locations and corresponding weights of each particle σ_l^k, $k = 1, \cdots, N_p$, are adjusted with the updating of the BP algorithm. The belief $b(\sigma_l^k)$ of each particle is essentially the particle weight w_l^k, whose update is achieved by updating variable nodes using standard BP.

The first step of the particle-based BP algorithm is the initialization of each particle value $(\sigma_l^k)^2$ to $(\hat{\sigma}^2)$ and each particle weight w_l^k to $1/N_p$. $(\hat{\sigma}^2)$ is chosen with some prior knowledge of the source statistics. If the input codeword has the same parity as the received one in each bit-plane, the algorithm terminates; otherwise, all variable nodes, factor nodes, and particles

are updated iteratively.

Systematic resampling [42] is applied once all weights have been updated for all N_p particles in each variable node σ_l in Region I to discard particles with negligible weight and concentrate on particles with larger weights. However, after the resampling step, particles tend to congregate around values with large weight. To maintain diversity of the particles, the new particle locations are perturbed by applying the random walk Metropolis-Hastings algorithm, which essentially adds Gaussian or Laplacian noise on the current value σ_l for each of the N particles. The weight of each particle is then reset to a uniform weight for each particle. A new codeword is generated at the end of each iteration until the BP algorithm finds a valid codeword or until it reaches a maximum number of iterations.

The message passing schedule for the factor graph in Fig. 3.5, incorporating PF, is summarized in Table 3.1.

3.3 Experimental Results

To verify the performance of correlation tracking across WZ-encoded frames in a video sequence, I tested the above setup with many standard QCIF 15Hz video sequences, "Soccer", "Coastguard", "Foreman", "Hall& monitor"and "Salesman". These videos covered fast, median and slow motion conditions. In the video sequences of our experiments, I consider two frames per group of picture (GOP) and a total of 149 frames for each video sequence. The odd frames are the key frames which are intraframe coded and reconstructed using H.264/AVC with profile used in [45]. The even frames, between two intra coded key frames, are WZ frames which are encoded using LDPCA codes and recovered through our proposed joint bit-plane decoding with correlation

Table 3.1: Message passing algorithm jointly updating inference on source and correlation variance variable nodes.

1: Initialise the values of N_p variable particles in Region I, $\sigma_l = \hat{\sigma}$, for estimating
the correlation noise and a uniform weight $1/N_p$.
2: Initialise messages sent from factor nodes $g_i(Q[x_i], y_i, l)$ connecting Regions I
and II to variable nodes x_i in Region II as in (3.6), for each of the q bit-planes,
where $\sigma_l = \hat{\sigma}$.
3: If the decoded estimate has the same syndrome as the received one or
maximum number of iterations is reached, export the decoded codeword and
finish. If not, go to Step 4.
4: Update variable nodes in Region II using standard BP (sum-product algorithm) for channel decoding.
5: Update particles in Region I by updating variable nodes using BP.
6: Compute the belief for each variable in Region II being $x_i \in \{0, 1\}$.
7: Compute the belief (=weight) of each particle for each variable node in Region I.
8: Systematic resampling of particles in Region I, followed by the Metropolis-
Hastings algorithm and resetting the weight of particles to a uniform weight.
9: Update factor nodes firstly in Region II, the Region I and finally those
connecting the two regions.
10: Generate a new codeword based on the belief of variable nodes in Region II.
11: Go back to Step 3.

Table 3.2: Average results in terms of Bjontegaard delta PSNR and bitrate for sequences including Foreman, Soccer, Coastguard, Hall and Salesman.

		BJM	
		Δ PSNR (dB)	Δ Rate %
Foreman	Q1	-0.1341	-6.0592
	Q3	-0.2327	-10.5428
	Q5	-0.2576	-14.1864
Soccer	Q1	-0.4253	-16.1156
	Q3	-0.0104	-14.6673
	Q5	-0.0315	-18.1785
Coastguard	Q1	-0.0384	-0.0271
	Q3	-0.7409	-11.7729
	Q5	-0.9804	-18.9212
Hall	Q1	-0.2010	-6.1717
	Q3	-0.2609	-7.5663
	Q5	-0.8541	-12.8799
Salesman	Q1	-0.0318	-5.1683
	Q3	-0.0971	-7.9557
	Q5	-0.3693	-10.5094

estimation described in previous Section.

The test conditions for the WZ frame are described in the following. A 4×4 float DCT transform is performed for WZ frame first, and the uniform scalar quantizer with data range $[0, 2^{11})$ and a dead-zone quantizer with doubled zero interval and the dynamic data range $[-\text{MaxVal}_b, \text{MaxVal}_b)$ are applied for DC and AC band [45], respectively. At the decoder, side information frame is generated by the recovered frames from H.264/AVC decoding, where search range with ±32 pixels is used for the forward motion estimation [45]. The following parameters are used in our simulation: the number of particles $N_p = 12$, connection ratio $C = 4$ and random walk step $\sigma_r = 0.005$. In addition, PBP algorithm is used only after 50 number of BP iterations and then is performed every 20 number of BP iterations.

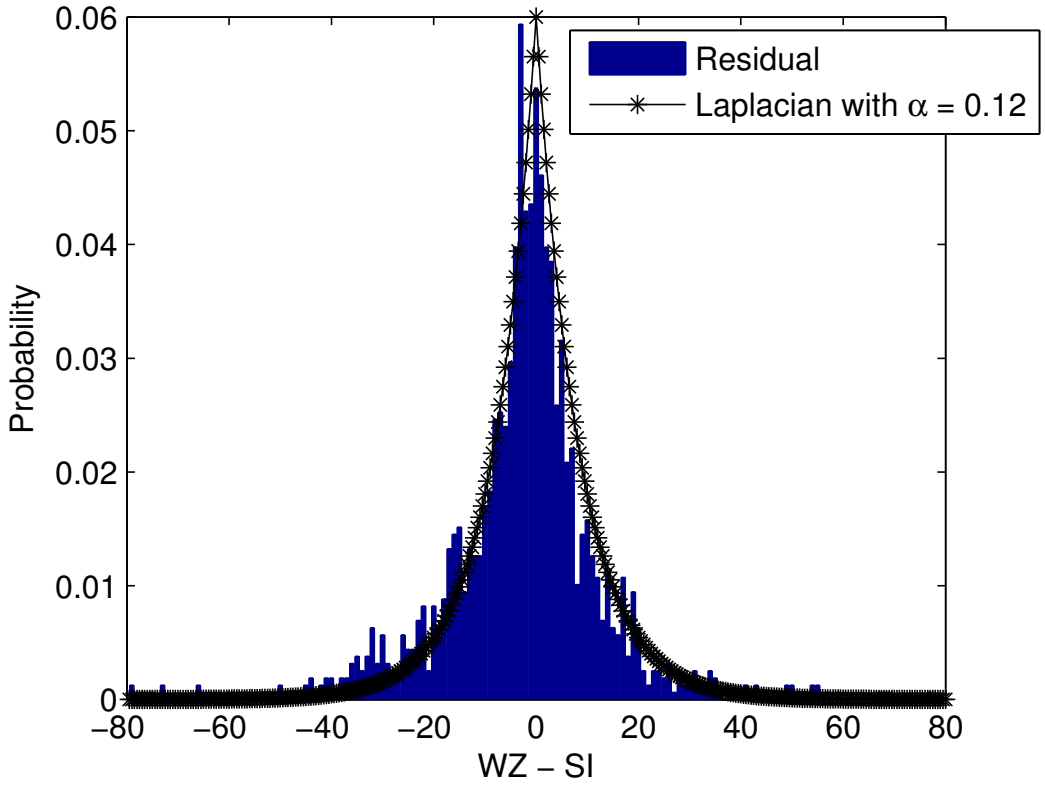


Figure 3.6: Residual histogram for Foreman sequence at 15 Hz (DCT domain - DC coefficient band)

In Fig. 3.6, I verified the Laplacian assumption of the correlation between WZ frame and side information frame. By setting $\alpha = 0.12$, Laplace distribution provides an accurate approximation to the residual between the WZ frame and side information frame.

Moreover, the Bjontegaard delta metric (BJM) [46] is used to illustrate the average difference between two rate-distortion curves in terms of PSNR or bitrate. The Bjontegaard delta measurements of our proposed model and the DISCOVER model are given in Table 3.2, where the test is performed on all the sequences with different motion characteristics and difference quantization matrices. This metric shows that our proposed model outperforms the previous method in terms of bitrate saving.

Results comparing the relative performance of DISCOVER (with the correlation estimator of [16]), joint bit-plane PBP (our proposed transform-based codec with adaptive correlation estimation), joint bit-plane off-line (joint bit-plane decoding using true correlation statistics estimated off-line [16]) and joint bit-plane on-line (joint bit-plane decoding using correlation statistics estimated on-line [16]) codecs for the Soccer, Coastguard, Foreman, Hall & Monitor and Salesman video sequences, respectively, are shown in Figs. 3.7 to 3.11. Note that the off-line estimation method of [16] models the correlation noise as Laplacian distributed variable whose true Laplacian parameter is calculated off-line at the DCT-band/coefficient level for each frame using the residual between the WZ frame and the side information. This is impractical since in this case the encoder would need to perform side information generation. On the other hand, the on-line estimation method of [16] models the correlation noise as Laplacian distributed whose Laplacian parameter is estimated using the difference between backward and forward motion compensated frames at the decoder.

As expected, the joint bit-plane decoder (off-line) always achieves the best performance for all sequences (slow, median and fast motion), since it knows the true correlation statistics between source and side information. Most importantly, our proposed PBP-based codec consistently has the next best performance for all sequences, clearly outperforming the DISCOVER codec for all sequences since our PBP estimator iteratively refines the correlation statistics. I also note that the joint bit-plane setup also shows a better performance than that of the DISCOVER codec since each code bit may obtain more information from its neighboring bit-planes than the traditional

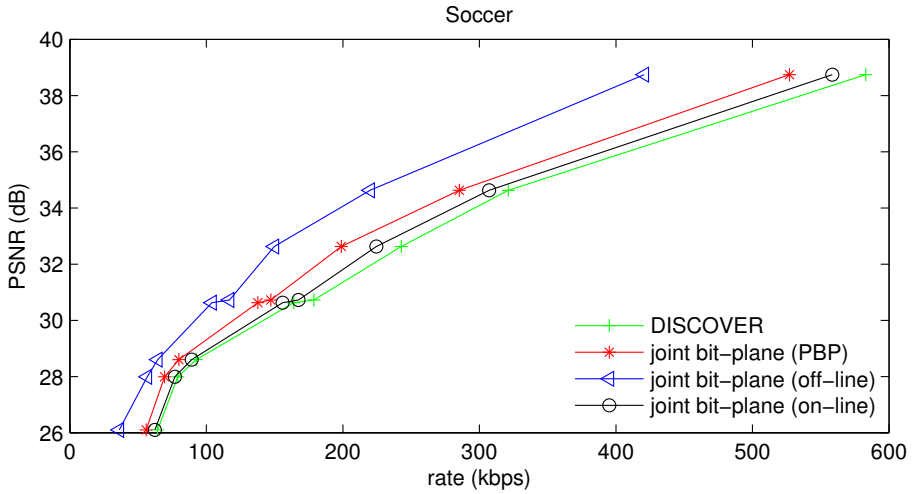


Figure 3.7: PSNR comparison of the proposed PBP joint bit-plane DVC for the QCIF Soccer sequence, compressed at 15 fps.

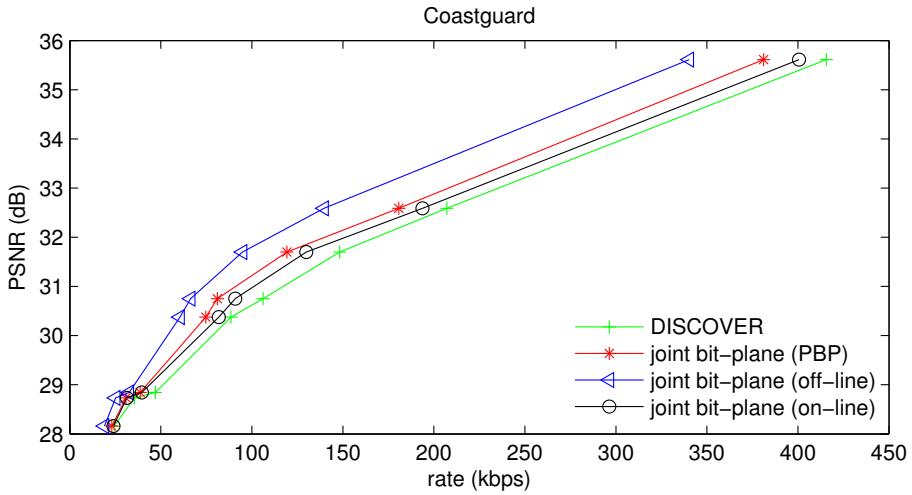


Figure 3.8: PSNR comparison of the proposed PBP joint bit-plane DVC for the QCIF Coastguard sequence, compressed at 15 fps.

separate bit-plane decoding setup[1].

The estimation accuracy is studied in Fig. 3.13. I can see that the proposed PBP algorithm improves the online estimate [16], which also explains

[1]Please note that the LDPC code length in the joint bit-plane decoder is M_b times longer than that of separate bit-plane decoder, where M_b is the quantization level of DCT band b. Since the decoding performance of LDPC code quite depends on the code length, the joint bit-plane decoder with longer code length is expected to outperform separate bit-plane decoder in the simulation. Another reason for this behavior is that the factor g_i in joint bit-plane decoder could obtain information from M_b variable nodes in Region III simultaneously, while such advantage is not available in separate bit-plane decoder.

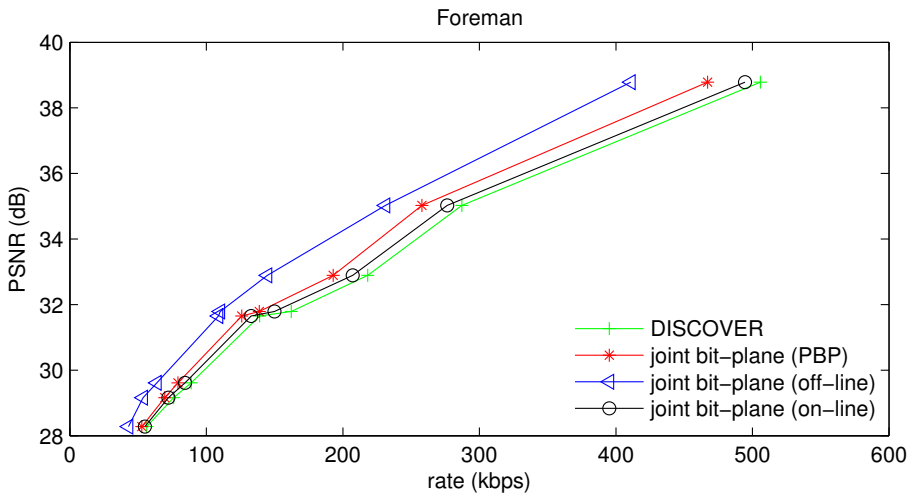


Figure 3.9: PSNR comparison of the proposed PBP joint bit-plane DVC for the QCIF Foreman sequence, compressed at 15 fps.

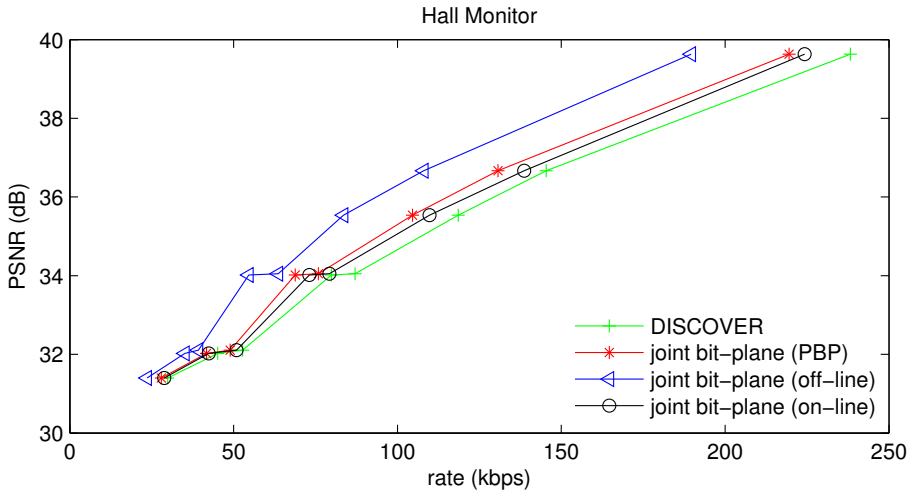


Figure 3.10: PSNR comparison of the proposed PBP joint bit-plane DVC for the QCIF Hall Monitor sequence, compressed at 15 fps.

why the proposed PBP algorithm outperforms DISCOVER codec. Furthermore, the algorithm complexity in terms of execution time is compared in Table 3.3 (will be added by tomorrow). The proposed PBP algorithm needs longer execution time to achieve a better decoding performance in terms of bitrate saving. Please note that since the proposed PBP algorithm is currently implemented by MATLAB incorporating JAVA, a shorter execution

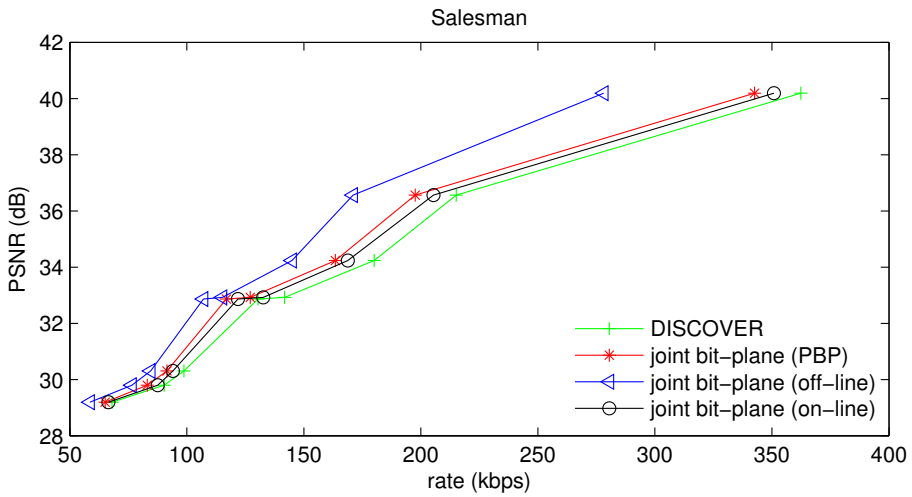


Figure 3.11: PSNR comparison of the proposed PBP joint bit-plane DVC for the QCIF Salesman sequence, compressed at 15 fps.

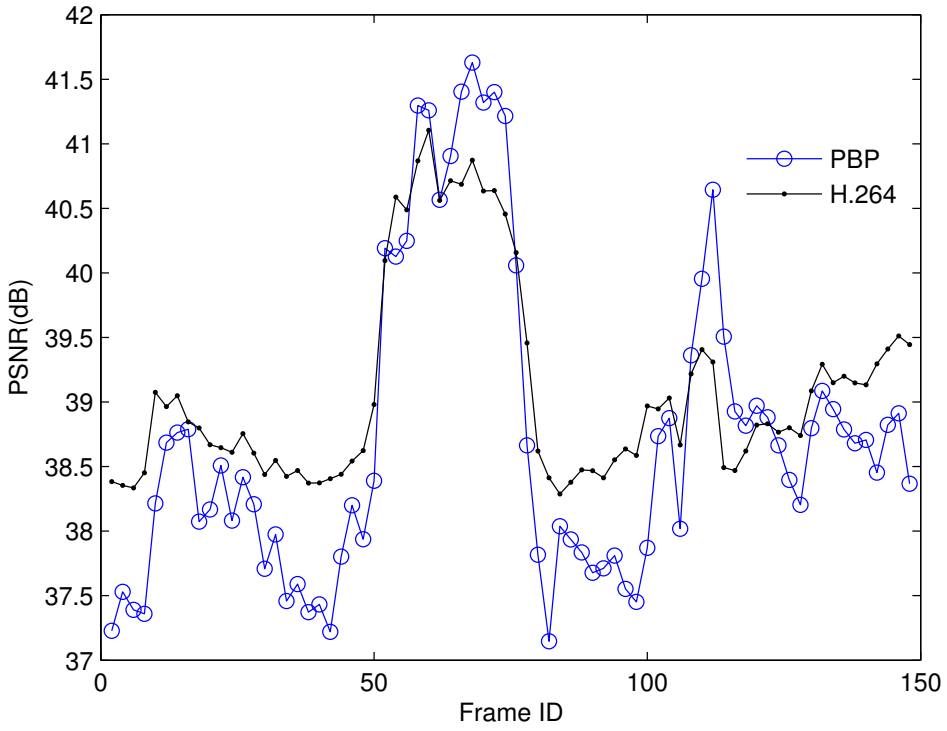


Figure 3.12: Frame-by-frame PSNR variance for Soccer sequence with quantization matrix Q8

time is expected in the future by using more efficient programming language (e.g. C++).

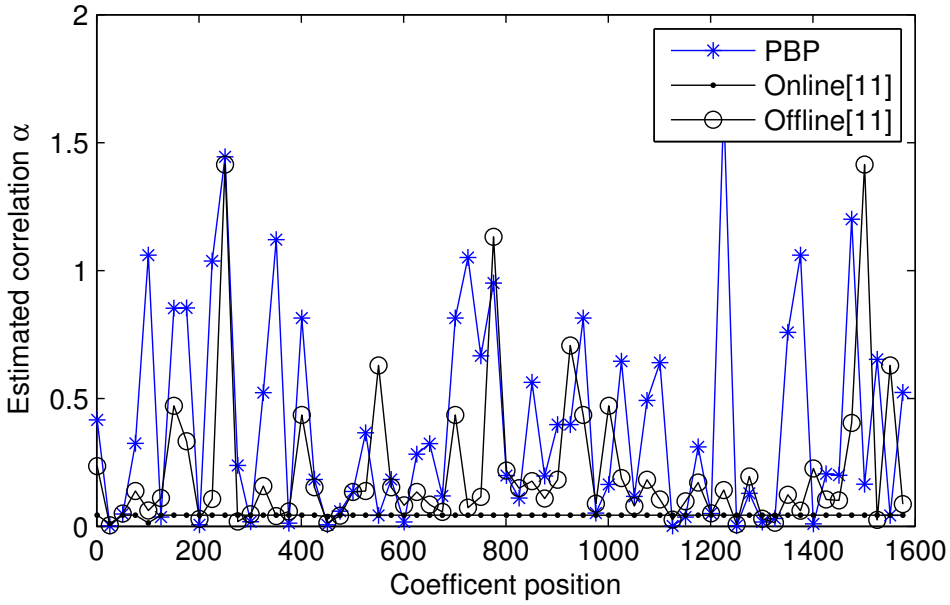


Figure 3.13: Estimation accuracy of proposed PBP method for the AC band of Soccer sequence.

Finally, a frame-by-frame PSNR variation for Soccer sequence with quantization matrix Q8 is shown in Fig.3.12. I found that the PSNR variation across frames for soccer sequence is about 4.48 dB for the proposed PBP codec and 2.81 dB for H.264 codec. Although, the PSNR variation of the proposed codec is slightly larger than that of H.264, the difference of average PSNR between H.264 and PBP codec is only 0.35 dB. Moreover, the result shows that the PSNR fluctuations of PBP and H.264 have similar trend and the maximum PSNR difference between them is about 1.33 dB.

3.4 Conclusion

This Chapter proposes an adaptive correlation estimation scheme for distributed video coding. Unlike current work in that direction, our proposed technique is embedded within the SW decoder itself, thus ensuring true dynamic tracking of correlation estimation taking into account the variance

Table 3.3: Execution Time (full sequence in seconds) of Joint bit-plane on-line, Joint bit-plane PBP codec vs. DISCOVER codec.

Sequences		DISCOVER	Joint bit-plane on-line	Joint bit-plane PBP
Foreman	Q1	347.9376	359.9462	1266.4
	Q3	457.0563	481.8287	3159.1
	Q5	508.5244	631.7438	6035.6
Soccer	Q1	581.0807	586.4243	2527.3
	Q3	615.0184	623.4325	4634.3
	Q5	657.3840	699.3597	6847.5

of side information. This is achieved by augmenting the SW code factor graph with correlation parameter variable nodes together with additional factor nodes that connect the SW graph with the correlation variable nodes. In our examples, correlation is modeled as Laplacian noise although I note that other correlation models including Gaussian may be used with minimal change to the factor graph. Inference on the graph with multiple connected regions can then be achieved with standard belief propagation (sum product algorithm) together with particle filtering that allows correlation parameter to take real values. The correlation variable nodes incorporate particles on which the particle filter operates, but also require joint operation with the BP algorithm which updates the weights of the particles. The proposed scheme boasts accurate correlation estimation together with ease of integration with existing DVC codecs, as all that is required is replacing the SW decoder block.

The results demonstrate the benefit of using the proposed scheme with a state-of-the-art transform-domain based DVC using adjacent H.264/AVC compressed frames to generate side-information through motion-compensated interpolation (MCI). Simulation results for a range of slow to fast motion se-

quences show significant performance improvement due to correlation tracking by our proposed PBP algorithm over state-of-the-art DVC codec with correlation estimation. The results also show that correlation statistics are accurately estimated online as the performance of our PBP algorithm closely (within 1-2dB PSNR) tracks that of the DVC codec which has offline knowledge of exact correlation statistics.

CHAPTER 4

LOW COMPLEXITY CORRELATION ESTIMATION USING EXPECTATION PROPAGATION

Although *on-the-fly correlation estimation* methods with particle filtering described in Chapter 3 usually outperforms online estimation techniques, stochastic approximation methods usually introduce a large computational cost at the decoder. Since deterministic approximation method (e.g. expectation propagation (EP)) provides low complexity and comparable estimation performance for modeling unimodal distribution, I will investigate deterministic approximation method (e.g., EP) for the on-the-fly correlation estimation in DVC in this Chapter.

4.1 Related Work of Correlation estimation

Since the SW decoding process refines starting beliefs, our prior works [17,18] demonstrated that unifying the process of correlation estimation using sampling method and joint bit-plane decoding into a single joint process (i.e., OTF estimation mode) can provide better statistics estimate and consequently improved performance for both pixel- and transform-domain DVCs. Additionally, this unification of correlation estimation and SW decoding will also enable the correlation estimator to take into account side information statistics and any of the methods of [16, 37–40] can be used as an initial point that will be refined during SW decoding. While OTF estimation is also discussed in our prior works [17, 18], our prior schemes were based on the sampling method for dynamically tracking the correlation statistics,

which result a large computational complexity at the decoder side. Instead of incorporating particle filtering method into BP algorithm for correlation tracking [17, 18], an ultra-low complexity alternative (i.e. EP algorithm) is studied and proposed in this Chapter.

4.2 System Architecture

To precisely catch the correlation between frames while recovering source frames, we proposed an adaptive DVC framework. In Bayesian perspective, capturing correlation corresponds to estimating the posterior distribution of correlation parameter. Since a factor graph, as a particular type of graphical model, enables efficient computation of marginal distributions through message passing algorithm, our proposed framework is carried out on the factor graph as shown in Fig. 4.1. The key steps of the proposed adaptive DVC framework can be outlined as follows: 1) factor graph construction: design a factor graph with appropriately defined factor functions to capture and connect SW coding and correlation tracking (see Section 4.2.1); 2) message passing algorithm implementation: perform message passing algorithm on the constructed factor graph to calculate the posterior distribution of interested variables (see Section 4.2.2).

4.2.1 Factor graph construction

DVC, a video compression technology based on DSC principle, is usually implemented on a factor graph utilizing WZ coding scheme. Compared with standard DVC, the factor graph (see Fig. 4.1) of the proposed adaptive DVC with correlation tracking consists of two regions, where Region I refers to the correlation parameter tracking and Region II corresponds the traditional WZ

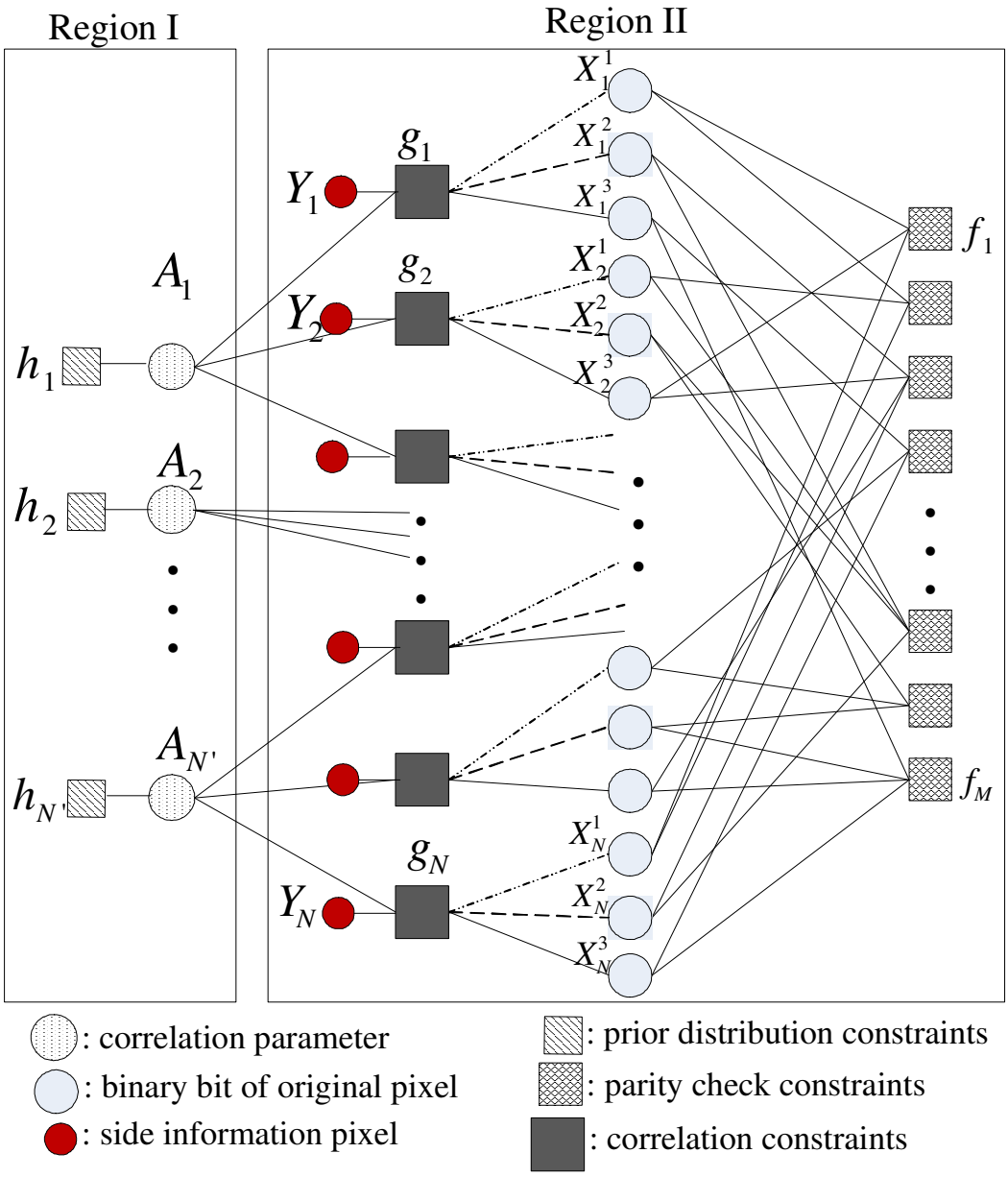


Figure 4.1: Factor graph of joint bit-plane SW decoding with correlation estimation.

coding. In Fig. 4.1, variable nodes (usually depicted by a circle) denote unknown variables such as coded bits, correlation parameter, and factor nodes (depicted by small squares) represent the relationship among the connected variable nodes.

Joint bit-plane SW coding (Region II)

WZ coding, a.k.a. the lossy version of SW coding, is usually realized by quantization followed by SW coding of the quantized indices based on channel coding [12]. Here, for WZ coding, we carry out LDPC based joint bit-plane SW coding after performing quantization, the factor graph of which is described as Region II in Fig. 4.1.

Note that we suppose a N-length source sample x_i, $i = 1, \cdots, N$ is quantized into $Q[x_i]$ using 2^q levels quantization, where $q = 3$ is taken as an example in Region II of Fig.4.1. We denote $x_i^1, x_i^2, \cdots, x_i^q$ as the binary format of the quantization index $Q[x_i]$, and denote $\mathbf{B} = x_1^1, x_1^2, \cdots, x_1^q, x_2^1, x_2^2, \cdots, x_N^q$ as the block which combines all the bit variables together. The block $\mathbf{B}$ is then encoded using LDPC-based SW codes and generates an M-length syndrome bits $\mathbf{S} = s_1, s_2, \cdots, s_M$, which results in a $qN : M$ SW compression ratio.

Similar to the standard LDPC decoding, the factor nodes $f_1, f_2, \cdots, f_M$ in Region II take into account the constraints imposed by the received syndrome bits. Thus, the factor function of factor node $f_a, a = 1, \ldots, M$ is defined as

$$f_a(\tilde{\mathbf{x}}_\mathbf{a}, s_a) = \begin{cases} 1, & \text{if } s_a \oplus \bigoplus \tilde{\mathbf{x}}_\mathbf{a} = 0, \\ 0, & \text{otherwise.} \end{cases} \tag{4.1}$$

where $\tilde{\mathbf{x}}_\mathbf{a}$ denotes the set of neighbors of factor node f_a, and $\bigoplus \tilde{\mathbf{x}}_\mathbf{a}$ denotes the binary sum of all elements of the set $\tilde{\mathbf{x}}_\mathbf{a}$.

Let a N-length sample y_i, $i = 1, \cdots, N$, the realizations of variable nodes Y_i, be the side information available at the decoder. The factor nodes g_i are introduced in the factor graph to capture the correlation constraints as

shown in (2.4) between source x_i and side information y_i for SW decoding. Since source samples are first passed through a quantization process, the correlation constraints (i.e., the factor function of g_i) between the quantized indices $Q[x_i]$ and the side information y_i can be expressed as:

$$g_i(Q[x_i], y_i, \lambda) = \int_{P(Q[x_i])}^{P(Q[x_i]+1)} \frac{\lambda}{2} e^{\lambda|x-y_i|} dx, \tag{4.2}$$

where λ is the correlation parameter[1] of the Laplace distribution, $P(\bullet)$ denotes the lower boundary of quantization partition at index "$\bullet$", e.g. if a coefficient x_i satisfies $P(\bullet) \leq x_i < P(\bullet + 1)$, the quantization index $Q[x_i]$ of coefficient x_i is equal to "$\bullet$". Actually, given a parameter λ, the factor node g_i plays a role of providing a predetermined likelihood $p(y_i|Q[x_i], \lambda)$ to variable node X_i^j, $j = 1, \cdots, q$ for LDPC based SW decoding.

Correlation parameter tracking (Region I)

As described in previous Section, the correlation parameter, denoted by λ_l^t, can vary along both time and space, where the superscript t indicates the time dependence (i.e., frame) and the subscript l corresponds to the location of a single pixel or a group of pixels (i.e., block). In this Chapter, for the simplicity of notation, we drop the superscript t and denote λ_l as the correlation of the l-th pixel block in a given frame, where each pixel block possesses C number of pixels. Then, the correlation parameter λ in (4.2) can be replaced by λ_l for different pixel block.

Let us denote by N' the number of pixel blocks within a frame. Then, we introduce additional variable nodes A_l, $l = 1, 2, \ldots, N'$ to represent the

[1]In this chapter, we use λ instead of α to denotet he correlation parameter, where α is reserved for other purpose.

correlation parameters λ_l in factor graph (see, Region I of Fig. 4.1). Since a block of C source samples (i.e., pixels) share the same correlation parameter, every C number of factor nodes g_i in Region II will be connected to the same variable node A_l, where we call C [2] as the connection ratio. Moreover, to initialize a prior distribution for correlation parameter λ_l, additional factor nodes h_l, $l = 1, \cdots, N'$ are introduced, where Gamma distribution is assigned to each factor function $h_l(\lambda_l)$ for the mathematical convenience. Then, by implementing message passing rules introduced in the next subsection on the proposed factor graph, each factor node g_i will periodically update the likelihood $p(y_i|Q[x_i], \lambda_l)$ for the corresponding bit variable nodes $X_i^1, X_i^2, \cdots, X_i^q$ when a new estimate of correlation parameter λ_l is available, instead of using a predetermined likelihood $p(y_i|Q[x_i], \lambda)$.

Consequently, by introducing correlation parameter estimation in Region I, likelihood factor function in (4.2) will be updated as

$$g_i(Q[x_i], y_i, \lambda_l) = \int_{P(Q[x_i])}^{P(Q[x_i]+1)} \frac{\lambda_l}{2} e^{\lambda_l |x - y_i|} dx. \tag{4.3}$$

4.2.2 Message passing on the constructed factor graph

In Bayesian inference, message passing algorithm (e.g., BP) on a factor graph offers an very efficient way to calculate the marginal distributions (i.e. beliefs) of the unknown variables represented by their corresponding variable nodes. In the proposed adaptive DVC factor graph (see Fig. 4.1), we are interested in two unknown variables, which are represented by source variable nodes X_i^j in Region II and correlation parameter variable nodes A_l in Region I,

[2] To estimate a stationary correlation parameter, we can set the connection ratio equal to the code length. Moreover, connection ratio provides a trade-off between complexity and spatial variation.

respectively.

In Region II, without considering the connection to the Region I, the factor graph is identical to that of standard LDPC codes with discrete variables x_i^j. Hence, the posterior distribution of x_i^j can be calculated through standard BP algorithm. However, in region I, BP algorithm cannot be applied directly, as the correlation parameter λ_l represented by the variable node A_l is generally non-Gaussian continuous variable and BP algorithm only handles discrete variable with small alphabets size or continuous variable with linear Gaussian distribution.

To seek a workaround for this difficulty, let us start with the derivation of posterior distribution of the correlation parameter λ_l. According to Bayes' rule and the message passing rule, the posterior distribution of correlation parameter λ_l can be expressed as:

$$\begin{aligned} p(\lambda_l|\mathbf{y}_l) &= \frac{1}{Z_l} \prod_{i \in \mathcal{N}^{\backslash h_l}(A_l)} p(\lambda_l) p(y_i|\lambda_l) \\ &= \frac{1}{Z_l} \prod_{i \in \mathcal{N}^{\backslash h_l}(A_l)} \int_{Q[x_i]} p(\lambda_l) p(Q[x_i]) p(y_i|Q[x_i];\lambda_l) \\ &= \frac{1}{Z_l} h(\lambda_l) \prod_{i \in \mathcal{N}^{\backslash h_l}(A_l)} \sum_{\mathbf{x}_i^q} g(y_i; Q[x_i], \lambda_l) \prod_{j \in 1,2,\cdots,q} m_{X_i^j \to g_i}(x_i^j) \\ &= \frac{1}{Z_l} m_{h_l \to A_l}(\lambda_l) \prod_{i \in \mathcal{N}^{\backslash h_l}(A_l)} m_{g_i \to A_l}(\lambda_l), \end{aligned} \tag{4.4}$$

where Z_l is a normalization constant, $\sum_{\mathbf{x}_i^q}$ denotes a sum over all the bit variables in $\mathbf{x}_i^q$, the value of message $m_{X_i^j \to g_i}(x_i^j)$ is updated iteratively by variable node X_i^j in Region II according to BP update rule, message $m_{h_l \to A_l}(\lambda_l) = h(\lambda_l)$ comes from prior factor node in Region I, and message $m_{g_i \to A_l}(\lambda_l) =$

$\sum_{\mathbf{x}_i^q} g(y_i; Q[x_i], \lambda_l) \prod_{j\in 1,2,\cdots,q} m_{X_i^j \to g_i}(x_i^j)$ comes from likelihood factor node in Region II according to the BP update rule.

So far, we have shown that the posterior distribution of correlation variable λ_l can be expressed as the product of all the incoming messages. In the rest of this subsections, we investigate how to efficiently compute the posterior distribution using BP and EP based approximation algorithms.

Belief propagation

The BP algorithm [47] is an efficient and exact inference algorithm for computing local marginals over variables on tree-structured graphs. For graphs with loops, a lot of applications (e.g. LDPC deocoding [48]) show that BP algorithm (or loopy BP algorithm) still provides a good performance. While this technique is extremely powerful in handling variables of small alphabet sizes, they cannot handle a continuous variable with arbitrary distribution or even a variable of a medium alphabet size as the computational complexities of these algorithms increase exponentially with the alphabet size.

For our problem in (4.4), since all the bit variables x_i^j, $j = 1, \cdots, q$, in $\mathbf{x}_i^q$ are discrete and taking values 0 or 1, the message $m_{g_i \to A_l}(\lambda_l) = \sum_{\mathbf{x}_i^q} g(y_i; Q[x_i], \lambda_l) \prod_{j\in 1,2,\cdots,q} m_{X_i^j \to g_i}(x_i^j)$ has 2^q terms and the product of all the messages $\prod_{i\in \mathcal{N}^{\setminus h_l}(A_l)} m_{g_i \to A_l}(\lambda_l)$ is a mixture of 2^{qC} number of Laplace distributions, where $C = |\mathcal{N}^{\setminus h_l}(A_l)|$ is the connection ratio, q is the number of bit-planes, and qC can be a large number. Thus, the direct evaluation of the posterior distribution using BP would be infeasible.

Expectation propagation

An approximate inference for solving the problem in (4.4) is to parametrize the variables through variational inference. Deterministic approximation schemes (e.g. EP [49]) provide some low complexity alternatives based on the analytical approximations to the posterior distribution. For example, suppose that posterior distribution $p(\theta)$ of parameter θ is infeasible to be calculated directly. If the posterior can be factorized as $p(\theta) = \prod_k g_k(\theta)$, where each factor function $g_k(\theta)$ only depends a small subset of observations, EP solves this difficulty by replacing the true posterior distribution $p(\theta)$ with an approximate distribution $q(\theta) = \prod_k \tilde{g}_k(\theta)$ by sequentially computing each approximate term $\tilde{g}_k(\theta)$ for $g_k(\theta)$. The general workflow of the EP algorithm has been listed in Table 4.1. In particular, for our problem in (4.4), EP is used to sequentially compute approximate messages $\tilde{m}_{h_l \to A_l}(\lambda_l)$ and $\tilde{m}_{g_i \to A_l}(\lambda_l)$ in replace of true messages $m_{h_l \to A_l}(\lambda_l)$ and $m_{g_i \to A_l}(\lambda_l)$ in (4.4), then get an approximate posterior on λ_l by combining these approximations together. The details of correlation parameter estimation through EP for our problem will be discussed in the next section.

4.3 Posterior approximation of correlation parameter using Expectation propagation

In this section, we will derive the proposed EP based correlation estimator, which can provide a fast and accurate way to approximate the posterior distribution on the factor graph as shown in Fig. 4.1. The procedures of the proposed EP algorithm has been detailed as follows:

Table 4.1: Expectation Propagation

Initialize the term approximation $\tilde{g}_k(\theta)$ and $q(\theta) = \frac{1}{Z}\prod_{k=1}^{C}\tilde{g}_k(\theta)$, where $Z = \int_\theta \prod_{k=1}^{C}\tilde{g}_k(\theta)$
repeat
 for $k = 1, \cdots, C$ **do**
  Compute $q^{\backslash k}(\theta) \propto q(\theta)/\tilde{g}_k(\theta)$
  Minimize Kullback Leibler (KL) divergence between $q(\theta)$ and $g_i(\theta)q^{\backslash k}(\theta)$ by performing
  moment matching
  Set approximate term $\tilde{g}_k(\theta) \propto q(\theta)/q^{\backslash k}(\theta)$
 end for
until parameters converged

1. Initialize the prior term

$$h_l(\lambda_l) = \text{Gamma}(\lambda_l, \alpha_l^0, \beta_l^0) = z_l^0 \lambda_l^{\alpha_l^0 - 1}\exp(-\beta_l^0\lambda_l) \tag{4.5}$$

   with $\alpha_l^0 = 2$, $\beta_l^0 = \frac{\alpha_l^0 - 1}{\lambda^0}$, $z_l^0 = \frac{{\beta_l^0}^{\alpha_l^0}}{\Gamma(\alpha_l^0)}$, where λ^0 is the initial correlation parameter, and β_l^0 and α_l^0 are scale and shape parameters for Gamma distribution, respectively. The selection of the initial values for the above parameters guarantees the mode of prior distribution equals to the initial correlation λ^0.

2. Initialize the approximation term (uniform distribution)

$$\tilde{m}_{g_i \to A_l}(\lambda_l) = \text{Gamma}(\lambda_l, \alpha_{il}, \beta_{il}) = z_{il}\lambda_l^{\alpha_{il}-1}\exp(-\beta_{il}\lambda_l) \tag{4.6}$$

   with $\beta_{il} = 0$, $\alpha_{il} = 1$, $z_{il} = 1$.

3. Initialize α_l^{new} and β_l^{new} for approximate posterior $q(\lambda_l) = \text{Gamma}(\lambda_l, \alpha_l^{\text{new}}, \beta_l^{\text{new}})$, where $\alpha_l^{\text{new}} = \alpha_l^0 = 2$, and $\beta_l^{\text{new}} = \beta_l^0$.

4. For each variable node λ_l

   For each factor node g_i, where $g_i \in \mathcal{N}(\lambda_l)$

   (a) Remove $\tilde{m}_{g_i \to A_l}(\lambda_l)$ from the posterior $q(\lambda_l)$, we get $q^{\setminus g_i}(\lambda_l) = \text{Gamma}(\alpha_l^{\text{tmp}}, \beta_l^{\text{tmp}})$

$$\begin{aligned} \alpha_l^{\text{tmp}} &= \alpha_l^{\text{new}} - (\alpha_{il} - 1) \\ \beta_l^{\text{tmp}} &= \beta_l^{\text{new}} - \beta_{il} \end{aligned} \tag{4.7}$$

   (b) Update $q^{\text{new}}(\lambda_l)$ by minimizing the Kullback Leibler (KL) divergence $D(q^{\setminus g_i}(\lambda_l) m_{g_i \to A_l}(\lambda_l) || q^{\text{new}}(\lambda_l))$ (i.e., performing moment matching (**Proj**)) (see Section 4.3.1 for detail).

$$q^{\text{new}}(\lambda_l) = \frac{1}{Z_l} \mathbf{Proj}[q^{\setminus g_i}(\lambda_l) m_{g_i \to A_l}(\lambda_l)] \tag{4.8}$$

   where $Z_l = \int_{\lambda_l} q^{\setminus g_i}(\lambda_l) m_{g_i \to A_l}(\lambda_l)$.

   (c) Set approximated message

$$\begin{aligned} \alpha_{il} &= \alpha_l^{\text{new}} - (\alpha_l^{\text{tmp}} - 1) \\ \beta_{il} &= \beta_l^{\text{new}} - \beta_l^{\text{tmp}} \\ z_{il} &= Z_l \frac{{\beta_l^{\text{new}}}^{\alpha_l^{\text{new}}}}{\Gamma(\alpha_l^{\text{new}})} \left(\frac{{\beta_l^{\text{tmp}}}^{\alpha_l^{\text{tmp}}}}{\Gamma(\alpha_l^{\text{tmp}})} \right)^{-1} \left(\frac{{\beta_{il}}^{\alpha_{il}}}{\Gamma(\alpha_{il})} \right)^{-1} \end{aligned} \tag{4.9}$$

4.3.1 Moment matching

Through moment matching, $q(\lambda_l)$ is obtained by matching the mean and variance of $q(\lambda_l)$ to those of $q^{\setminus g_i}(\lambda_l) m_{g_i \to A_l}(\lambda_l)$. Then, we get the updated

α_l^{new} and β_l^{new}, the parameters of $q(\lambda_l)$ as follows,

$$\begin{aligned} \alpha_l^{\text{new}} &= m_1 \beta_l^{\text{new}} \\ \beta_l^{\text{new}} &= m_1/(m_2 - m_1^2), \end{aligned} \tag{4.10}$$

where m_1 and m_2 are the first and second moments of the approximate distribution as shown below.

$$\begin{aligned} m_1 &= \frac{1}{Z} \sum_{Q[x_i]} (\mathcal{F}_1(z_1) - \mathcal{F}_1(z_2)) \prod_{j=1}^{j=q} m_{X_i^j \to g_i}(x_i^j) \\ m_2 &= \frac{1}{Z} \sum_{Q[x_i]} (\mathcal{F}_2(z_1) - \mathcal{F}_2(z_2)) \prod_{j=1}^{j=q} m_{X_i^j \to g_i}(x_i^j) \\ Z &= \sum_{Q[x_i]} (\mathcal{F}_0(z_1) - \mathcal{F}_0(z_2)) \prod_{j=1}^{j=q} m_{X_i^j \to g_i}(x_i^j) \end{aligned} \tag{4.11}$$

Here, Z is the normalization term and these unknown functions in (4.11) can be evaluated according to (4.12).

$$\begin{aligned} z_1 &= P(Q[x_i] + 1) \\ z_2 &= P(Q[x_i]) \\ \mathcal{F}_0(z) &= A(z) + B(z) \\ \mathcal{F}_1(z) &= \frac{\alpha^{\text{tmp}}}{\beta^{\text{tmp}}} A(z) + \frac{\alpha^{\text{tmp}}}{\beta^{\text{tmp}} + |z - y_i|} B(z) \\ \mathcal{F}_2(z) &= \frac{\alpha^{\text{tmp}}(\alpha^{\text{tmp}} + 1)}{(\beta^{\text{tmp}})^2} A(z) + \frac{\alpha^{\text{tmp}}(\alpha^{\text{tmp}} + 1)}{(\beta^{\text{tmp}} + |z - y_i|)^2} B(z) \\ A(z) &= \frac{1}{2}\left(1 + \text{sgn}(z - y_i)\right) \\ B(z) &= -\text{sgn}(z - y_i) \frac{1}{2} \left(\frac{\beta^{\text{tmp}}}{\beta^{\text{tmp}} + |z - y_i|} \right)^{\alpha^{\text{tmp}}} \end{aligned} \tag{4.12}$$

4.4 Results

In this section, we employ a pixel-based DVC setup to demonstrate the benefit of the proposed OTF correlation tracking. As in references [1,16,33], group of pictures (GOP) is equal to 2 in our study, where all even frames are treated as WZ frames and all odd frames are considered as key frames. The key frames are conventionally intra-coded, for example, using H.264 Advanced Video Coding (AVC) [50] intra coding mode. WZ frames are first quantized pixel-by-pixel and then all bit-plane of the resulting quantization indices are combined together and compressed using an LDPCA codes. At the decoder, side information frame Y is generated using motion-compensated interpolation of the forward and backward key frames [1, 33]. Spatial smoothing [33], via vector-median filtering, is used to improve the result together with half-pixel motion search. Each WZ frame is decoded by the proposed EP based OTF WZ decoder described in Section 4.2. Moreover, we incorporate LDPCA codes with a feedback channel for rate adaptive decoding.

To verify the effectiveness of correlation tracking across WZ-encoded frames in a video sequence, we tested the above set-up with three standard QCIF (i.e., 176×144) video sequences, carphone, foreman and soccer, with different scene dynamics of low, medium and high motions, respectively. All the results are based on the average of 50 WZ frames. The quantization parameters Q of H.264/AVC encoder for different video sequences with different WZ quantization bits q have been listed in Table 4.2. The selections of quantization parameters Q for different WZ quantization bits q make sure that both the decoded key and WZ frames have similar visual qualities in terms of PSNR. Moreover, we split each 176×144 WZ frame into 16 sub-frames with size 44×36 (i.e. $N = 1584$) for efficient coding purpose. Within

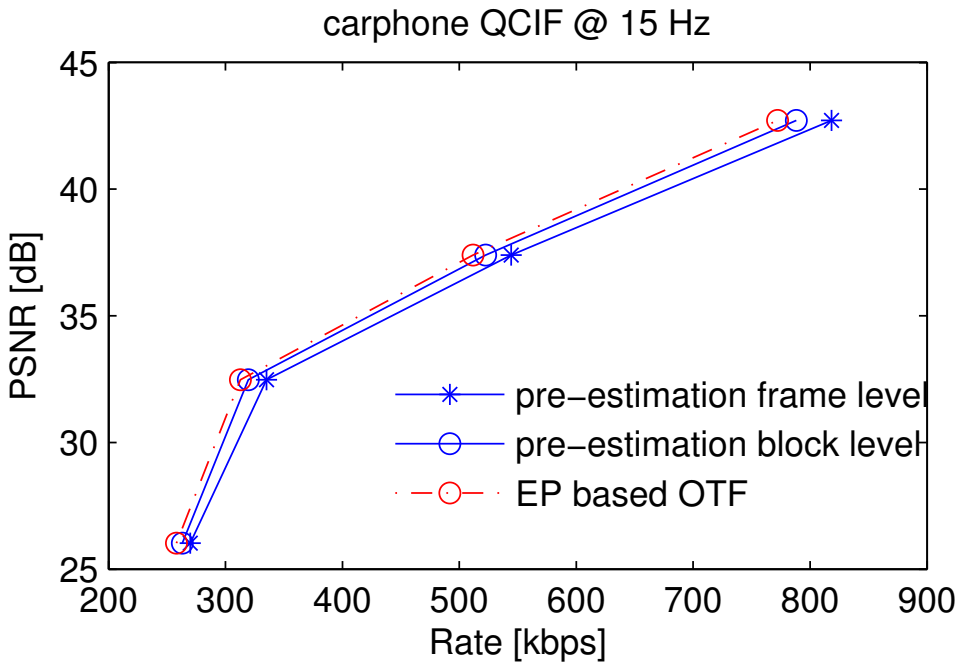


Figure 4.2: PSNR comparison of the proposed EP based OTF and pre-estimation DVC for the QCIF carphone sequence, compressed at 15 fps.

each sub-frame, the block size for correlation estimation is equal to 4×6 (i.e., $C = 24$) for total $N' = 66$ number of blocks.

Table 4.2: H.264/AVC quantization parameter Q for different video sequences

Quantization bits	Carphone	Foreman	Soccer
	Q	Q	Q
2	46	46	44
3	36	36	34
4	28	28	26
5	22	21	19

Results comparing the relative performance of pre-estimation in frame level DVC [16], pre-estimation in block level DVC [16], and the proposed EP based OTF DVC for the carphone, foreman, and soccer video sequences, respectively, are shown in Figs. 4.2, 4.3, 4.4, where the implementation of

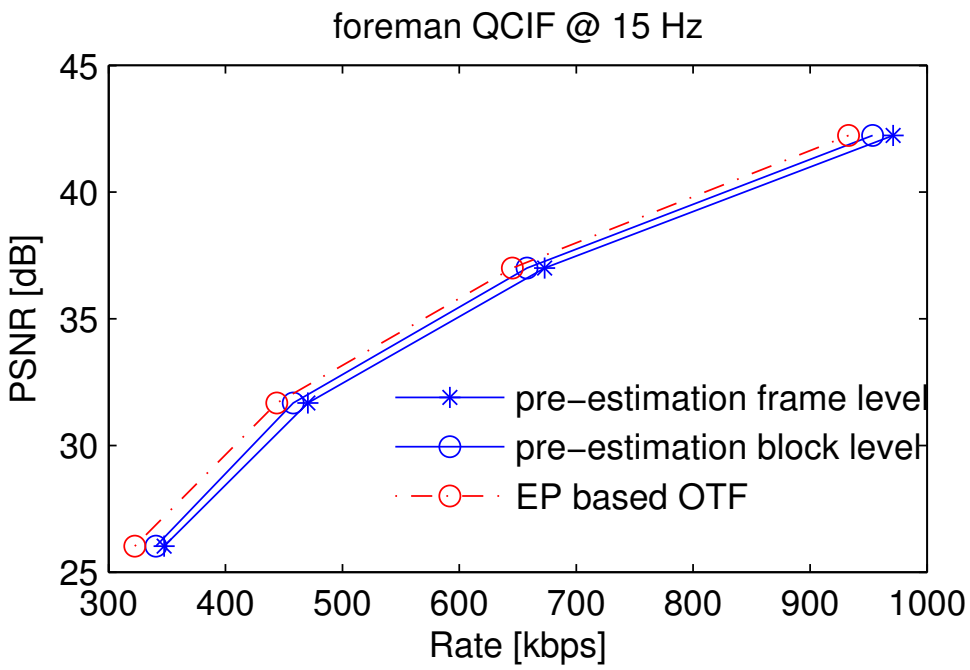


Figure 4.3: PSNR comparison of the proposed EP based OTF and pre-estimation DVC for the QCIF foreman sequence, compressed at 15 fps.

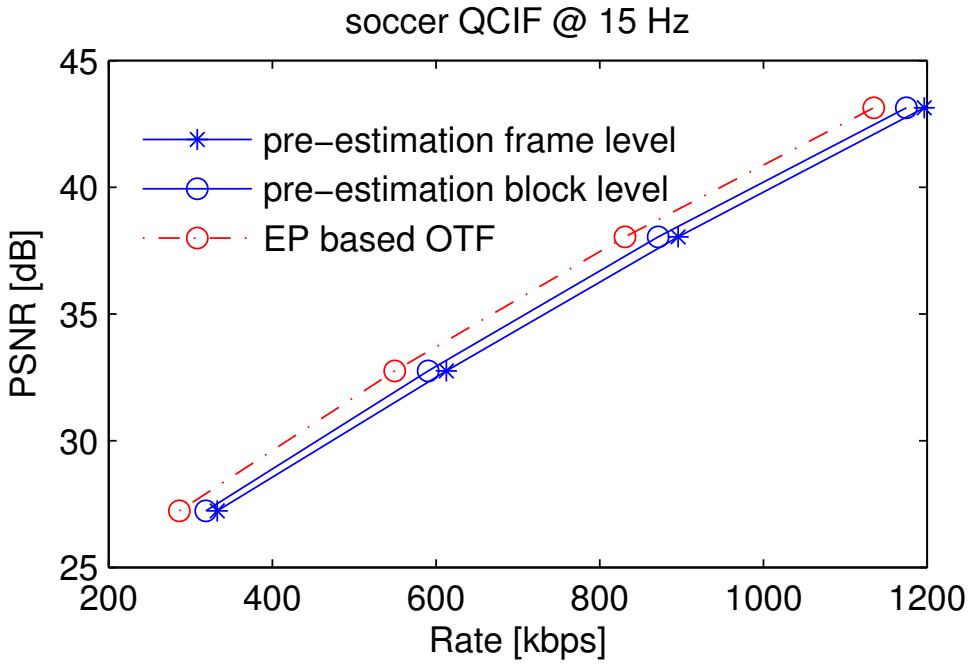


Figure 4.4: PSNR comparison of the proposed EP based OTF and pre-estimation DVC for the QCIF soccer sequence, compressed at 15 fps.

standard DVC codec is based on the DISCOVER framework [16] with joint bit-plane setup.

The pre-estimation methods [16], either in frame or block levels, model the correlation as Laplace distribution, whose correlation parameter is estimated using the difference between backward and forward motion compensated frames/blocks at the decoder. Our proposed OTF estimator unifies the process of correlation estimation using EP and joint bit-plane decoding into a single joint process, where the updated decoding information can be used to improve the correlation estimation and vice versa. As expected, pre-estimation in block level has better performance than that of frame level in terms of bit rate saving, since block level correlation offers a finer granularity than the frame level correlation. More importantly, our proposed EP based OTF codec always achieves the best performance for all sequences (slow, medium and fast motions), since the proposed EP based OTF estimator can iteratively refine the correlation statistics in each block.

In particular, for the carphone sequence (slow motion), to obtain the same visual qualities (i.e., PNSRs), our proposed EP based OTF codec achieves about 10 kbps and 30 kbps saving compared to pre-estimation in block and frame levels, respectively. For foreman sequence (medium motion), the average rate decrease of EP based OTF codec is 20 kbps for pre-estimation in block level and 30 kbps for pre-estimation in frame level. Moreover, for soccer sequence with fast motion, we again observe the superiority of our proposed EP based OTF codec over the pre-estimation codecs, where the proposed EP based OTF codec offers about 38.25 kbps and 56.5 kbps saving compared to pre-estimation in block and frame levels, respectively. These results demonstrated that our proposed EP based OTF codec are more powerful for video

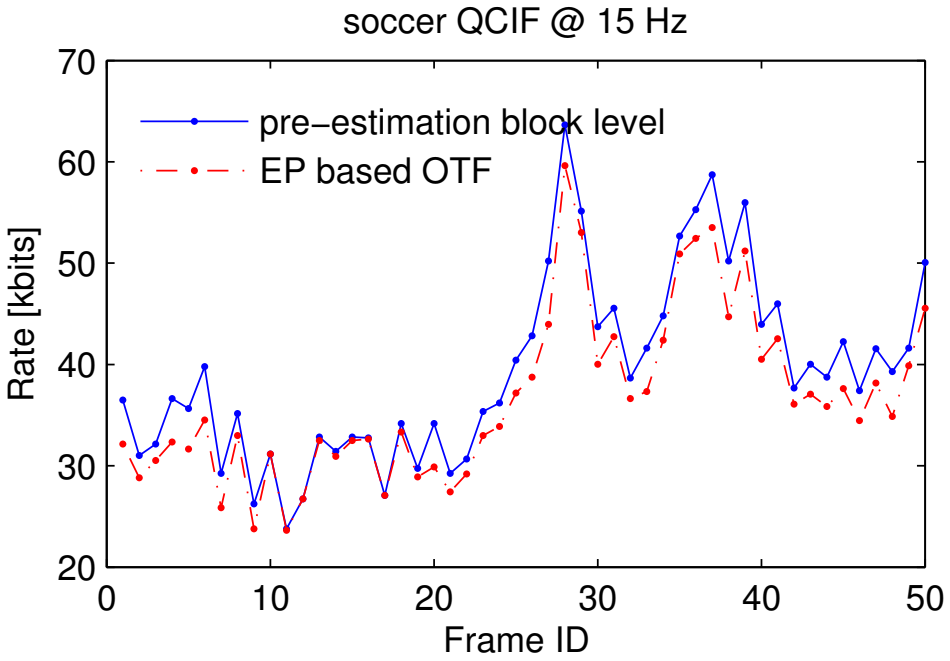


Figure 4.5: Subframe-by-subframe rate variance for soccer sequence with quantization bits equal to 3.

sequences with fast motions.

A sub-frame-by-sub-frame (i.e., the first sub-frame of each WZ frame) rate variation for the soccer sequence with quantization bits equal to 3 is shown in Fig. 4.5. We found that the rate variation across frames is about 36.01 kbits for the proposed EP based OTF DVC codec and 40.01 kbits for DVC codec with pre-estimation in block level. Moreover, the result shows that the rate fluctuations of EP based OTF and pre-estimation in block level DVC codecs have similar trend and the proposed EP based OTF codec always has equal or lower code rate than that of the pre-estimation in block level DVC codec. The maximum difference of code rate between EP based OTF and pre-estimation block level codecs is about -5.32 kbits. Similar results are obtained for other sub-frames in all three testing sequences.

The estimation accuracy of correlation parameter is studied in Fig. 4.6 for

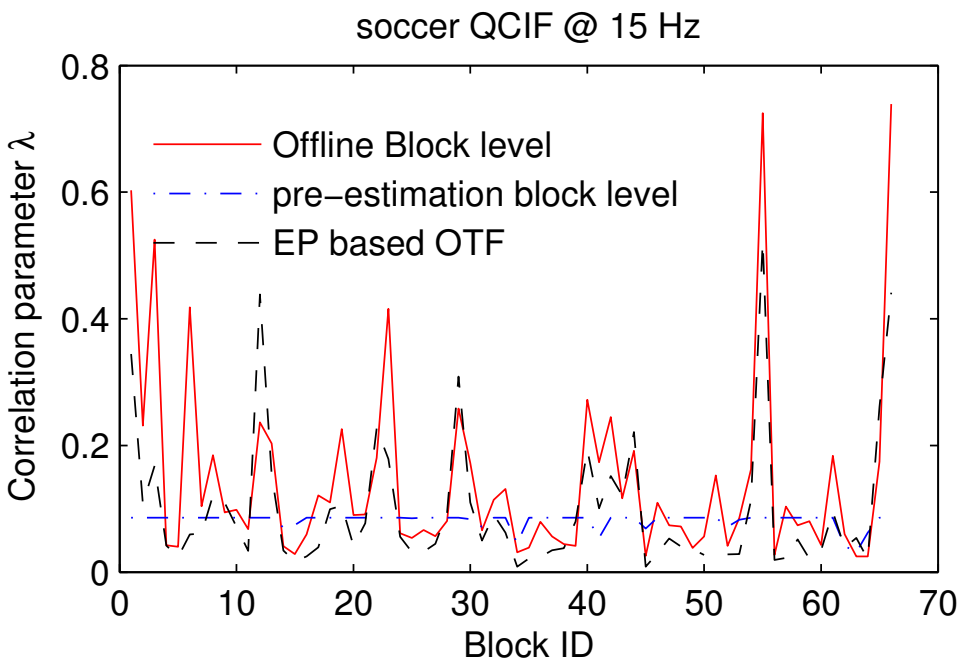


Figure 4.6: Estimation accuracy of the proposed EP based OTF DVC for the correlation parameter of the soccer sequence

the soccer sequence. Here, we use the offline estimated correlation parameter as benchmark, where the benchmark Laplacian parameter is calculated offline at the block level for each frame using the residual between the WZ frame and the side information frame. We can see that the proposed OTF correlation estimation scheme improves the estimates obtained through pre-estimation method [16], which also explains why the proposed EP based OTF DVC outperforms the pre-estimation based DVC codec.

Finally, the proposed EP based OTF estimator offers a very low complexity overhead compared with the standard BP algorithm. The complexity of the proposed estimator lies in the evaluation of equations (4.10), (4.11) and (4.12) as shown in Section 4.3.1. Roughly speaking, the EP based OTF estimator introduces less than 10% computational overhead compared with the standard BP algorithm.

4.5 Conclusion

This Chapter proposes an on-the-fly (OTF) correlation estimation scheme for distributed video coding using expectation propagation (EP). Unlike previous work performing pre-estimation where estimation starts before decoding, our proposed correlation estimation technique is embedded within the WZ decoder itself, thus ensuring dynamic estimation of correlation changes in block level. This is achieved by augmenting the SW code factor graph to connect correlation parameter variable nodes together with additional factor nodes. Inference on the factor graph for continuous correlation parameter variable is achieved through EP based deterministic approximation methods, which offers better tradeoff between accuracy and complexity compared with other methods. The proposed scheme boosts coding performance together with the ease of integration with existing DVC codecs. We demonstrate the benefits of using the proposed scheme via a pixel-based DVC setup. Simulation results show significant performance improvement due to correlation tracking for multiple video sequences with the Laplacian correlation model.

CHAPTER 5

SECURE DISTRIBUTED IMAGE CODING

5.1 Introduction

The introduction of the Virtual Lifetime Electronic Record (VLER) [51] by the Department of Veterans Affairs (VA) indicates a new era of electronic healthcare. VLER contains not only a veteran's administrative (i.e., personnel and benefits) information but also the complete medical record throughout his/her entire military career and beyond. The next phase of electronic healthcare intends to make patient information available securely, through the Nationwide Health Information Network (NHIN), to private and public healthcare providers for military officers [52]. This activity significantly improves the healthcare performance as well as increases the flexibility and convenience for sharing patient diagnoses. Ultimately, the service is expected to be provided to all residents in the US. For example, president Obama has proposed a massive effort to push the digital revolution in health records. Moreover, the FDA has approved an iPad/iPhone radiology app for mobile diagnoses, which means mobile applications may become a big potential market for cloud-based private data sharing and healthcare.

All these efforts lead to the explosion of medical data transmission over the network, which creates the urgent need for ensuring both efficiency and confidentiality in the process. On one hand, since medical data contains sensitive information about personal privacy, privacy protection has become a crucial issue which requires that medical data always be safe. On the other hand, the transmission of medical data, especially medical imaging

data, requires a significant amount of bandwidth. Efficient transmission has turned out to be another imperative concern, where promising compression algorithms with privacy-preserving capability are highly needed. However, privacy-preserving compression algorithms generally have to deal with the encrypted biomedical data directly, which creates big challenges for conventional compression techniques. This happens because the principle of compression is to remove redundancy through exploiting structure of data, whereas encrypted data typically appear to be random, as they mask the structure of the original data (otherwise traces of original data could be leaked out from any perceived structure of the encrypted data [53]). The "best" practice has been to reverse the order of these steps by compressing the data before encryption [14]. However, the conventional "best" practice obviously may not be suitable in medical related applications with privacy concerns, since data hosts may not always be the data owners, where any manipulation (e.g. compression) on the unencrypted data initiated by the data host could lead to insecure exposure of private data.

To always keep data under encryption such that the privacy of the data and security of the system are well preserved, encryption followed by compression is essential but challenging. Although the encrypted data appears to be "completely"random, the fact that the encrypted data is not random conditioned on the cryptographic key offers some hope for tackling the impediment. More precisely, redundancy within the original data might be recovered by performing joint decompression and decryption at the decoder side given the cryptographic key making direct compression on encrypted data feasible. Surprisingly, the aforementioned setup is nothing but a distributed source coding (DSC) problem with side information only available

at the decoder. More important, the theoretical result of DSC [3] based compression guarantees that there is no performance loss even when performing compression without side information (e.g., the cryptographic key) at the encoder, which is identical to the compression of encrypted data. This essentially means that encrypted data can be compressed very efficiently just like raw data, as long as the key is given to the decoder without sacrificing any security or privacy at the encoder side. Therefore, privacy-preserving compression opens up many possibilities, e.g., efficient medical data management, where the data hosts, who primarily focus on efficient data transmission/dissemination, and data owners, who are mainly concerned about data security and privacy, can be totally separated.

DSC offers a promising solution for this tricky problem (i.e., on compression of encrypted data), on which many secure compression algorithms have been investigated in [14, 15, 54–56]. For example, Johnson *et. al.* [54] proved the feasibility of compressing encrypted data in theory. The encrypted data are as compressible as the original data when the cryptographic key is available at joint decompression and decryption [54]. Moreover, Schonberg *et. al.* described a low-density parity-check (LDPC) codes based secure compression scheme for binary images using 1D [55] Markov correlation structure at the joint decoder. Since the 1D Markov correlation model has poor match performance for exploring 2D correlation within a binary image, a 2D Markov source model [15] is proposed to enhance the secure compression performance. However, all above works [15, 55] limit themselves within the scope of binary images. Recently, the work proposed by Schonberg *et. al.* in [14] described a way to losslessly compress encrypted gray-scale video sequences frame by frame. In [14], each pixel in a gray-scale frame is bitwisely decomposed

into 8 binary bits with the outputs of 8 binary sub-frames for any given gray-scale frame. Moreover, by assuming that all above 8 binary sub-frames are equally significant in their gray-scale pixel representation, each binary sub-frame can be processed separately according to the aforementioned 2D Markov based secure compression algorithm [15]. However, the correlation exploited within each independent binary sub-frame [15] cannot guarantee the ultimate compression efficiency, as the correlations among sub-frames are lost. Consequently, directly applying these models [15, 55] to practical gray-scale images/videos compression with privacy concerns encounters many difficulties in coding efficiency.

In practical image/video coding, two most common techniques such as DCT (discrete cosine transform) and localized predictor, are infeasible for dealing with encrypted images, since the exploitable structure or local information of original images are totally masked in the encrypted images. Fortunately, the temporal or spatial correlation among adjacent frames in video/image sequences offers a possible opportunity for frame by frame compression after encryption. In this Chapter, primarily inspiring by spatial correlation among adjacent CT image slices, we proposed a practical SUPERMICRO framework to provide robust privacy-preserving compression on gray-scale medical images (i.e., CT image sequences).

5.2 System Architecture

The block diagram of the proposed SUPERMICRO framework with compression of encrypted data is presented in Fig. 5.1. Let $\tilde{X}$ and $\tilde{Y}$ denote the i-th and $(i-1)$-th original CT image slices, which are first encrypted into X and Y, respectively, through some standard encryption techniques (e.g. stream

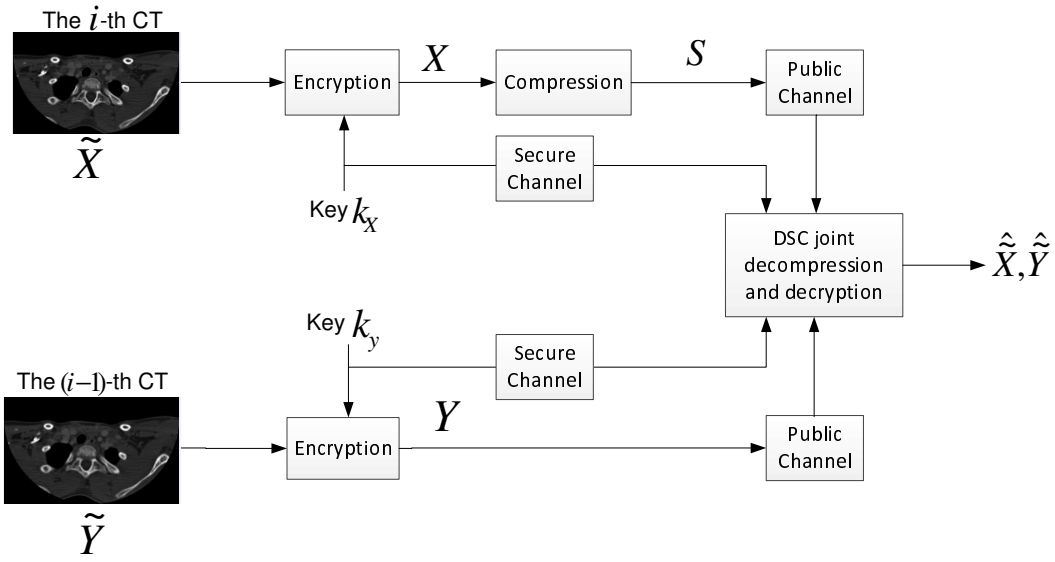
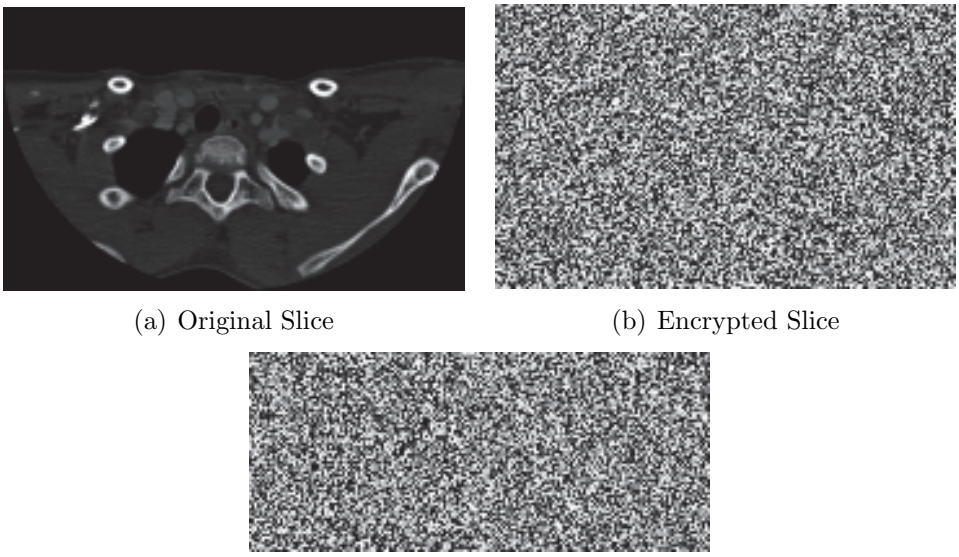


Figure 5.1: The workflow of the proposed framework.

cipher). Furthermore, cryptographic keys used to encrypt these slices are sent to joint decoder through secure channels. Then the encrypted slice X is compressed into S (i.e., so called syndromes) using a standard DSC encoder without accessing the cryptographic key. At the decoder, the encrypted slice Y is treated as side information for helping the joint decompression and decryption given the received syndromes S and cryptographic keys of both X and Y, where $\widehat{\tilde{X}}$ and $\widehat{\tilde{Y}}$ denote the estimated slices by the decoder. The central process of the proposed SUPERMICRO system consists of two key components as (1) compression of encrypted data (see Section 5.2.1); (2) joint decompression and decryption (see Section 5.2.2). The following subsections present the details of each process.

5.2.1 Compression of encrypted data

In this module, each CT image slice is first encrypted with stream cipher and thus perfect secrecy is guaranteed [57]. This is illustrated in Figs. 5.2(a)



(a) Original Slice (b) Encrypted Slice

(c) Compressed Slice

Figure 5.2: Illustration of DSC-based compression of an encrypted CT image slice. The original slice (Fig. 5.2(a)) is encrypted into the slice shown in Fig. 5.2(b) using stream cypher. The encrypted slice is then compressed using DSC method into that shown in Fig. 5.2(c) with much smaller size.

and 5.2(b). For example, the pixels of the original slice in Fig. 5.2(a) have values in the range of 0 to 255. Therefore, each pixel can be represented by 8 bits such that each image consists of 8 bit planes (from the most to the least significance), where each bit plane includes all the bits with equal significance in the bitwise pixel representation. Then, 8 randomly sampled keys with uniform distribution, where each of them corresponds to one bit plane and has the same size as the original slice, will be generated. Encrypting the original slice in Fig. 5.2(a) can be achieved by applying a bitwise exclusive-OR (XOR) between each bit plane of original slice and the generated key, where the encrypted slice is shown in Fig. 5.2(b). Since the cryptographic key is i.i.d across all pixels and bit planes, the correlation and redundancy among pixels after encryption will be totally destroyed and thus is impossible to be compressed in a traditional sense. Fortunately, given that the key is

available at the decoder, we can compress the encrypted slice of CT image through DSC principle.

The implementation of DSC is largely based on the idea of random code. Bits of different bit planes and pixels are randomly intermixed and the resulting bitwise sum will be sent to the decoder as a "syndrome". The number of these one-bit syndromes M will be smaller than the total number of encrypted bits and thus results in a compression, where each encrypted slice has N 8-bit pixels for total $8N$ encrypted bits and each pixel x_i represented by $x_i^1, x_i^2, \cdots, x_i^8$ in its binary format. However, the decoding complexity in DSC highly depends on the code length, which means that direct compression and decompression on encrypted bits with $8N$ bits may result in significant computational burden for the decoder with limited computation power. One workaround to tackle this difficulty is to compress only the first q ($q \in [2, 8]$) most significant bit planes instead of all 8 bits, which offers a great trade-off between compression performance and decompression complexity. The impact of different selections of q value on the compression performance will be studied in our experimental results.

Now, let us denote $\mathbf{B} = \{x_1^1, x_1^2, \cdots, x_1^q, \cdots, x_N^1, \cdots, x_N^q\}$ by the binary vector to be compressed. The remaining $(8-q)N$ encrypted bits will be sent to the decoder directly without any compression. For LDPC codes based Slepian-Wolf (SW) coding (lossless DSC scheme [3]), the compressed syndromes are generated through $\mathbf{S} = \mathbf{H} \times \mathbf{B}^T$, where $\mathbf{H}$ is a parity check matrix of LDPC codes with size $M \times qN$ and $M < qN$. Thus, the LDPC based SW codes results in a $R = ((8-q)N + M) : 8N$ code rate.

5.2.2 Joint decompression and decryption design

To make joint decompression and decryption for CT image sequences possible, a key factor is to be able to explore the spatial correlation between adjacent CT image slices at the decoder, which typically can be achieved by applying Bayesian inference on graphical models. The graphical model of our proposed SUPERMICRO scheme for joint decompression and decryption of encrypted CT slices is shown in Fig. 5.3, where variable nodes (usually depicted by circles) denote variables such as encrypted variables, encryption keys, syndromes (i.e., compressed bits) and unencrypted variables, and factor nodes (depicted by small squares) represent the relationship among the connected variable nodes.

Our proposed SUPERMICRO scheme is based on SW codes and only the first q significant bit planes are compressed as an example ($q = 3$) shown in Fig. 5.3. In the right hand side of Fig. 5.3, x_i^l, the realization of variable node X_i^l, with $i = 1, \cdots, N$, $l = 1, \cdots, q$ represent the encrypted bits to be recovered, while x_i^l with $i = 1, \cdots, N$, $l = q+1, \cdots, 8$ denote the received encrypted bits without any compression. In addition, s_j, the realization of variable node S_j, $j = 1, \cdots, M$, represent the received syndromes. The factor nodes f_j, $j = 1, \cdots, M$ connecting these variable nodes X_i^l and S_j take into account the parity check constraints. Since the received encrypted bits without any compression are directly available at the decoder side, we no longer need the factor nodes f_j to guarantee the parity check for the corresponding variable nodes. Actually, the right hand side in Fig. 5.3 without these uncompressed bits is identical to the factor graph of the standard LDPC codes. For the factor node f_j, $j = 1, \cdots, M$, the corresponding factor function can be expressed as

$$f_j(\mathbf{x}_{f_j}, s_j) = \begin{cases} 1, & \text{if } s_j \oplus \bigoplus \mathbf{x}_{f_j} = 0, \\ 0, & \text{otherwise.} \end{cases} \tag{5.1}$$

where $\mathbf{x}_{f_j}$ denotes the set of neighbors of factor node f_j, and $\bigoplus \mathbf{x}_{f_j}$ denotes the binary sum of all elements of the set $\mathbf{x}_{f_j}$.

Since pixels in the encrypted slices have approximately a uniform distribution, it is infeasible to directly obtain the correlation between original slices based on the encrypted versions, and the correlation information is essential for the inference algorithm to reconstruct the original slices. To find the correlation, one could resort to the cryptographic keys available at the decoder, where the factor nodes h_i^Y and h_{il}^X, $i = 1, \cdots, N$ is used to incorporate the constraints of cryptographic keys. The factor functions for factor nodes h_i^Y can be defined as

$$h_i^Y(y_i, \tilde{y}_i, \mathbf{k}_i^Y) = \begin{cases} 1, & \text{if } y_i^l \oplus \tilde{y}_i^l \oplus k_{il}^Y = 0, \\ & \text{for all } l = 1, \cdots, 8 \\ 0, & \text{otherwise} \end{cases} \tag{5.2}$$

where y_i^l and $\tilde{y}_i^l$ represent the l-th significant bit of pixel y_i and $\tilde{y}_i$ with $l = 1, \cdots, 8$, and key $\mathbf{k}_i^Y = \{k_{i1}^Y, \cdots, k_{i8}^Y\}$, respectively.

Similarly, the factor functions for factor nodes h_{il}^X can be expressed as

$$h_{il}^X(x_i^l, \tilde{x}_i^l, k_{il}^X) = 1 \oplus x_i^l \oplus \tilde{x}_i^l \oplus k_{il}^X \tag{5.3}$$

The definitions of the above factor functions guarantee that both each side information pixel $\tilde{y}_i$ and any candidate source pixel $\tilde{x}_i$ will satisfy the

encryption constraints. Then, we introduce additional factor nodes g_i to capture the correlation between $\tilde{y}_i$ and $\tilde{x}_i$. As identified by many previous studies [16–18], the statistical correlation between adjacent frames in a video sequence can be effectively modeled by a Laplace distribution. Here, for neighboring CT image slices, we observed that the inter-slice correlation also satisfies Laplace distribution as depicted in Fig. 5.4. Therefore, the factor function of factor node g_i is defined as

$$g_i(\tilde{x}_i, \tilde{y}_i, \alpha) = \int_{P(\tilde{x}_i)}^{P(\tilde{x}_i+1)} \frac{\alpha}{2} e^{-\alpha|x-\tilde{y}_i|} dx, \tag{5.4}$$

where α is the scale parameter of the Laplace distribution, $P(\tilde{x}_i)$ denotes the lower partition boundary at index $\tilde{x}_i$, e.g. whether a coefficient $\tilde{x}_i$ satisfies $P(\tilde{x}_i) \leq \tilde{x}_i < P(\tilde{x}_i + 1)$.

Based on the factor graph with factor function defined above, an estimate of the original slice $\widehat{\tilde{X}}$ can be decoded by performing the belief propagation (BP) [47] algorithm on the factor graph, which offers an efficient way to calculate the marginal distribution of unknown variables using Bayesian inference.

5.3 Practical implementation issues

In this section, we discuss some practical implementation issues of our proposed SUPERMICRO system. The first practical implementation issue is the side information availability at the decoder side. In the proposed framework, the previous slice (i.e., $(i-1)$-th slice) is considered as the side information slice of the i-th source slice, as shown in the Fig. 5.1. Therefore, the most efficient way in terms of code rate saving is to send only the first encrypted

slice uncompressed to the decoder as side information. Next, the second compressed slice can be jointly decoded with the help from side information slice at decoder and then it can serve as the side information for the third slice. We can continue the above process for the rest of the compressed slices. The aforementioned setup yields high compression efficiency, as only one uncompressed slice (a.k.a, key slice) is sent to the decoder. However, the cost for such a single key slice setup is an increased decoding latency, since to decode an interested slice, one should go through all its previous compressed slices. One workaround for this drawback is to increase the number of key slices and evenly distribute them among all compressed slices with a slight loss of compression performance.

Then second practical implementation issue is an accurate estimate of the correlation parameter α for different source and side information slice pair, due to the dynamical changes of pixels within each slice. According to the assumption that the adjacent image slices are highly correlated, the correlation parameter α of the source and side information slice pair can be learnt from the residuals between the previous two decoded neighboring slices, as in [16]. It also implies that at least two uncompressed slice should be available at the decoder if we want to fully take advantage of correlation estimation. Hence, in this Chapter, we defined the group of slices (GOS) size as the number that represents how frequently every two uncompressed slices will be distributed among the compressed slices. The impact of GOS size on compression performance will be studied in the results section.

5.4 Experimental Results

In this section, we demonstrate the performance of the proposed SUPER-MICRO in privacy-preserving capability and compression efficiency through experimental results.

5.4.1 Experiment Setup

In our experiments, we used two different CT image sets with 100 slices in each set, for a total of 200 slices. Figs. 5.5 (a) - (b) and (c) - (d) illustrate the first and the last slices in the CT image set 1 and set 2, respectively. As shown in Figs. 5.5 (a) to (d), great changes have taken place from the first to the last slice in each CT image set, therefore, the image sets offer diversely experimental conditions for verifying the robustness of the proposed SUPERMICRO framework. The size of each slice is 484×396 for all CT image sets. For the sake of coding efficiency, each slice is partitioned into sub slices with size of 44×36, which results in a total of $11 \times 11 = 121$ sub slices for each slice, as depicted in Fig. 5.6 (a). Furthermore, the slice partitioning technique can be used to easily fit a CT image slice with any size in practice. Then, in each sub slice, the number of pixels is equal to $N = 44 \times 36 = 1584$. Hence, the length of possible encode bits qN varies from 3168 to 12672 for $q \in [2, 8]$ in our experiments, where LDPCA codes with corresponding lengths were used for rate adaptive decoding. In the rest of this section, the GOS size is equal to 100 for all experiments, except where explicitly indicated.

5.4.2 Security and privacy protection

The secrecy of a privacy-preserving compression scheme is essential, where the strong security [14] is achieved by using the stream cipher technique in our experiment. In the proposed framework, each bit plane of the original CT image slice is encrypted separately by using its own encryption key with the same size. For example, Fig. 5.6 (b) shows the encryption keys of all 8 bit planes in decimal representation. By applying each encryption key on the corresponding bit plane, the encrypted slice can be found in the Fig. 5.6 (c). Here, the grids in Figs. 5.6 (a) (b) and (c) partition the whole slice into sub slices. In Fig. 5.6 (c), we can see the correlation of pixels among the original slice has been totally destroyed. Therefore, without knowing the encryption key, the stream cipher based encryption technique offers strong security and privacy protection. Besides the security, the compression performance is another important criterion for privacy-preserving compression scheme, which will be studied in the next subsection.

5.4.3 Compression performance

First, we investigate how the compression efficiency varies partition-by-partition. Fig. 5.6 (d) depicts the code rate of each partition after applying the proposed SUPERMICRO framework, where a lower code rate refers to a higher compression efficiency and the average code rate of the given slice is $R = 0.38$. In Fig. 5.6 (d), we can see that a partition (i.e., a sub slice) with homogeneous background (e.g., the boundary regions with dark background) is usually associated with the lowest code rate, while a partition with irregularly illuminating changes (e.g. the central regions) normally results in a higher code rate. This is because the achievable code rate of a partition highly

depends on how well the source and the side information partitions are correlated, where a source partition with a homogeneous background usually has a higher correlation with its side information partition.

As illustrated in Section 5.2.1, the decoding complexity is directly proportional to the number of encoded bits. Second, we are interested in the impact of number of encoded bits on the compression performance in terms of code rate, which offers a trade-off between decoding complexity and compression performance. Fig. 5.7 shows the average code rates with error bars for the two CT image sets based on all encoded slices, where the code rates for both CT image sets decrease as the number of encoded bits increases. Moreover, the lowest achievable code rates of CT image set 1 and set 2 are 0.41 and 0.32, respectively, when the number of encoded bits is $q = 8$.

Next, we study the varieties of each individual slice's code rate with three different setups, i.e., the proposed SUPERMICRO on encrypted slices, JPEG 2000 lossless compression on both original slices and encrypted slices in Figs. 5.8 and 5.9, where the first two slices are sent to the decoder uncompressed in our SUPERMICRO framework. Here, the state-of-the-art JPEG 2000 lossless compression on original slices serves as the baseline performance to show how well a slice could be compressed without any encryption. Ideally, the closer the performance of the proposed SUPERMICRO on encrypted slices can approach the baseline performance, the higher the compression performance of our proposed framework gets. Moreover, the setup of JPEG 2000 lossless compression on encrypted slices depicts an intuitive sense on how difficult the encrypted slice could be compressed through traditional compression techniques. In both Figs.5.8 and 5.9, we can see that the performance of the proposed SUPERMICRO framework on encrypted slices is very

close to the baseline performance in terms of code rate, where the minimum differences are 0.029 and 0.074 for CT image set 1 and set 2, respectively. In addition, the maximum differences are bounded by 0.11 and 0.095 for set 1 and set 2, respectively. Interestingly, the average code rates of JPEG 2000 lossless compression on encrypted slices for both set 1 and set 2 are 1.0889 and 1.0888, respectively, which means that JPEG 2000 lossless "compression" on encrypted slices will result in data size increase instead of data size reduction after compression. Such observations verify our previous statement that direct compression on encrypted data is usually infeasible. It is even worse than not compressing the image, as the traditional compression technique (e.g., JPEG 2000) needs to introduce additional overhead for compression.

Finally, we investigate how the average code rate changes with different GOS sizes. In Section 5.3, we have shown that a smaller GOS size could bring a shorter decoding latency, as the proposed SUPERMICRO framework requires that the decoding must start from the immediate neighboring slice of the key slice. In other words, a smaller GOS size offers more key slices, which are evenly distributed among the encoded slices, therefore, an interested encoded slice can be decoded through discovery of the shortest pathway from its nearest neighboring key slice. However, the cost of using a smaller GOS size is that the overall code rate is also increased, as key slices can not be compressed. As expected, Fig. 5.10 shows that the overall code rates of both image sets indeed increase as the GOS size decreases.

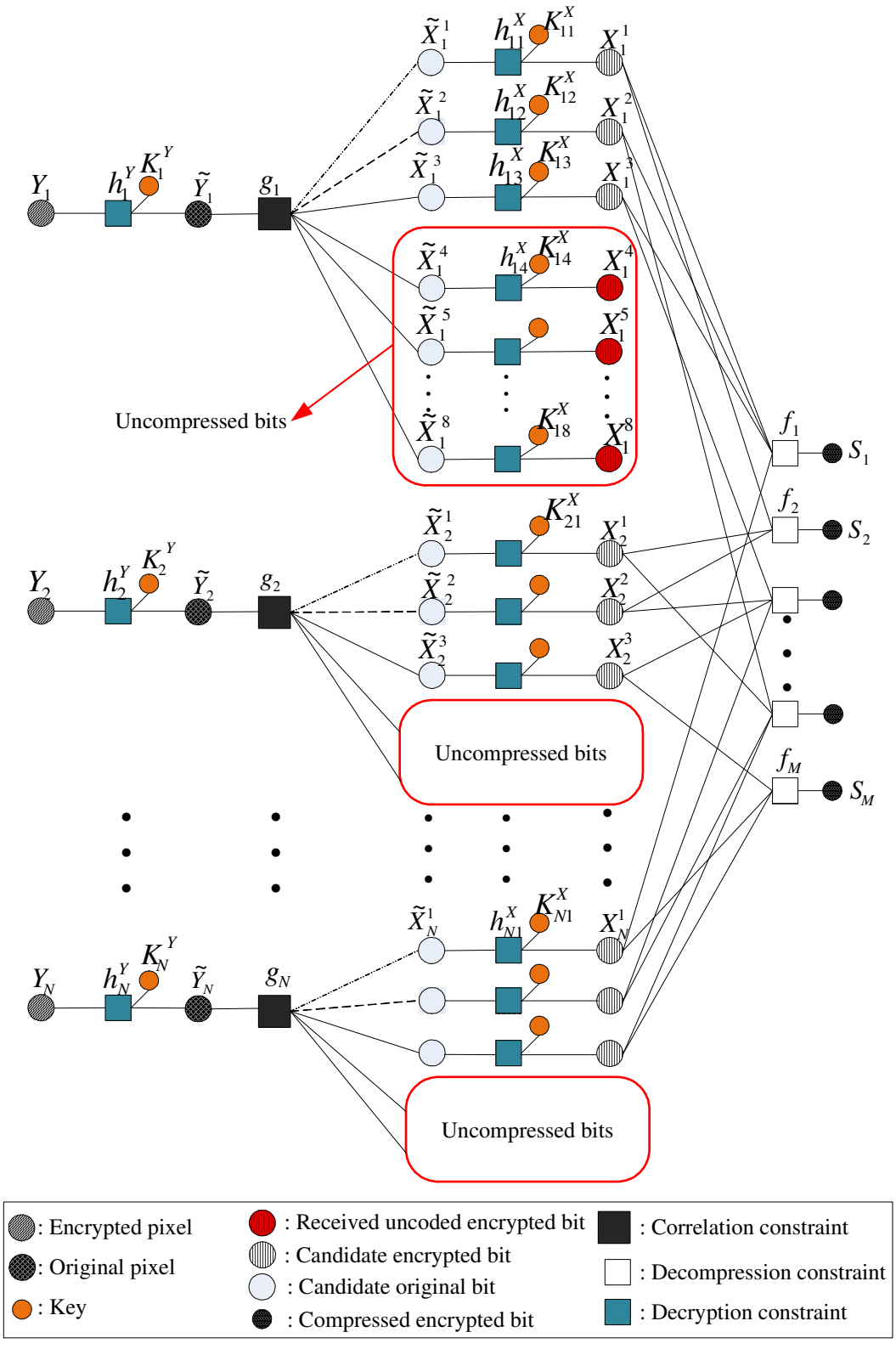


Figure 5.3: Factor graph for decompression of compressed encrypted data.

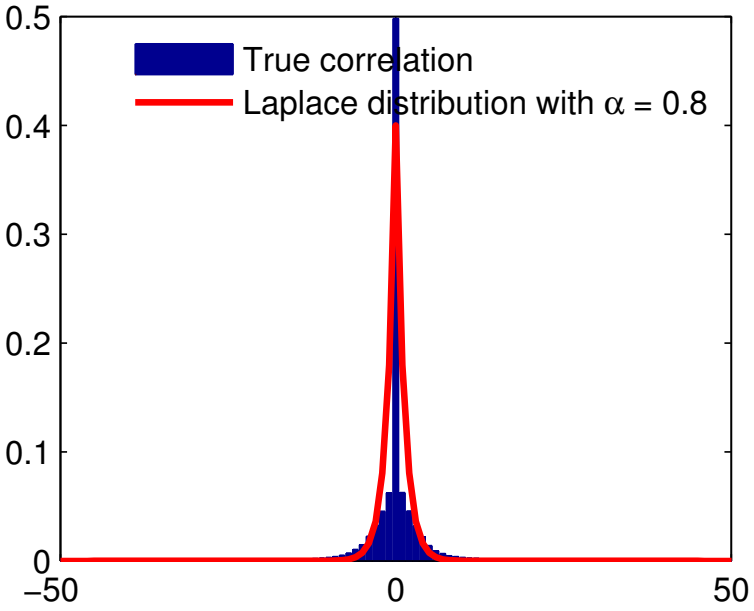


Figure 5.4: Residual histogram for slice sequences of a CT image, where the Laplace distribution with $\alpha = 0.8$ is shown as reference.

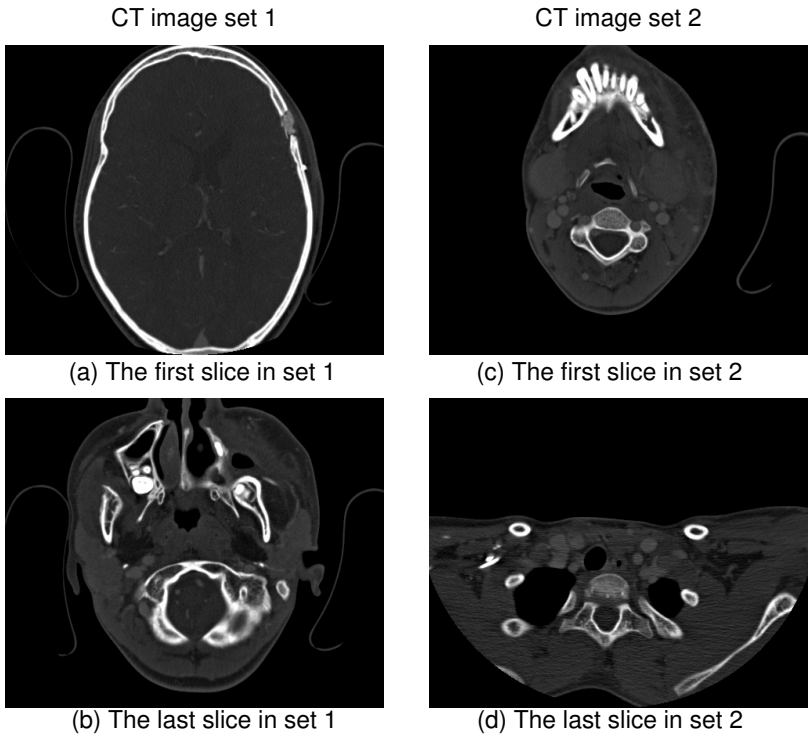


Figure 5.5: The first and the last slices in CT image set 1 (i.e., (a) and (b) respectively) and set 2 (i.e., (c) and (d) respectively)

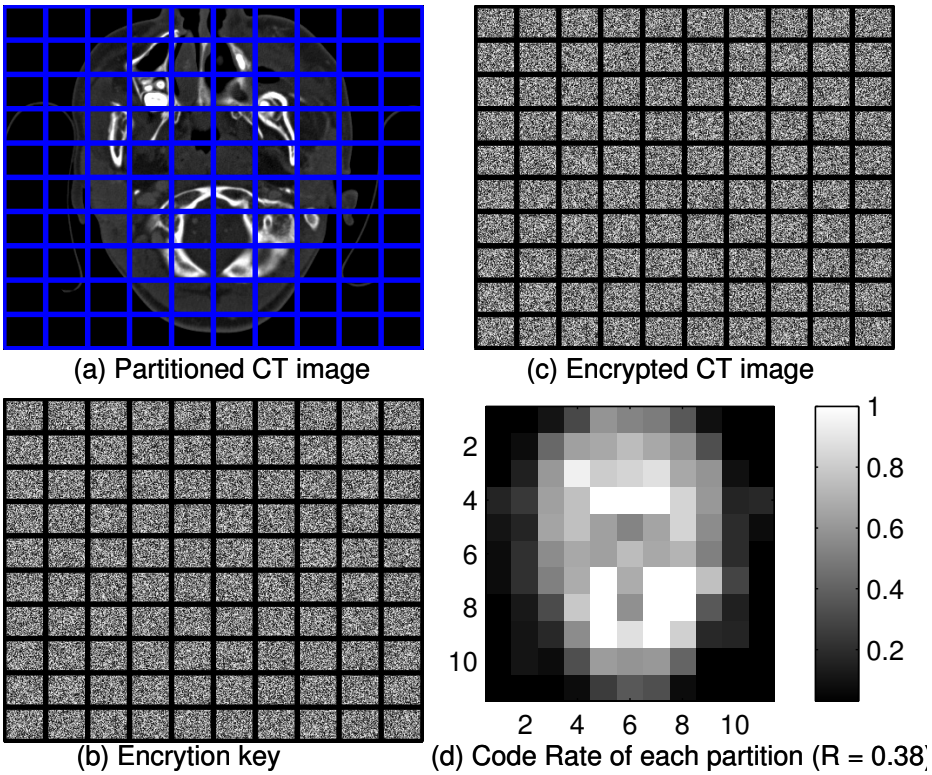


Figure 5.6: Examples of (a) partitioned CT image slice; (b) encryption key; (c) encrypted CT image; (d) code rates of each partition by using the proposed SUPERMICRO framework, where the grids in (a), (b), (c) partition the whole slices into sub slices and the average code rate in (d) is $R = 0.38$ for the given slice.

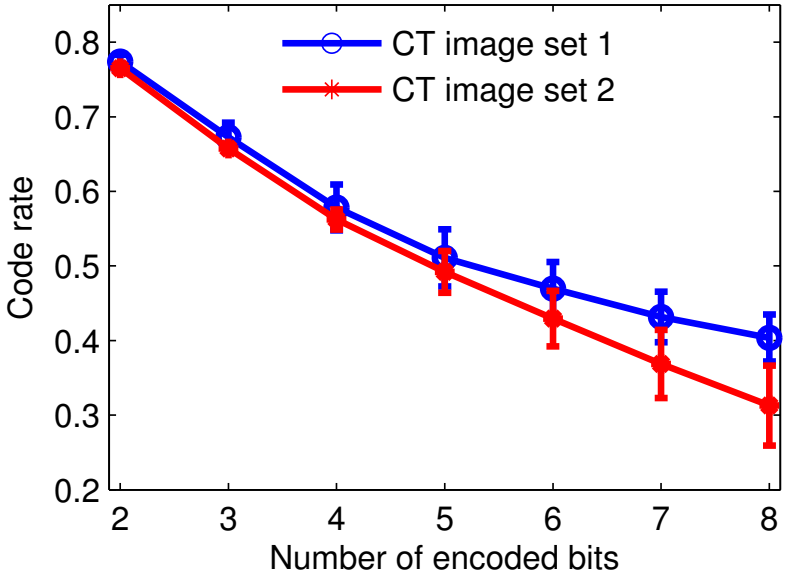


Figure 5.7: Code rate vs. different number of encoded bits for both CT image set 1 (i.e., blue dash-circle line) and set 2 (i.e., red dash-dot line) using the proposed SUPERMICRO system.

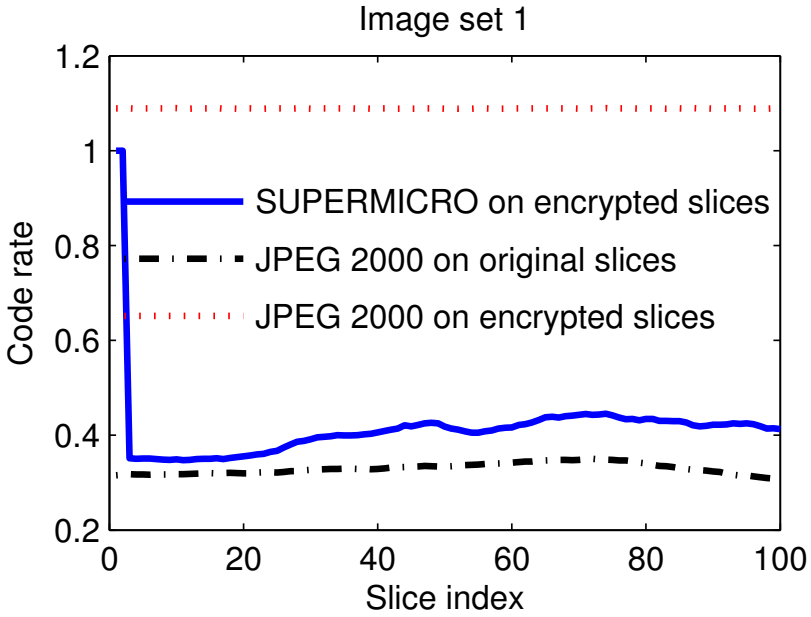


Figure 5.8: Code rate vs. different CT image slices for image set 1, which compared three different setups, i.e., the proposed SUPERMICRO on encrypted slices, JPEG 2000 lossless compression on both original slices and encrypted slices.

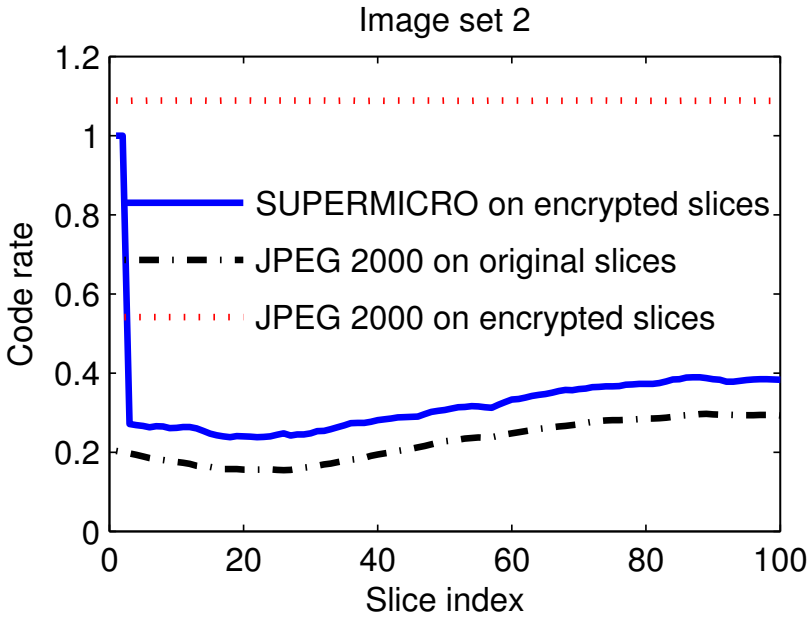


Figure 5.9: Code rate vs. different CT image slices for image set 2, which compared three different setups, i.e., the proposed SUPERMICRO on encrypted slices, JPEG 2000 lossless compression on both original slices and encrypted slices.

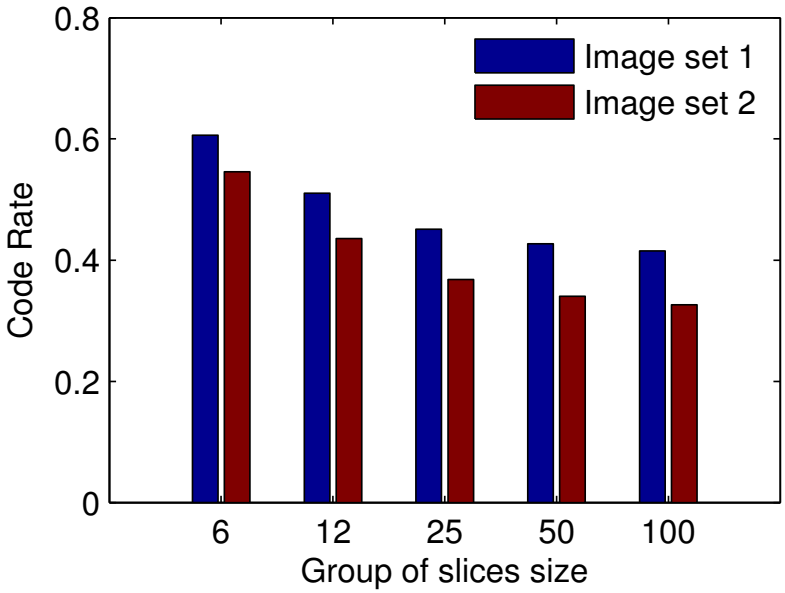


Figure 5.10: Code rate vs. different GOS sizes of 6, 12, 25, 50 and 100 for image set 1 and set 2.

CHAPTER 6

CONCLUSIONS

Distributed video coding (DVC) is rapidly increasing in popularity by the way of shifting the complexity from encoder to decoder, whereas no compression performance degrades, at least in theory. Compared with conventional video technique, the inter-frame correlation in DVC is explored at decoder based on the received syndromes of Wyner-Ziv (WZ) frame and side information (SI) frame generated from key frames available only at decoder. Generally, the existing correlation estimation methods in DVC are twofolds: online-estimation where estimation starts before decoding and on-the-fly (OTF) estimation where estimation can be refined iteratively during decoding. The online correlation estimation scheme in [16] provided different estimation granularities, such as frame, block and pixel levels. Through experimental results, the authors shows that a fine grained estimation would results better DVC performance. However, the aforementioned estimation methods fails to take any advantage of the received WZ frame syndromes, which could be used to further refine the estimated correlation iteratively during decoding. Therefore, I proposed an OTF correlation estimation algorithm in this dissertation by incorporating sampling technique based on factor graph. The experimental results depict that the proposed OTF outperforms the online correlation estimation algorithms. However, sampling based method usually results in a significant computational burden for the decoder. I proposed an improved model based EP, a deterministic approximate inference methods, which significantly reduced the complexity of the correlation estimator.

Finally, we present a compression of encrypted medical image sequences

framework based on the distributed source coding (DSC) principle, called Secure Privacy-presERving Medical Image CompRessiOn (SUPERMICRO), which makes the compression of the encrypted data possible without compromising security and compression efficiency. SUPERMICRO also guarantees the data transmission in a privacy-preserving manner. Compared to the previous work, performing decompression on each bit-plane (i.e., sub-frame) separately with a 2D Markov based symmetric correlation model, the proposed SUPERMICRO framework employs a joint bit-plane decoder based on a factor graph that can handle the inter-pixel correlation between adjacent CT image slices. It uses a realistic Laplacian correlation model. Moreover, our proposed SUPERMICRO framework incorporates LDPCA codes for rate adaptive decoding. The experimental results based on two CT image sets with 200 slices show that the proposed system provides high-level security and privacy protection, as well as significant compression performance, even when comparing with that of the state-of-the-art JPEG 2000 lossless compression on unencrypted slices.

BIBLIOGRAPHY

[1] A. Aaron, R. Zhang, and B. Girod, "Wyner-Ziv coding of motion video," in *the Thirty-Sixth Asilomar Conf. Signals, Syst. Computers*, vol. 1, pp. 240–244, 2002.

[2] R. Puri and K. Ramchandran, "PRISM: A new robust video coding architecture based on distributed compression principles," in *Proc. Annual Allerton Conf. Commun., Control and Computing*, vol. 40, pp. 586–595, 2002.

[3] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inform. Theory*, vol. 19, pp. 471–480, Jul. 1973.

[4] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inform. Theory*, vol. 22, pp. 1–10, Jan. 1976.

[5] B. Girod, A. M. Aaron, S. Rane, and D. A. R.-M. D. Rebollo-Monedero, "Distributed video coding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 71–83, 2005.

[6] C. Guillemot, F. Pereira, L. Torres, T. Ebrahimi, R. Leonardi, and J. Ostermann, "Distributed monoview and multiview video coding: basics, problems and recent advances," *IEEE Signal Processing Magazine*, vol. 24, no. 5, pp. 67–76, 2007.

[7] L. Stankovic, V. Stankovic, and S. Cheng, "Distributed compression: Overview of current and emerging multimedia applications," in *Proc. ICIP-2011 IEEE Int. Conf. Image Processing*, Sept. 2011.

[8] A. Wyner, "Recent results in the Shannon theory," *IEEE Trans. Inform. Theory*, vol. 20, pp. 2–10, Jan. 1974.

[9] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (discus): design and construction," in *Proc. DCC*, pp. 158–167, 1999.

[10] D. Schonberg, K. Ramchandran, and S. S. Pradhan, "Distributed code constructions for the entire Slepian-Wolf rate region for arbitrarily correlated sources," in *Data Compression Conference, 2004. Proceedings. DCC 2004*, pp. 292–301, 2004.

[11] B. Rimoldi and R. Urbanke, "Asynchronous Slepian-Wolf coding via source-splitting," in *ISIT'97*, (Ulm, Germany), p. 271, 1997.

[12] Z. Xiong, A. Liveris, and S. Cheng, "Distributed source coding for sensor networks," *IEEE Signal Process. Magazine*, vol. 21, pp. 80–94, Sep. 2004.

[13] Y. Yang, V. Stankovic, Z. Xiong, and W. Zhao, "On multiterminal source code design," *IEEE Trans. Inform. Theory*, vol. 54, no. 5, pp. 2278–2302, 2008.

[14] D. Schonberg, C. Yeo, S. Draper, and K. Ramchandran, "On compression of encrypted video," in *Data Compression Conference, 2007. DCC'07*, pp. 173–182, IEEE, 2007.

[15] D. Schonberg, S. Draper, and K. Ramchandran, "On compression of encrypted images," in *Image Processing, 2006 IEEE International Conference on*, pp. 269–272, IEEE, 2006.

[16] C. Brites and F. Pereira, "Correlation noise modeling for efficient pixel and transform domain wyner–ziv video coding," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 9, pp. 1177–1190, 2008.

[17] S. Wang, L. Cui, L. Stankovic, V. Stankovic, and S. Cheng, "Adaptive correlation estimation with particle filtering for distributed video coding," *Circuits and Systems for Video Technology, IEEE Transactions on*, 2011.

[18] L. Stankovic, V. Stankovic, S. Wang, and S. Cheng, "Distributed Video Coding with Particle Filtering for Correlation Tracking," *Proc. EUSIPCO, Aalborg, Denmark*, 2010.

[19] S. Wang, L. Cui, S. Cheng, Y. Zhai, M. Yeary, and Q. Wu, "Noise adaptive ldpc decoding using particle filtering," *Communications, IEEE Transactions on*, vol. 25, pp. 1–4, 4 2011.

[20] L. Cui, S. Wang, S. Cheng, and M. Yeary, "Adaptive binary Slepian-Wolf decoding using particle based belief propagation," *IEEE Transactions on Communications*, no. 99, pp. 1–6, 2011.

[21] S. Wang, L. Cui, L. Stankovic, V. Stankovic, and S. Cheng, "Adaptive correlation estimation with particle filtering for distributed video coding," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 5, pp. 649–658, 2012.

[22] S. Wang, L. Cui, S. Cheng, L. Stankovic, and V. Stankovic, "Onboard Low-Complexity Compression of Solar Stereo Images," *Image Processing, IEEE Transactions on*, vol. 21, no. 6, pp. 3114–3118, 2012.

[23] S. Cheng, S. Wang, and L. Cui, "Adaptive Slepian-Wolf decoding using particle filtering based belief propagation," in *Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on*, pp. 607–612, IEEE, 2009.

[24] S. Wang, L. Cui, and S. Cheng, "Adaptive Wyner-Ziv decoding using particle-based belief propagation," in *2010 IEEE Global Telecommunications Conference GLOBECOM*, IEEE, 2010.

[25] S. Cheng, S. Wang, and L. Cui, "Adaptive nonasymmetric Slepian-Wolf decoding using particle filtering based belief propagation," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pp. 3354–3357, IEEE, 2010.

[26] S. Wang, L. Cui, S. Cheng, L. Stankovic, and V. Stankovic, "Onboard Low-complexity Compression of Solar Images," in *IEEE International Conference on Image Processing*, 2011.

[27] L. Cui, S. Wang, S. Cheng, and Q. Wu, "Noise adaptive ldpc decoding using particle filter," in *Information Sciences and Systems, 2009. CISS 2009. 43rd Annual Conference on*, pp. 37–42, 2009.

[28] L. Cui, S. Wang, and S. Cheng, "Adaptive Slepian-Wolf decoding based on expectation propagation," *Communications Letters, IEEE*, vol. 16, no. 2, pp. 252–255, 2012.

[29] S. Wang, L. Cui, and S. Cheng, "Noise Adaptive LDPC Decoding Using Expectation Propagation," in *IEEE Global Telecommunications Conference*, 2011.

[30] L. Cui, S. Wang, X. Jiang, and S. Cheng, "Adaptive distributed video coding with correlation estimation using expectation propagation," pp. 84990M–84990M–13, 2012.

[31] S. Wang, X. Jiang, L. Ohno-Machado, L. Cui, and S. Cheng, "Secure privacy-preserving medical image compression (supermicro)," in *Healthcare Informatics, Imaging and Systems Biology (HISB), 2012 IEEE Second International Conference on*, pp. 130–130, 2012.

[32] C. Brites, J. Ascenso, and F. Pereira, "Improving transform domain wyner-ziv video coding performance," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 2, pp. II–II, Ieee, 2006.

[33] J. Ascenso, C. Brites, and F. Pereira, "Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding," in *5th EURASIP Conf. Speech, Image Proc., Multimedia Commun., Services*, 2005.

[34] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 71–83, 2005.

[35] A. Aaron, S. D. Rane, E. Setton, and B. Girod, "Transform-domain wyner-ziv codec for video," *Proceedings of SPIE*, vol. 5308, pp. 520–528, 2004.

[36] A. Aaron, S. Rane, and B. Girod, "Wyner-Ziv video coding with hash-based motion compensation at the receiver," in *IEEE ICIP'04*, vol. 5, pp. 3097–3100, 2005.

[37] P. Meyer, R. Westerlaken, R. Gunnewiek, and R. Lagendijk, "Distributed source coding of video with non-stationary side-information," in *Proc. SPIE*, vol. 5960, pp. 857–866, 2005.

[38] M. Dalai, R. Leonardi, and F. Pereira, "Improving turbo codec integration in pixel-domain distributed video coding," in *Proc. ICASSP-2006 IEEE Int. Conf. Acoustics, Speech and Sig. Proc.*, vol. 2, pp. II–II.

[39] X. Fan, O. Au, and N. Cheung, "Adaptive correlation estimation for general Wyner-Ziv video coding," in *Proc. ICIP-2009 IEEE Int. Conf. Image Proc.*, pp. 1409–1412, 2009.

[40] X. Huang and S. Forchhammer, "Improved virtual channel noise model for transform domain Wyner-Ziv video coding," in *Proc. ICASSP-2009 IEEE Int. Conf. Acoustics, Speech and Sig. Proc.*, pp. 921–924, 2009.

[41] A. Doucet and N. De Freitas, *Sequential Monte Carlo Methods in Practice*. Springer Verlag, 2001.

[42] M. Bolic, P. M. Djuric, and S. Hong, "Resampling algorithms for particle filters: A computational complexity perspective," *EURASIP Journal on Applied Signal Processing*, vol. 15, no. 5, pp. 2267–2277, 2004.

[43] S. Chib and E. Greenberg, "Understanding the Metropolis-Hastings algorithm," *American Statistician*, vol. 49, no. 4, pp. 327–335, 1995.

[44] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *Signal Processing, IEEE Transactions on*, vol. 50, no. 2, pp. 174–188, 2002.

[45] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, and M. Ouaret, "The DISCOVER codec: architecture, techniques and evaluation," in *Picture Coding Symposium*, Citeseer, 2007.

[46] G. Bjontegard, "Calculation of average psnr differences between rd-curves," *ITU-T VCEG-M33*, 2001.

[47] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[48] D. J. C. MacKay and R. M. Neal, "Near shannon limit performance of low density parity check codes," *Electronics Letters*, vol. 32, no. 18, 1996.

[49] T. P. Minka, "Expectation propagation for approximate bayesian inference," in *Uncertainty in Artificial Intelligence*, vol. 17, pp. 362–369, 2001.

[50] I. T. U. Recommendation, "Advanced Video Coding for Generic Audiovisual Services," *ITU-T Rec. H.264*, 2007.

[51] G. Chambers, M. Rockey, R. Marshall, T. Russell, M. Weiner, and N. Kerkenbush, "Achieving meaningful use going forward with the ehr and integrated service-lead initiatives," tech. rep., DTIC Document, 2011.

[52] O. Bouhaddou, J. Bennett, T. Cromwell, G. Nixon, J. Teal, M. Davis, R. Smith, L. Fischetti, D. Parker, Z. Gillen, *et al.*, "The Department of Veterans Affairs, Department of Defense, and Kaiser Permanente Nationwide Health Information Network Exchange in San Diego: Patient Selection, Consent, and Identity Matching," in *AMIA Annual Symposium Proceedings*, vol. 2011, p. 135, American Medical Informatics Association, 2011.

[53] M. Hellman, "An extension of the Shannon theory approach to cryptography," *Information Theory, IEEE Transactions on*, vol. 23, no. 3, pp. 289–294, 1977.

[54] M. Johnson, P. Ishwar, V. Prabhakaran, D. Schonberg, and K. Ramchandran, "On compressing encrypted data," *Signal Processing, IEEE Transactions on*, vol. 52, no. 10, pp. 2992–3006, 2004.

[55] D. Schonberg, S. Draper, and K. Ramchandran, "On blind compression of encrypted correlated data approaching the source entropy rate," in *Proc. 43rd Annual Allerton Conf. on Comm., Control, and Computing*, 2005.

[56] D. Schonberg, S. Draper, C. Yeo, and K. Ramchandran, "Toward compression of encrypted images and video sequences," *Information Forensics and Security, IEEE Transactions on*, vol. 3, no. 4, pp. 749–762, 2008.

[57] C. Shannon, "Communication in the presence of noise," *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, 1949.