Low-complexity Compression Techniques for High Frame Rate Video

LOW-COMPLEXITY COMPRESSION TECHNIQUES FOR HIGH FRAME RATE VIDEO

BY

DUO YANG, B.Eng.

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

© Copyright by Duo Yang, May 2017

All Rights Reserved

Master of Applied Science (2017) McMaster University

(Electrical & Computer Engineering) Hamilton, Ontario, Canada

TITLE:	Low-complexity Compression Techniques for High Frame Rate Video
AUTHOR:	Duo Yang B.Eng., (Communication Engineering) Beijing University of Posts and Telecommunications, Beijing, China
SUPERVISOR:	Dr. Jun Chen Dr. Xiaolin Wu
NUMBER OF PAGES:	xiii, 96

This thesis is dedicated to my family,

for their love

Abstract

Recently, video has become one of the most important multimedia resources to be shared in our work and daily life. With the development of high frame rate video (HFV), the write speed from high speed camera array sensor to the massive data storage device has been regarded as the main constraints on HFV applications. In this thesis, some low-complexity compression techniques are proposed for HFV acquisition and transmission.

The core technique of our developed codec is the application of Slepian-Wolf coding theorem in video compression. The light-duty encoder employs SW encoding, resulting in lower computational cost. The pixel values are transformed into bit sequences, and then we assemble the bits on same bit plane into 8 bit streams. For each bit plane, there is a statistical BSC being constructed to describe the dependency between the source image and the SI image. Furthermore, an improved coding scheme is applied to exploit the spatial correlation between two consecutive bit planes, which is able to reduce the source coding rates. Different from the encoder, the collaborative heavy-duty decoder shoulders the burden of realizing high reconstruction fidelity. Motion estimation and motion compensation employ the block-matching algorithm to predict the SI image. And then the received syndrome sequence is able to be SW decoded with SI. To realize different compression goals, compression are separated

to the original and the downsampled cases. With regard to the compression at the original resolution, it completes after SW decoding. While with respect to compression at reduced resolution, the SW decoded image is necessary to be upsampled by the state-of-the-art learning based SR technique: $A+$. Since there are some important image details lost after the resolution resizing, ME and MC is applied to modify the upsampled image again, promoting the reconstruction PSNR. Experimental results show that the proposed low-complexity compression techniques are effective on improving reconstruction fidelity and compression ratio.

Acknowledgements

I would like to thank all the people who has provided help on my thesis. First and foremost, I would like to express the deepest gratitude to my supervisor Dr. Jun Chen for his kindness, encouragement and enthusiasm. His rigorous academic attitudes and passions on research has positively influenced me on my research and life. Being a student of Dr. Chen has been an unforgettable and fortunate experience in my past academic years.

I would also like to thank my co-supervisor Dr. Xiaolin Wu for his support to my graduate studies. His insightful thinking and widened vision has helped me to overcome the difficulties in image/video processing.

I am also very grateful to Dr. Sorina Dumitrescu and Dr. Jiankang Zhang as my defence committee member, also for their instructions on the courses I took.

In addition, I would like to appreciate Dr. Xiao Shu, Dr. Jin Xu, Mr. Mahdi Nangir and Mr. Huihui Wu for their support to my thesis. And I would like to thank all the staff and faculty members in the ECE department. Moreover, I would like to thanks my friends, Guiqiang, Weijie and etc, for our friendships.

Finally, I would like to thank my parents and sister for their unconditioned support and love to me. They inspire me to overcome all the challenges in my life with courage and confidence.

Notation and abbreviations

PSNR	Peak Signal to Noise Ratio
ME	Motion Estimation
ITU	International Telecommunication Union
MEPG	Motion Expert Picture Group
CIF	Common Interface Format
FPS	Frame per Second
SR	Super-resolution
SW	Slepian-Wolf theorem
WZ	Wyner-Ziv theorem
DSC	Distributed Source Coding
LDPC	Low Density Parity Check
EXIT	Extrinsic Information Transfer
BP	Extrinsic Information Transfer
BSC	Binary Symmetric Channel
VND	Variable Node Decoder
CND	Check Node Decoder

PEG	Progressive Edge-Growth
CND	Check Node Decoder
HFV	High Frame Rate Video
CP	Crossover Probability
CE	Conditional Entropy
BMA	Block-Matching Algorithm
MAD	Mean Absolute Difference
MSE	Mean Squared Error
RMSE	Root Mean Squared Error
Bpp	Bit per Pixel

Contents

List of Figures

Chapter 1

Introduction

Video has becomes one of the most important part in our life. As the demand for video contents and qualities increases, video compression has been the key to solve the problem in network usages and video applications. In this chapter, a comprehensive review of video coding techniques is completed and the motivation of this research is summarized. Finally, the outline of this thesis is shown.

1.1 Motivation and Literature Review

Nowadays, with an increasing number of usage of Internet, especially the Mobile Internet, video content services are playing a more significant role in many aspects of our work and life. It is widely admitted that 90% information that travels to the brain is via visual, which relates to the rise of video contents. This is the reason why it becomes so important in social media world. Thus, both researchers and industries have paid high attention to the enormous amount of video data processed in transmission, storage and display. This has become the bottleneck of development

in digital video industry.

Owing to the limited network bandwidth and resources, which may lead to network outage and congestion, multimedia compression on images and videos has drawn the academia's attention. Video compression is a practical implementation of source coding in information theory. Generally, the goal of compression technique on video is to decrease the size of data, aiming at comsuming less bandwidth and storage in hard or cloud drives.

It is known that video data may be transformed to a sequence of still images, called as frames. The size of video could be decreased by applying compression technologies that can cut down the spatial and temporal redundancy. For instances, if there are high correlations among the consecutive frames, the redundancy can be removed by only transmitting and storing the differences between frames. Some other data compression techniques make use of perceptual features of human vision. For example, small differences in color are more difficult to recognize than changes in brightness. Video coding schemes can average the color in a certain area so that the size of data can be shrunk down.

1.1.1 Video Compression Techniques

As stated in [1], the concepts of video compression are introduced and the typical compression techniques proposed by ITU-T Video Coding Experts Group and the ISO/IEC Moving Picture Expert Group are demonstrated. Intra-frame video coding always refers to the reduction of spatial redundancy within one frame. Then it may result in the smaller size of video frame data by storing the critical blocks or pixels that represent the frame. On the other hand, temporal redundancy among video

frames could be eliminated in inter-frame video coding by only coding the differences between consecutive frames.

There are several important conventional compression standards, namely H.261, 263 and 263+, MPEG -1, 2, 4, 7, and H.264/AVC coding standards. H.261, H.263 were developed by ITU respectively in 1990 and 1996. A scheme using motion estimation and compensation was proposed to implement the inter-frame prediction and temporal redundancy elimination in [2], [3], [4], [5]. In order to improve the error resilience and the ability of decreasing the time delay, H.263+ was proposed as an extension of H.263 by [6],[7]. It also allows high frame rate acquisition and enhances the error bit control in [8]. With the development of H.26x standards series, the joint appearance of MPEG-1,2 standards boosted a series of multimedia applications in [9] and [10]. To meet the high demand of new generation of video production in pursuit of less network bandwidth, high-quality broadcast televison, and high-resolution video streaming over the Internet, MPEG-4 was claimed as a new generation video coding standard during 1995-1999 in [11] [12]. It is involved in audio and video transmission and storage systems, and it is also capable of coding individual objects, with high efficiency. Although the performance of video coding with MPEG-4 is good, the high complexity of the system became the constraint in computations mentioned in [13]. Focusing on the visual content description, MPEG-7 put forwards a novel coding system in efficient video searching, recognition, and matching by optimizing video contents descriptors in [14].

Combined with the video coding standards mentioned above, H.264/AVC standard was finally approved as the new efficient and conventional video coding standard

of the ITU-T and the ISO/IEC in 2003. It has been widely adopted in industry during recent years [15],[16]. The advantages of H.264/AVC contains improved entropy coding, motion estimation with variable block size, enhanced motion vectors over picture boundaries and novel multiple reference picture motion compensation, and in-the-loop deblocking filter. Of the benefits mentioned, promoted video compression is the foremost feature in H.264/AVC distinguished from earlier standards. Thus, it has been regarded as the common and international video compression standard for consumers and manufacturers.

1.1.2 Motion Estimation and Compensation

Motion estimation and compensation is an important method to predict the frames by exploiting the correlation between the successive frames. H.264/AVC demonstrates a complete and efficient motion compensated structure to Inter-frame compression. It is one of the most critical techniques for video compression. Inter-frame compression uses one or more earlier or later frames in a sequence to compress the current frame, resulting in temporal redundancy reduction. In contrast of inter-frame coding, intra-frame compression is equivalent to image compression, uses only the current frame and reduces the spatial redundancy .

Therefore, the combination of spatial image compression and temporal motion compensation is contained in most video compression techniques and algorithms. The most commonly used schemes mainly explore the differences between previous frames and current frames. If current frame contains areas where nothing has changed comparing with the earlier frame, the system identifies that there is nothing different between them and simply send the indication to receiver. If there is any moving or

changing sections such as shifting, rotation, brightness changes, the transmitter will send the difference to receiver.

Among the mentioned video compression schemes, motion estimation and compensation are widely applied in inter-frame coding. Owing to the high correlation between the consecutive frames, the spatial and time redundancy could be greatly reduced by finding out the differences between the reference frames and the current frame.

The complete hybrid video encoder of H.264/AVC has been stated in [1], where motion estimation and compensation is the critical prediction technique.

Motion Estimation contains Block-matching algorithm, Optical flow, Phase correlation and frequency domain methods, and Pixel recursive algorithm. All of them make use of the correlation between frames, within which objects could be pixels, blocks, even descriptors and other categories [17],[18]. To meet the requirement for a high speed encoder, the efficiency of a motion estimation algorithm plays a important role. Of the stated schemes in ME, the block-matching algorithm is a widely used technique leading to high efficiency and simplicity [19].

The earlier block-matching algorithm exploited Exhaustive Search (ES), which searches for the matching block within a presumed square region. Later, other new searching schemes such as three-step search (3SS), new three-step search (N3SS) [20], four-step search [21], diamond search [22] and others have been proposed to improve efficiency. Those searching algorithms have been compared in [23], from aspects of different search patterns and searched regions shapes. In this thesis, motion estimation and motion compensation have been employed twice in video decoder. To achieve the highest accuracy when reconstructing, we employ ES to achieve the local

optimum regardless of the high computational cost.

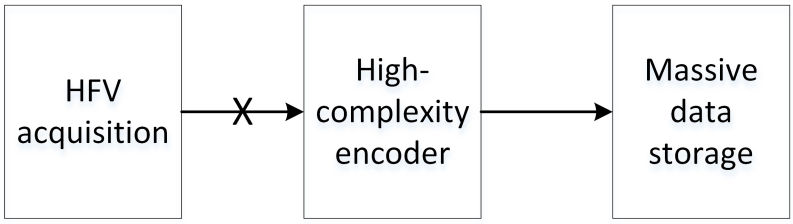
However, the correlation between reference and current frame depends on the frame rate, moving speed, video resolution, and complexity of the scene. It may have difficulties in applying ME and MC techniques especially the block-matching scheme.

1.1.3 High Frame Rate Video Coding

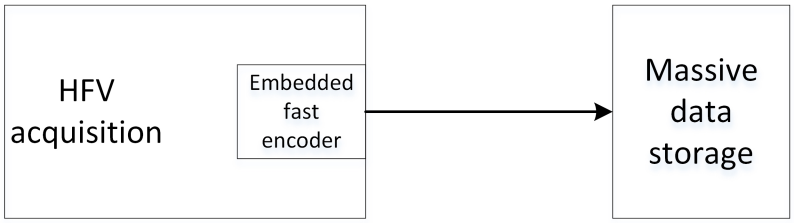
MPEG-X and H.26X series video coding standards mainly draw their attention on scalable video coding so as to achieve high compression efficiency, high flexibility (bandwidth scalability) and/or low complexity. However, in the application of high speed camera, the storage and write speed constraints become the bottleneck of this new video production.

With the development of video industry, high frame rate video is applied to high speed manufacturing, medical and research fields. Usually, video camera can record at 24 to 30 fps, which is sufficient when movement is slow or changes between consecutive frames cannot be found easily. But in recording fast action and play back in slow motion, details of fast action will be visible. Meanwhile, with the significant increase in frame rate which may reach hundreds even thousands of frames per seconds, the recording, transmitting, and storing of high frame rate video will be a new challenge to industry [24]. To obtain a high speed and continuous video acquisition, the memory bandwidth from the sensor array to the mass data storage device is the primary limit.

In acquisition of video, the write speed of on-camera mass storage device is only 500 MB/second, which is much slower than the required one in ultra-high speed camera. For instance, for a video frame of 2M pixels of 8-bit grey level, the raw output throughput may reach 5000 MB/s with 2500 fps, which is an order of magnitude faster



(a) Transmission from HFV acquisition sensor to massive data storage via high-complexity encoder.



(b) Embedded ligh-duty encoder facilitate the transmission from HFV acquisition sensor to massive data storage.

Figure 1.1: Designs of high speed camera acquisition system

than the write speed limit. To overcome the limit, the commonly applied method is changing spatial and temporal resolutions. Therefore, the write speed from high-speed camera acquisition sensor to massive data storage device is the main constraint and challenge.

Compressing the video data from the raw output data throughput is an accessible way to remove the bottleneck of memory bandwidth. Obviously, an efficient and fast compressor/encoder performed in real time is necessary to keep up with the speed of data generation. Besides the high compression efficiency, the reconstruction accuracy at the decompressor/decoder should also keep in high fidelity. That's because most high speed videos are exploited in research and medical proposes.

However, typical video compression standards are not able to meet the requirement of write speed and high fidelity in Fig. 1.1 (a). The video encoder in H.264/AVC requires sophisticated systems. All existing compression schemes containing a three-step process: decorrelation, quantization, and entropy coding. The decorrelated frame/image is quantized and then entropy coded via either Huffman or arithmetic coding. Entropy coding may lead to expensive computation cost. In [25], Chuah et al. developed an asymmetric DPCM-based near-lossless compression technique to eliminate the memory bandwidth obstacle of high speed camera. Nevertheless, the combination with the entropy coding in the proposed scheme is also a handicap to satisfy the ultra-high speed video.

To overcome the difficulty of encoder obstacles in typical video compression standards, we design a novel video codec with light-duty, hardware-friendly encoder and a collaborative relatively heavy-duty decoder, in which the faster write speed, better compression ratio and higher reconstruction fidelity could be satisfied. The light-duty encoder could be embedded in the HFV acquisition sensor illustrated in Fig. 1.1 (b). Distributed source coding techniques are exploited in our scheme, which may improve the computation performance of the encoder and the reconstruction level of the decoder. Applied with Slepian-Wolf coding method, in fast encoder, the chosen source codes stated in [26] generate the parity-check matrix $\mathbf{H}$. With that matrix, the data source is transformed to a shorter-length code, without decorrelation, quantization and entropy coding. This implementation only involves the operations using exclusive-OR adders and simple logic circuits, facilitating the write speed from camera sensor to the storage device. Different from the simple encoder, the burden of reaching better compression ratio and reconstruction fidelity of the new codec has

shifted to the heavy-duty decoder. To achieve better compression ratio, the compression can be implemented on the downsampled image. At the decoder, ME and MC are employed to generate the side information with previous and subsequent consecutive frames. After Slepian-wolf decoding, the decoded downsampled image is upsampled by approaches such as simple 'bicubic' ,'box' interpolation, and super-resolution (SR). Finally, ME and MC are applied again to improve the reconstruction fidelity.

1.1.4 Distributed Source Coding

In the designed codec, distributed source coding (DSC) schemes shift the ME and MC in traditional video encoders to the decoder, and comes up with a light-duty encoder. DSC contains Slepian-Wolf (SW) theorem [27] and Wyner-Ziv (WZ) theorem [28]. The former one demonstrates a lossless compression scheme, while the other one proposes a lossy compression, with side information at the decoder.

The decoder will obtain the prediction of video frame by temporal interpolation techniques such as ME and MC. The consecutive frames are used to estimate the side information (SI) as the vital information to recover the original image/frame. SW and WZ coders both exploit the dependency error between the original frame and the estimated SI to model a virtual channel. The latter author extended the SW theorem to lossy compression. As shown in Fig. 1.2, $\boldsymbol{X}$ is the original image/frame (it would be quantized in WZ), $\boldsymbol{Y}$ is the decoder's SI predicted by temporal estimation and compensation. Based on their dependency , SW coding is able to be implemented [29]. In order to retain high fidelity in reconstructing the original frame, Slepian-Wolf coding is adopted to achieve the lossless compression over the modeling dependency channel.

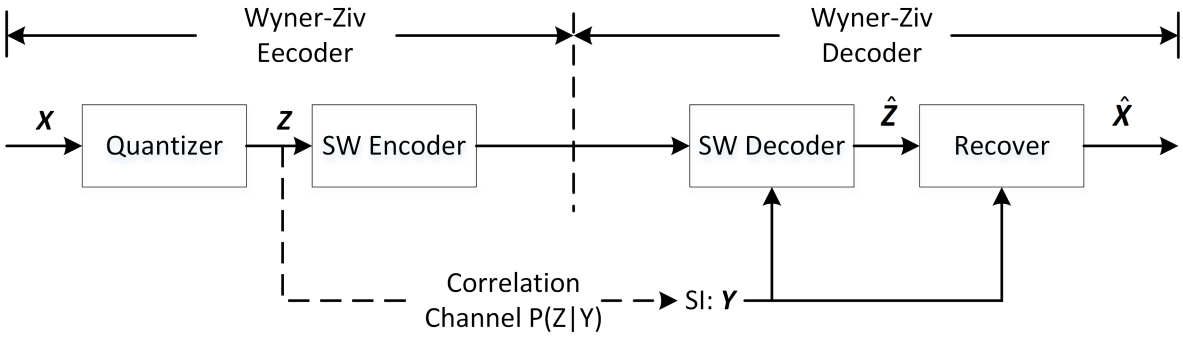


Figure 1.2: A distributed source coding system for lossless and lossy compression

Syndrome codes, Turbo Codes, and LDPC Codes are typical options to implement a DSC coder, of which LDPC code is commonly applied [30]. In a practical distributed video coding system, an appropriate coding rate applied in encoder which compresses the source data to the syndromes is a necessity for the decoder to recover the original image. If the channel model used at the decoder is an inaccurate representation of the actual dependency channel, sometimes a higher coding rate need be adopted to achieve the lossless reconstruction with the loss of compression ratio. Therefore, a good statistics of dependency channel that referred to the $P(X|Y)$ becomes an important factor in SW coding application.

Especially with regard to the video signals, there are several early architectures and implementations of distributed video coding, such as the method in [31],[32] by Girod et al. from Stanford University , $PRISM$ in [33],[34] by Puri et al. from Berkeley University, $DISCOVER$ in [35] from a European project. All of them exploit the dependency between the video frames. In the methods proposed by Stanford and Berkeley, the applied noise model is obtained via offline training. With respect to $DISCOVER$, the noise model is obtained online. Although the statistics of different video signals may vary a lot, those three scheme are still able to provide a reliable

distributed video coding system. Moreover, $PRISM$ and $DISCOVER$ also need to send a feedback from the decoder to the encoder, which is a constraint for real-time video coding.

1.2 Contribution and Organization of the Thesis

In this work, a new video codec combining a light-duty encoder and a collaborative heavy-duty decoder is developed to satisfy the requirement of high speed camera. To remove the obstacle during the transmission from camera sensor to massive storage device, ME and MC is shifted to the decoder. In addition, the SW encoder results in lower computation cost, by only involving the simple adder and XOR operations. Also, compression at different spatial resolutions is flexible to satisfy different compression requirements. An improved coding scheme which exploits the dependency between two consecutive bit planes is proposed to reduce the coding rate. Because most high speed videos are used for scientific and medical purposes, high reconstruction fidelity is vital in our codec. ME, MC and SW decoding at the decoder have ability to guarantee the lossless decompression (guarantee the bit planes we code are error free). Furthermore, an advanced upsampling technique such as SR is utilized to reconstruct the original frame from the SW decoded version. A further modification at corners and boundaries of the decoded image is resolved by ME and MC, achieving a higher fidelity - PSNR.

This thesis is organized as follows:

- Chapter 2 introduces some important theorems and techniques used in our developed codec, such as SW Coding Theorems, LDPC Codes, Belief Propagation

Algorithm, and EXIT charts technique. They have been applied to Degree Distribution generation, and SW encoding/decoding.

- Chapter 3 mainly describes a light-duty encoder which achieves an accessible and efficient transmission from camera sensor to storage device.The compression are implemented at different resolutions. And an improved coding scheme is developed by exploiting the spatial dependency between consecutive bit planes, and results in the decrease of the source coding rate.

- Different from the fast encoder, Chapter 4 takes the relatively heavy burden in reconstructing the original image/frame. ME and MC have been employed twice, by firstly recovering the SI for SW decoder and secondly further modifying image details. Meanwhile, as the advanced US technique, SR aims to upsample the SW decoded image, and outperforms the other conventional upsampling interpolations such as 'box','bilinear', 'bicubic', etc.

- The experimental results are presented in Chapter 5. With respect to different compression requirements, the trade-off between compression ratio and reconstruction fidelity is considered.

- A conclusion is drawn in Chapter 6. Lower complexity at the encoder, higher compression ratio and better image fidelity are satisfied in the proposed codec. In the future work, an effective estimation of source coding rate is supposed to be taken into consideration. Also the optimization of new degree distributions is ought to be resolved.

Chapter 2

Preliminaries

In this chapter, the Slepian-Wolf (SW) coding theorem is introduced, as the key technique in the proposed method. Some other conventional techniques related to SW coding such as LDPC Codes, Belief Propagation (BP) Algorithm and Extrinsic Information Transfer Charts are stated as well.

2.1 Slepian-Wolf Coding

In the landmark paper [27], Slepian and Wolf developed the SW coding theorem. With regard to the multiple correlated sources but may not communicate with each other, SW coding is able to be applied in the data compression.

In the distributed source compression system illustrated in Fig. 2.1, two random processes X and Y are dependent with a statistically joint distribution $P(X, Y)$. And in the DSC system, the correlated sources may not communicate with each other, then they are encoded by SW encoder separately. When the transmitted bit streams are received by the decoder, they are able to be jointly decoded to reduce

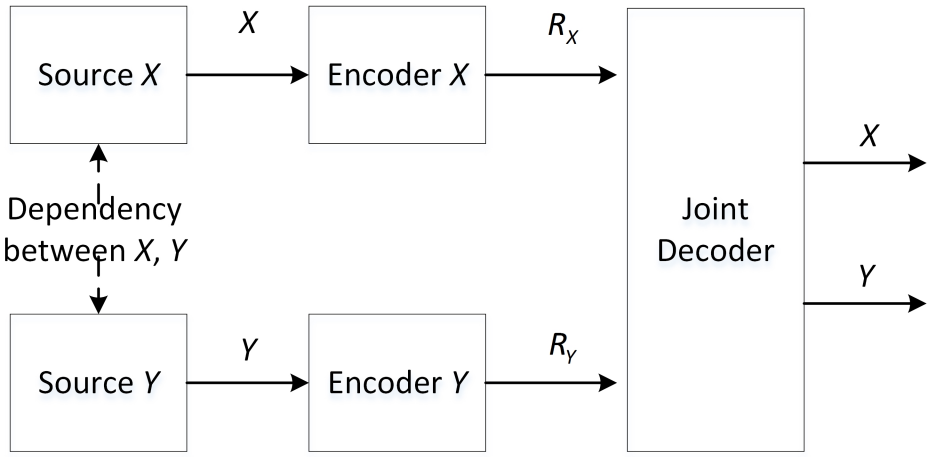


Figure 2.1: Distributed compression system with two correlated sources X and Y

the coding rates in practical use. The joint decoder makes the most of the statistical dependency between the correlated sources.

Considering two statistically dependent sources X and Y, which are independent identically distributed (i.i.d.) generated in Fig. 2.2, the dark gray region represents the independent coding, in which two uncorrelated sources is capable of being encoded and decoded separately. To achieve the lossless compression, Shannon proves that

$$\begin{aligned} R_X &\geq H(X), \\ R_Y &\geq H(Y), \end{aligned} \tag{2.1}$$

where R_X and R_Y are the coding rates of random variables X and Y. $H(X)$ and $H(Y)$ denote the entropies of two independent sources, respectively. In addition, we explore the dependency between the correlated X and Y to decrease the coding rates in transmission. Two distributed random sources may be encoded dividedly, then decoded jointly [27]. The light gray region in Fig. 2.2 represents the individual achievable rates region for the joint SW coder, where R_X and R_Y should be bounded

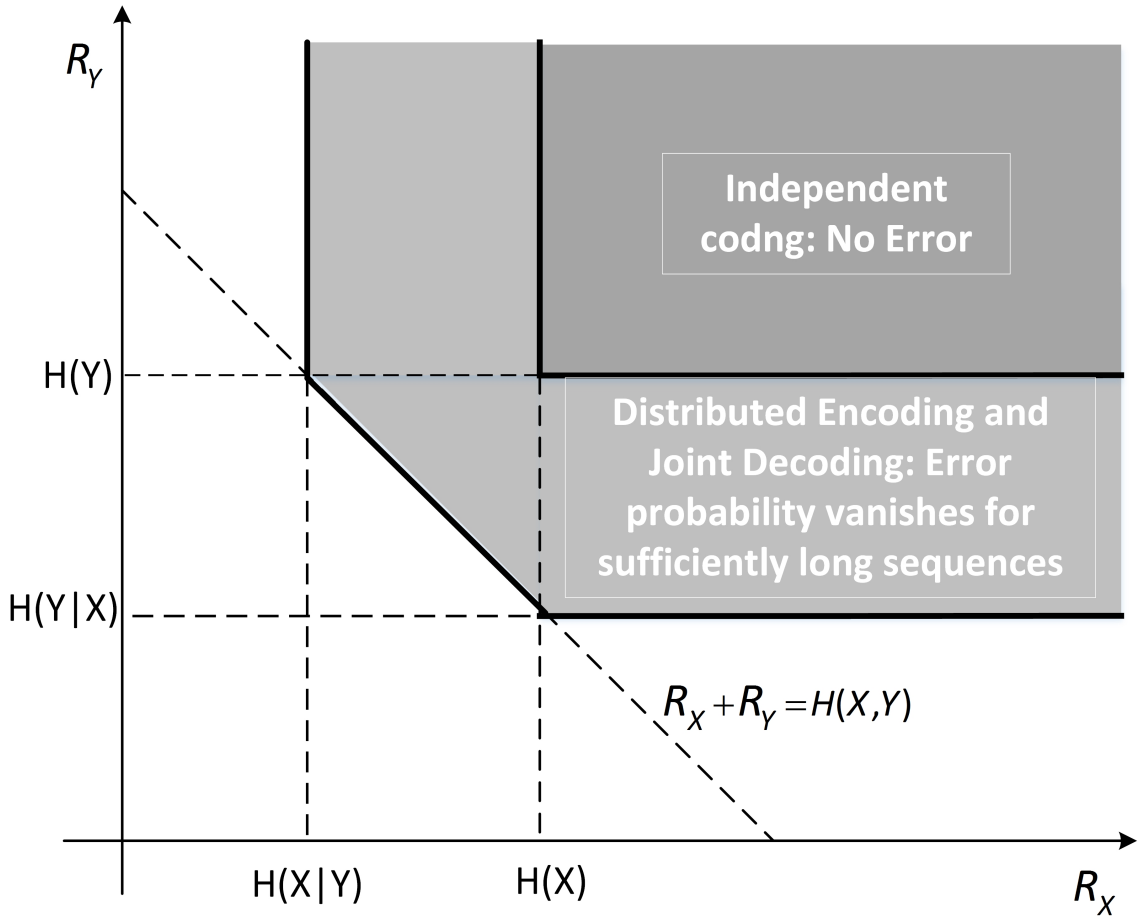


Figure 2.2: SW theorem: achievable rate region for two statistically dependent i.i.d. sequences in DSC system

as follows:

$$\begin{aligned} R_X &\geq H(X|Y), \\ R_Y &\geq H(Y|X), \\ R_X + R_Y &\geq H(X,Y). \end{aligned} \tag{2.2}$$

$H(X|Y)$ and $H(Y|X)$ are the conditional entropies. [27] has shown that it is possible to compress the correlated sources at rates no longer than those needed. Both X and Y are able to be reconstructed with asymptotically vanishing error probability for long sequences. Beside that, the sum rate $R_X + R_Y$ represents the joint encoder rather than the separate encoder.

A special case in Fig. 2.3 is the distributed SW compression of random generated sequence $\boldsymbol{X} = (X_1, X_2, ..., X_n)^T$ with the SI $\boldsymbol{Y} = (Y_1, Y_2, ..., Y_n)^T$. As distributed video coding system shows, $\boldsymbol{X}$ depends on SI $\boldsymbol{Y}$. The difference is that the $\boldsymbol{Y}$ is only available at the SW decoder, not known at the SW encoder. If R_X is higher than $H(X|Y)$ at the encoder, the original sequence $\boldsymbol{X}$ will be reconstructed perfectly with the help of SI $\boldsymbol{Y}$, since $R_Y = H(Y|X)$ is satisfied for encoding $\boldsymbol{Y}$. In our case, the SW coder is applied based on LDPC to compress the hash code.

The $\boldsymbol{X}$ and $\boldsymbol{Y}$ denote the original image and the SI image , between which there may be a Binary Symmetric Channel (BSC) model being constructed. A crossover probability in the modeled BSC is statistically detectable. However, they may vary in different data sources. After detecting the statistical crossover probability for each bit plane in practice, so the encoder has ability to realize SW encoding and compress the data source.

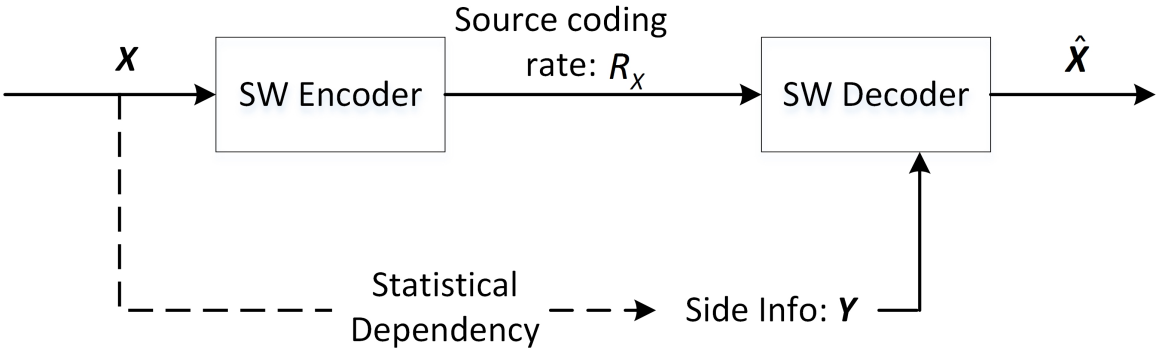


Figure 2.3: Compression of random source X with side information Y, which is only known at the decoder.

2.2 Low Density Parity Check Codes

The performance of a SW coder based on LDPC comes close to the SW bound. The application of low density parity check codes in distributed source coding, such as Slepian-Wolf problem, attracts the academia's attention. Low density graph codes such as LDPC and low density generator matrix (LDGM) codes are both graphically constructed. Here, LDPC is employed in SW coder [36]. LDPC is developed by Gallager in 1960's [37]. Moreover, LDPC codes are able to be employed in applications such as 10GBase-T Ethernet, which sends data at 10 GB/s over twisted-pair cables, Wi-Fi 802.11 standard and OFDM system.

The sparsity of the low density graph construction improves the encoding and decoding performance efficiently. Conventionally , LDPC codes are linear codes derived from the sparse bipartite graph, which is defined as follows:

$$\mathbb{C} = \{\boldsymbol{x} \in \{\mathbf{0,1}\}^n : \mathbf{H}\boldsymbol{x} = 0\}, \tag{2.3}$$

where $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T$ is a linear codeword. Accordingly, $\mathbf{H}$ is an $m \times n$ sparse

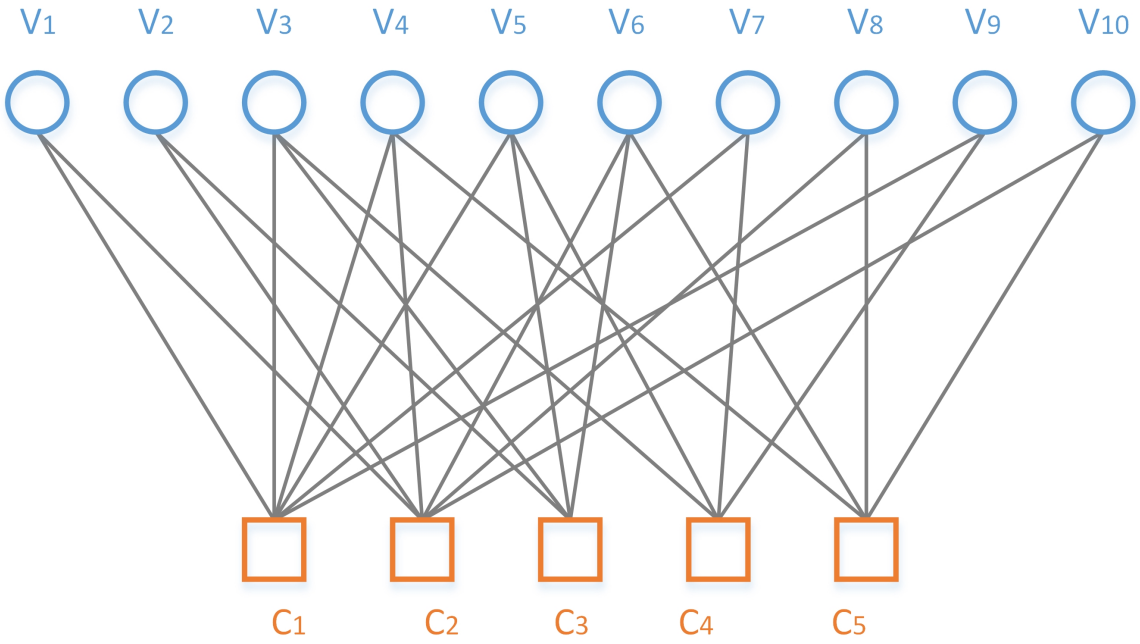


Figure 2.4: Bipartite graph of a LDPC code's construction.

matrix called parity-check matrix, and the code is given by its null space. The bipartite graph, which is also called factor graphs , which was first suggested by Tanner to capture the LDPC code structure (see Fig. 2.4). The following equations are supposed to be satisfied:

$$\begin{aligned} x_1 + x_3 + x_4 + x_5 + x_7 + x_9 &= 0, \\ x_1 + x_2 + x_4 + x_6 + x_8 + x_{10} &= 0, \\ x_2 + x_3 + x_5 + x_6 &= 0, \\ x_3 + x_5 + x_7 + x_9 &= 0, \\ x_4 + x_6 + x_8 + x_{10} &= 0. \end{aligned} \tag{2.4}$$

Hence, with the group linear equations, we can obtain the LDPC codes set. The sparse matrix **H** indicates that the number of **1** is sparsely distributed. The bipartite

graph in Fig. 2.4 illustrates the code structure of an LDPC code that is also analogous to a matrix representation. In the (n, m) Tanner graph construction, m and n are the number of check nodes and variable nodes, respectively. On variable nodes side, each node v refers to one code symbol in the n-length codeword, and on check nodes side, each c represents the parity check constraint specified by one row of $\mathbf{H}$ with XOR operations , which corresponds to the constraint equations in (2.4). If the entry (i, j) of parity check matrix $\mathbf{H}$ is 1, it indicates that the j-th variable node connects to the i-th check node by an edge in the bipartite graph.

LDPC codes contains two types: regular and irregular. The regular LDPC codes means that not only the numbers of check nodes connected to each variable node are the same as d_v, but also the numbers of variable nodes connected to each check node are the same as d_c. In other words, the numbers of 1 in each column and row are d_v and d_c, respectively. However, the irregular codes mentions that the number of adjacent variable and check nodes to each check and variable nodes may be different. Although it is easier to generate regular LDPC codes than irregular ones, the irregular LDPC codes outperform the regular codes in distributed coding system. It is supposed to follow that $m < n$ to satisfy $r < 1$. The variable node and check node degree distributions in the factor graph are represented as λ and ρ, respectively. Furthermore, they are specified as following polynomial expressions [38]:

$$\begin{aligned} \lambda(x) &= \sum_{i\geq 2}^{d_v^{max}} \lambda_i x^{i-1}, \\ \rho(x) &= \sum_{i\geq 2}^{d_c^{max}} \rho_i x^{i-1}. \end{aligned} \tag{2.5}$$

Precisely, λ_i presents the fraction of edges emanating from variable nodes of degree i,

and ρ_i represents the fraction of edges emanating from check nodes of degree i. It is noted that, $\lambda_i(\rho_i)$ corresponds to x^{i-1} rather than x^i. Later, the degree distribution design and optimization will be discussed.

2.3 Belief Propagation Algorithm

The low density - sparsity of the bipartite graph structure is the crucial feature of LDPC codes, with which the encoding and decoding efficiency of DSC is achieved. SW Decoder adequately exploits the sparsity in employing Message Passing Algorithms.

Message passing algorithm, which is also called iterative algorithm, is one of the general decoding scheme associated with the LDPC codes. According to the bipartite graph structure of LDPC codes, each iteration of the algorithm completes the messages not only passing from the check nodes to the connected variables, but also from variables to the connected check nodes. The message passed from variable(check) nodes to check(varible) nodes is computed with its observed message and the messages passing from the adjacent nodes in previous iteration. The message usually represents the probabilities or log-likelihood ratio. Here BP is employed to decode the original source with the dependent SI source. An important rule to be followed is that, the message passing from variable node v to check node c is calculated with the messages emanating from neighboring check nodes in previous iteration except for c itself. It works for the message passing from c to v, too.

Belief Propagation (BP) algorithm is one of message passing algorithms, in which the message is specified as the probability or beliefs of the variable(check) node. Here, likelihood or sometimes log-likelihood (LLR) is more convenient to compute in iterative passing.

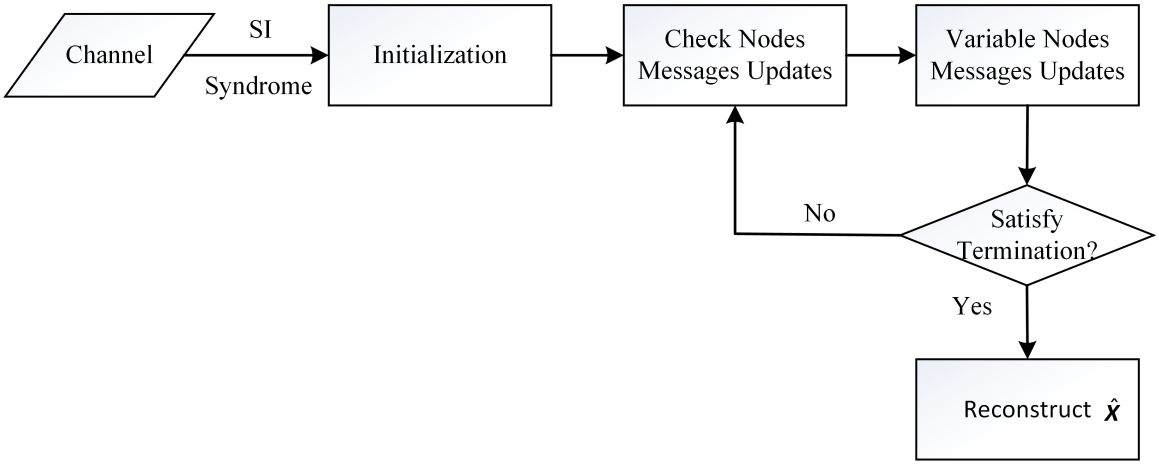


Figure 2.5: The flowchart to show how Belief Propagation Algorithms works in SW decoder

Considering two dependent sequences $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)^T$ and $\boldsymbol{Y} = (Y_1, Y_2, \ldots, Y_n)^T$ at the length n. The syndrome $\boldsymbol{S} = (S_1, S_2, \ldots, S_m)^T$ at the length m is obtained from $\boldsymbol{S} = \mathbf{H}\boldsymbol{X}$. The LR here denotes the conditional likelihood, defined as $\frac{P(X=0|Y)}{P(X=1|Y)}$, while LLR here denotes the conditional log-likelihood, defined as $log\frac{P(X=0|Y)}{P(X=1|Y)}$.

It starts from the variables passing messages to check nodes. First of all, the initial message is computed from the observed value of SI. Therefore, at 0-th iteration, the initial messages are obtained as [38]:

$$M_i^0 = log\frac{P(X_i = 0|Y_i)}{P(X_i = 1|Y_i)}. \tag{2.6}$$

In each l-th iteration, $M_{v\to c}$ and $N_{c\to v}$ are the messages passing from variable to check node and from check to variable node, respectively. More precisely, the messages passing from variable node v_i to the adjacent check node c_j are computed by:

$$M_{v_i\to c_j}^{(l)} = M_i^0 + \sum_{c_r \in C_{v_i}\backslash\{c_j\}} N_{c_r\to v_i}^{(l)}, \tag{2.7}$$

where v_i is included in the variable set $\{v_1, v_2, \ldots, v_n\}$, and c_j is included in the check nodes set $\{c_1, c_2, \ldots, c_m\}$. C_{v_i} is the set of all check nodes connected to variable v_i, and c_r is the neighboring check node in set C_{v_i} but except for c_j itself. The superscript l means at l-th iteration. Meanwhile, M_i^0 is obtained from (2.6), which is only relevant to the observed value Y_i.

Then the messages passing direction is from check nodes to variable nodes. Precisely, the message sent from c_i to the neighboring variable node v_j is defined as follows:

$$N_{c_i \to v_j}^{(l)} = ln \frac{1 + \prod_{v_r \in V_{c_i} \setminus v_j} tanh(\frac{M_{v_r \to c_i}^{(l-1)}}{2})}{1 - \prod_{v_r \in V_{c_i} \setminus v_j} tanh(\frac{M_{v_r \to c_i}^{(l-1)}}{2})}, \tag{2.8}$$

where V_{c_i} is the set of all variable nodes incident to check node c_i, and v_r is the neighboring check node in set C_{v_i} but except for v_j itself. From the equations (2.7) and (2.8), the rule mentioned above has been satisfied.

After each iteration completes, the reconstructed value $\hat{X}_i$ is able to be predicted as the following judgment:

$$\hat{X}_i = \begin{cases} 0, & \text{if} \quad M_i^0 + \sum_{c_r \in C_{v_i}} N_{c_r \to v_i}^{(l)} \geq 0, \\ 1, & \text{if} \quad M_i^0 + \sum_{c_r \in C_{v_i}} N_{c_r \to v_i}^{(l)} < 0. \end{cases} \tag{2.9}$$

Then with (2.9), the estimated sequence $\hat{\boldsymbol{X}} = \{\hat{X}_1, \hat{X}_2, \ldots, \hat{X}_n\}$ are obtained after each iteration completes. When the iteration reaches a certain round or the beliefs on variables reach a certain value, the running iteration will terminate. Belief propagation algorithm itself does not depend on the channel at all. However the initial messages are obtained according to the chosen channel.

2.4 Extrinsic Information Transfer Charts

With the development and application of LDPC codes, the design of good degree distributions for generating LDPC codes for various channels has arisen. Extrinsic Information Transfer (EXIT) Charts is a technique to construct a good degree distribution of LDPC codes. EXIT charts was proposed by Stephan ten Brink, et al [39],[40].

In this technique, there are two types of decoders: variable node decoder (VND) and check node decoder (CND). The extrinsic messages are passed between VND and CND. With respect to the information that exchanges between the two decoders in each iteration, I_A is defined as the apriori knowledge of the decoder, and the $I_E = T(I_A)$ is the extrinsic information function of I_A, which becomes a posteriori knowledge [41]. Moreover, the extrinsic information transfer function is developed to compute an extrinsic posteriori information with the apriori messages. The details about the structure of iterative EXIT chart information exchange is presented in Fig. 2.6 [40].

More precisely, $I_{E,VND} = T_1(I_{A,VND)}$ is the extrinsic output of VND, which is the mutual information of the extrinsic output messages of VND. It will then be forwarded to CND, as a priori information $I_{A,CND} = I_{E,VND}$. For the CND, $I_{E,CND} = T_2(I_{A,CND)}$ is the mutual information of the extrinsic output messages of CND, which would be sent back to VND regarded as the priori knowledge at next iteration.

The classic EXIT charts is designed for the AWGN channel. When the degree distribution is defined as (2.5), then the extrinsic transfer functions T_1 and T_2 are

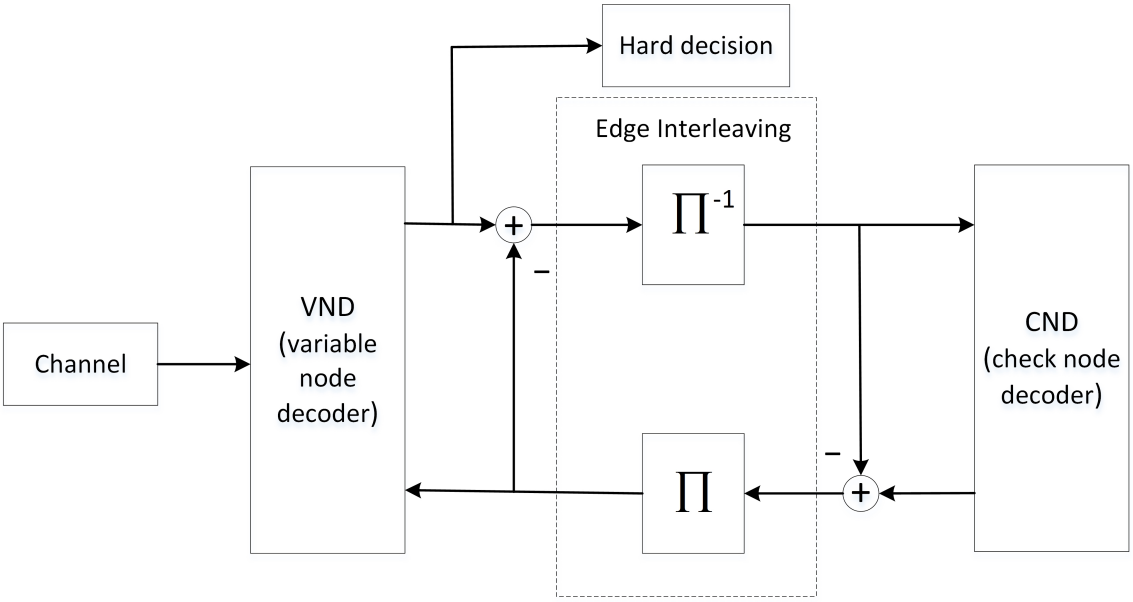


Figure 2.6: Iterative decoder in optimizing a good degree distribution.

defined as follows [40][42]:

$$I_{E,VND} = \sum_{i\geq 2}^{d_v^{max}} \lambda_i \cdot J(\sqrt{(d_{v,i}-1)[J^{-1}(I_{A,VND})] + \sigma_{ch}^2}), \tag{2.10}$$

where σ_{ch} is the variance of the channel message, which is obtained from σ_n^2 the noise variance of AWGN channel : $\sigma_{ch}^2 = \frac{4}{\sigma_n^2}$. Also, $d_{v,i}$ is the degree-i. Then for the extrinsic transfer function in CND, it is defined as:

$$I_{E,CND} = 1 - \sum_{i\geq 2}^{d_c^{max}} \rho_i \cdot J(\sqrt{(d_{c,i}-1)} \cdot J^{-1}(I_{A,CND})). \tag{2.11}$$

The functions $J(\cdot)$ and $J_{-1}(\cdot)$ in (2.10) and (2.11) are computed with [40]:

$$J(x) \approx \begin{cases} a_{J1}x^3 + b_{J1}x^2 + c_{J1}x, & if \quad 0 \leq x \leq x^*, \\ 1 - e^{a_{J2}x^3 + b_{J2}x^2 + c_{J2}x + d_{J2}}, & if \quad x^* < x < 10, \\ 1, & if \quad x \geq 10, \end{cases} \tag{2.12}$$

where $x^* = 1.6363$. The inverse mutual information function $J_{-1}(\cdot)$ is presented by:

$$J^{-1}(y) \approx \begin{cases} a_{J^{-1}1}y^2 + b_{J^{-1}1}y + c_{J^{-1}1}\sqrt{y}, & if \quad 0 \leq y \leq y^*, \\ a_{J^{-1}2}ln[b_{J^{-1}2}(1-y)] - c_{J^{-1}2}y, & if \quad y^* < y < 1, \end{cases} \tag{2.13}$$

where $y^* = 0.3646$. Those coefficients are described as:

$$\begin{cases} a_{J1} = -0.421061, b_{J1} = 0.209252, c_{J1} = -0.00640081, \\ a_{J2} = 0.00181491, b_{J2} = -0.142675, c_{J2} = -0.0822054, d_{J2} = 0.0549608, \\ a_{J^{-1}1} = 1.09542, b_{J^{-1}1} = 0.214217, c_{J^{-1}1} = 2.33727, \\ a_{J^{-1}2} = 0.706692, b_{J^{-1}2} = 0.386013, c_{J^{-1}2} = -1.75017. \end{cases} \tag{2.14}$$

The EXIT charts is built to predict the convergence behavior of iterative LDPC codes. When optimizing a degree distribution, the objective is to maximize the coding rate $R = 1 - \frac{m}{n}$. Meanwhile, the criterion should be satisfied as long as the EXIT function of VND $I_{E,VND}$ is above the inverse EXIT function of CND $I_{A,CND}$, which are computed by (2.10) and (2.11). The curves are indicated in Fig. 2.7. In addition, the curve fitting approach is developed to obtain the convergence threshold. That is because $I_{E,VND}$ is a decreasing function with regard to σ_n^2, but $I_{C,AND}$ is irrelevant to σ_n^2. Therefore, the convergence threshold could be obtained when the two curves

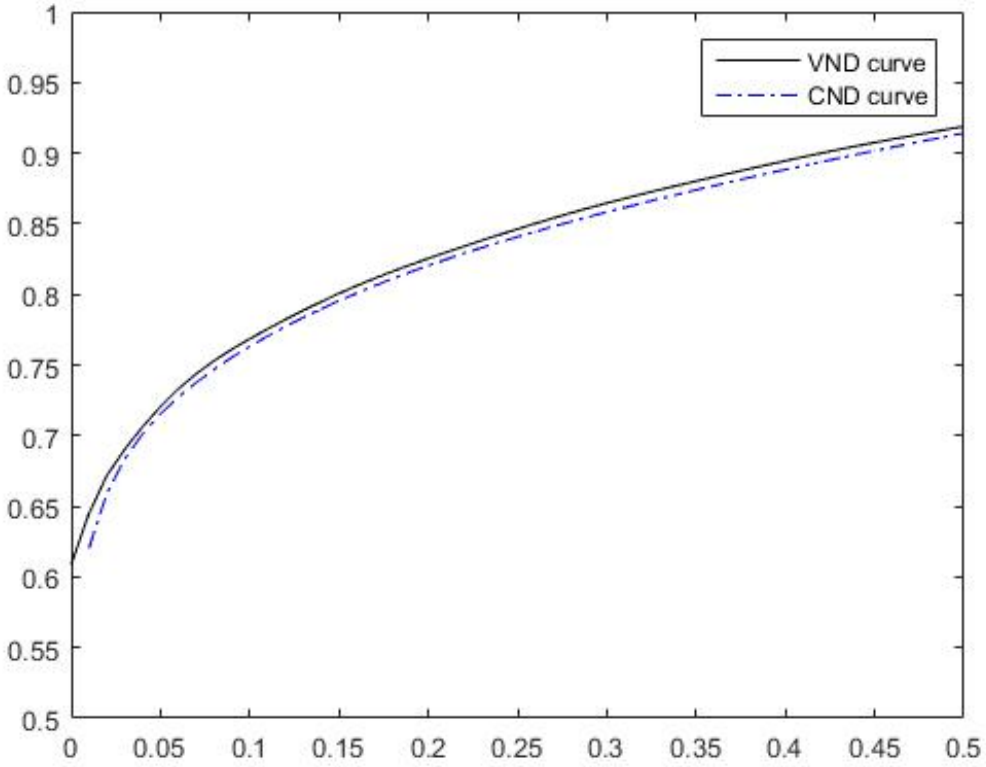


Figure 2.7: Curve fitting for an LDPC code at rate = 0.6

approximately meet at a certain noise variance $\sigma^2_{n,0}$.

Chapter 3

Light-Duty Fast Encoder

With the development of high frame rate video, and the increasing demand of application for fast and high-fidelity videos, the high speed and continuous video acquisition is the key technology need to be promoted so as to meet the above requirements. However, the memory bandwidth from the sensor array to the on-board mass data storage device blocks the real-time transmission. As shown in Fig. 1.1, common solution is to design a heavy-duty encoder which compresses the raw video data by three steps: decorrelation, quantization, and entropy coding. Obviously, the conventional method is not able to keep up with the high speed in recording the video, resulting in the constraints on data writing into the storage device.

To solve the problem above, a novel video codec with light-duty, hardware-friendly encoder and a collaborative relatively heavy-duty decoder is developed, in which the faster write speed, better compression ratio and lower computational complexity are able to be satisfied. With regard to the proposed fast encoder, the DSC technique is the core part: Slepian-Wolf coding compression scheme is capable of compressing the data source with an efficient computation. Also, the compressions have been

separated to two cases, at original resolution and reduced resolution. Moreover, by exploiting the dependency between two consecutive bit planes, an improved coding scheme is put forward to decrease the source coding rate.

3.1 Slepian-Wolf Encoding

The key technique to achieve the light-duty encoder is applying SW encoding. As mentioned above, SW encoder is able to exploit the dependency between multiple correlated sources.

The general SW coding theorem meets the criterion (2.2). The situation in Fig. 2.3 is the distributed SW compression of two correlated sources X and Y. $\boldsymbol{X}$ is able to represent the original image , produced by the source X. Also, $\boldsymbol{Y}$ represents the side information image, produced by the source Y. As explained before, the SI image/frame is available at the decoder. With regard to the original data source X, its coding rate R_X is required to be higher than $H(X|Y)$ at the encoder, so as to reconstruct the original image perfectly with the help of side information.

With respect to a RGB image, each pixel in one 8-bit-depth gray image, values from 0 to 255, which is too complicated to implement with SW coding theorem. That is because the computational cost with BP algorithm is propotional to the value ranges of variables.

Hence, to facilitate the SW encoding, the original data image and side information image of decoder need to be transformed into the binary sequences. Obviously, each 8-bit-depth pixel is transformed to a 8-bit-length bit sequence. Besides, each bit in the 8 bit planes counts differently, and the significance decreases from leftmost bit to the rightmost bit. Conventionally, the leftmost bit is called as the most significant bit,

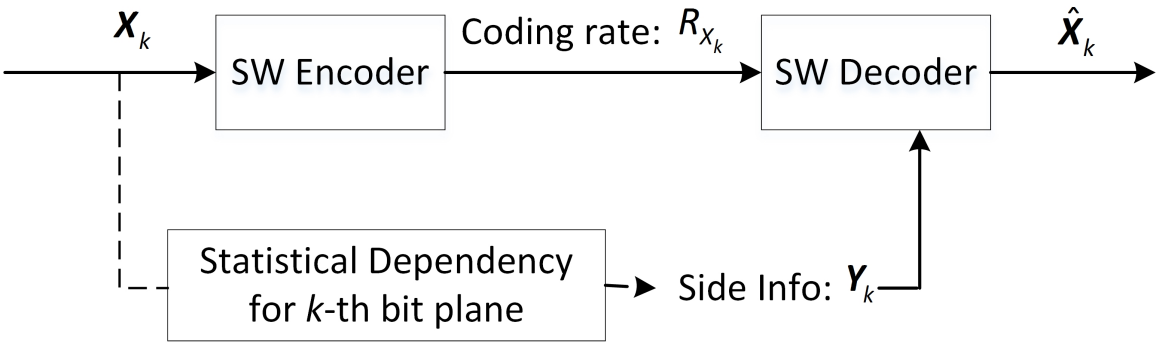


Figure 3.1: Compression of original bit sequence $\boldsymbol{X}_k$ with side information sequence $\boldsymbol{Y}_k$, which is only known at the decoder.

and the rightmost bit as the least significant bit. Here, define the bits from left to right positions as : 1^{st}, 2^{nd}, 3^{rd}, 4^{th}, 5^{th}, 6^{th}, 7^{th}, 8^{th} bit, respectively. Assemble the bits in the same position for all the pixels in one frame, there are 8 bit streams generated , also called 8 bit planes, which are defined as $\boldsymbol{X}_k \in \{\boldsymbol{X}_1, \boldsymbol{X}_2 \ldots, \boldsymbol{X}_8\}$. More precisely, each bit stream contains n bits (n is the number of all pixels in one frame). Each one bit represents the transformed bit of different pixels in the same bit planes. Then the element of bit sequence $\boldsymbol{X}_k$ is defined as $x_{k,i} \in \{x_{k,1}, x_{k,2}, \ldots, x_{k,n}\}$. Similarly, the side information image $\boldsymbol{Y}$ also includes 8 bit streams: $\boldsymbol{Y}_k, k \in \{1, 2 \ldots, 8\}$, and each element is defined as $y_{k,i}, i \in \{1, 2, \ldots, n\}$. As depicted in Fig. 3.1, the SI is only known at the decoder. The original image and side information have been transformed to the bit streams. Since each pair of $\boldsymbol{X}_k$ and $\boldsymbol{Y}_k$ are corrlated, then there may be a Binary Symmetric Channel (BSC) model being constructed to describe their dependency in each pair. In addition, there is a crossover probability $p_k = p(X_k|Y_k)$ being statistically detected in the modeled BSC . Afterwards, there are 8 different BSC models being generated with 8 different crossover probabilities. More precisely, X_k and Y_k are the random variables taken values in $\{0, 1\}$, in k-th bit stream $\boldsymbol{X}_k$

and $\boldsymbol{Y}_k$ repesctively. In order to decode the split source data without error when the source length tends to infinity, the coding rate R_{X_k} for the k-th bit plane is supposed to be equal or larger than the conditional entropy $H(X_k|Y_k)$, specified as:

$$\begin{aligned} H(X_k|Y_k) = H(p_k) = -p_k \cdot log_2(p_k) - (1-p_k) \cdot log_2(1-p_k). \\ R_{X_k} \geq H(X_k|Y_k). \end{aligned} \tag{3.1}$$

After detecting the conditional entropy of the modeled BSC, a degree distribution should be optimized to generate a parity-check matrix with lower coding rate. With the appropriate parity-check matrix $\mathbf{H}$, and m and n are the dimensions of $\mathbf{H}$. Afterwards, a syndrome sequence is generated with a XOR operations:

$$\boldsymbol{S}_k = \mathbf{H}_k \cdot \boldsymbol{X}_k, \tag{3.2}$$

where $\boldsymbol{X}_k$ is the k-th bit stream of original image, $\boldsymbol{X}_k = (x_{k,1}, x_{k,2}, \ldots, x_{k,n})^T$. Besides, $\boldsymbol{S}_k$ is the syndrome sequence generated from $\boldsymbol{X}_k$, $\boldsymbol{S}_k = (s_{k,1}, x_{k,2}, \ldots, s_{k,m})^T$. In addition, $\mathbf{H}_k$ is the parity-check matrix constructed for the k-th pair of bit streams. More clearly, the length of syndrome sequence m follows the dimension of matrix $\mathbf{H}_k$, which is the practical bit sequence in transmission transformed from original data. The practical coding rate is defined as:

$$R_{X_k} = \frac{m}{n}, \tag{3.3}$$

also called the source coding rate. This rate defined here is the objective to minimize, so as to improve the compression ratio. After being received by the systematic decoder, side information is able to be employed to reconstruct the original data

source.

With the optimized degree distribution for each bit planes, the progressive edge-growth tanner graph could be applied to construct the parity-check matrix **H** [43].

3.2 Degree Distribution Optimization

As explained about the SW encoding in last section, an appropriate degree distribution for LDPC codes is the key to minimize the source coding rate R_X. With regard to each bit plane, once the conditional entropy of the modeled BSC is computed, a degree distribution designed for the corresponding data is capable of being optimized.

The degree distributions of variable nodes and check nodes are defined as (2.5). λ_i and ρi denote the fraction of edges emanating respectively from variable nodes of degree i and from check nodes of degree i. Let a_i and b_i be the fractions of variable and check nodes of degree i. Then the relations between λ_i, ρi and a_i, b_i are derived as (3.4) and (3.5):

$$\begin{aligned} \lambda_i &= \frac{n a_i d_{v,i}}{\sum_{i\geq 2} n a_i d_{v,i}} = \frac{\frac{a_i d_{v,i}}{d_v^{max}}}{\sum_{i\geq 2} a_i d_{v,i}}, \\ \rho_i &= \frac{m b_i d_{c,i}}{\sum_{i\geq 2} m b_i d_{c,i}} = \frac{\frac{b_i d_{c,i}}{d_c^{max}}}{\sum_{i\geq 2} b_i d_{c,i}}. \end{aligned} \tag{3.4}$$

Since (3.4) has shown that $\frac{\lambda_i}{d_{v,i}} = \frac{a_i}{\sum a_i d_{v,i}}$, also $\sum a_i = 1$, then $\sum \frac{\lambda_i}{d_{v,i}} = \sum \frac{a_i}{\sum a_i d_{v,i}} =$

$\frac{1}{\sum a_i d_{v,i}}$. Afterwards, the fraction of nodes are obtained by:

$$\begin{aligned} a_i &= \frac{\lambda_i / d_{v,i}}{\sum_{i \geq 2} \lambda_i / d_{v,i}}, \\ b_i &= \frac{\rho_i / d_{c,i}}{\sum_{i \geq 2} \rho_i / d_{c,i}}. \end{aligned} \tag{3.5}$$

The authors in [44] and [45] claim the optimization objectives and constraints. As presented in [38], the source coding rate is also able to be defined as:

$$r(\lambda, \rho) = \frac{m}{n} = \frac{\sum_{i \geq 2}^{d_c^{max}} \rho_i / d_{c,i}}{\sum_{i \geq 2}^{d_v^{max}} \lambda_i / d_{v,i}}. \tag{3.6}$$

The (3.6) demonstrates the optimization objective that is minimizing the source coding rate $r(\lambda, \rho)$. Obviously, this rate depends on the degree distributions for variable and check nodes. The inequality constraints are requested by EXIT charts scheme. In addition, the linear programming is a reliable technique to implement this optimization for variable nodes and check nodes, respectively. The optimization precess could be stated as the flowchart: Fig. 3.2, in which the optimization is run iteratively with a maximum value N. First of all, initialize the degree distribution of check nodes $\rho(x)$, and the iteration number is $n = 0$ which adds 1 after each iteration. Each iteration

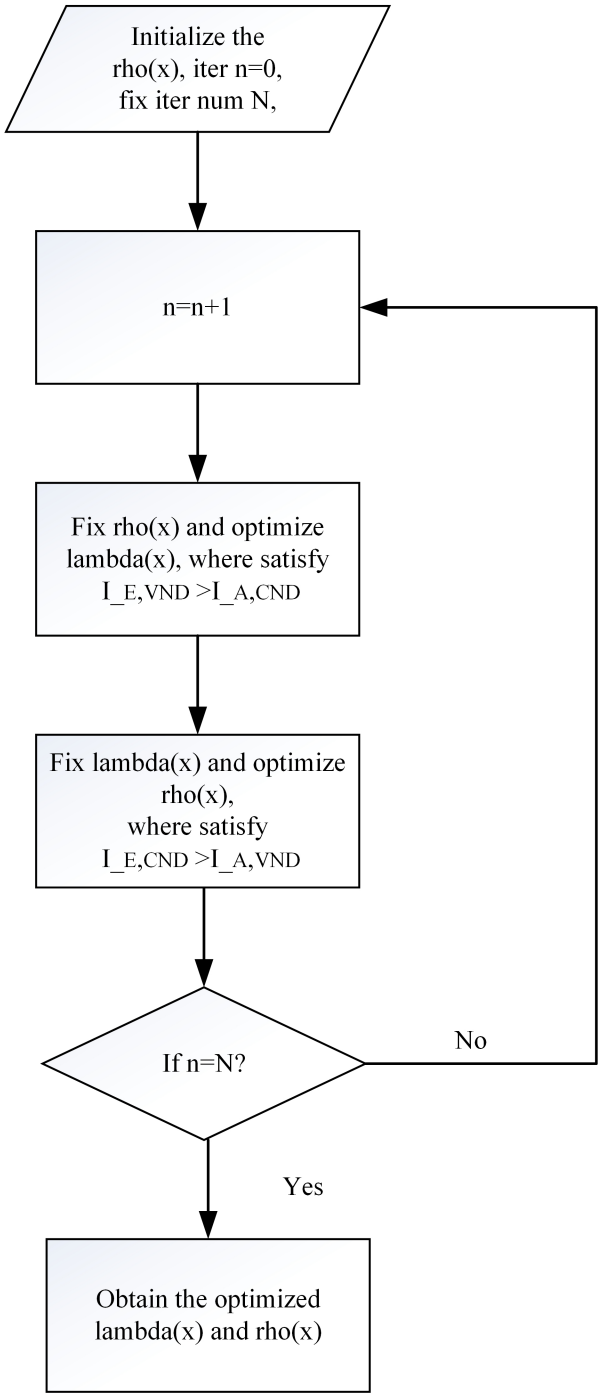


Figure 3.2: The flowchart of degree distribution optimization.

starts from optimizing $\lambda(x)$, and the problem could be formulated as:

$$\begin{aligned} \boldsymbol{\lambda'} = \arg\max_{\lambda} \quad & \sum_{i\geq 2}^{d_v^{max}} \frac{\lambda_i}{d_{v,i}}. \\ s.t. \quad & \sum_{i\geq 2}^{d_v^{max}} \lambda_i I_{E,VND}(i) \geq I_{A,CND}, \\ & 0 \leq \lambda_i \leq 1. \end{aligned} \tag{3.7}$$

Fixing the degree distribution of check nodes $\rho(x)$, the optimization objective is supposed to concentrate on the variables indicated in (3.7). Then with the obtained optimized $\lambda(x)$ in this iteration and fix it, the next process will complete as the following formulation:

$$\begin{aligned} \boldsymbol{\rho'} = \arg\min_{\rho} \quad & \sum_{i\geq 2}^{d_c^{max}} \frac{\rho_i}{d_{c,i}}. \\ s.t. \quad & \sum_{i\geq 2}^{d_c^{max}} \rho_i I_{E,CND}(i) \geq I_{A,VND}, \\ & 0 \leq \rho_i \leq 1. \end{aligned} \tag{3.8}$$

A pair of optimized degree distribution $\boldsymbol{\lambda'}(x)$ and $\boldsymbol{\rho'}(x)$ could be obtained, after the separate steps. Once the iteration number n reaches the setting N, the optimization will stop, then output the pair of degree distribution. Otherwise, it should goto the next round again.

3.3 Compression at Different Spatial Resolutions and An Improved Coding Scheme

The above optimization for degree distribution involves with EXIT charts and linear programming, and it may lead to the expensive computational cost for each pair of data source and side information. Since a light-duty encoder is the proposed codec's feature, how to select the appropriate coding rate for different bit planes with good compression ratio and high reconstruction fidelity is worth researching.

According to the statistics of conditional entropy for each bit plane, some appropriate source coding rates with degree distributions obtained in [46] are able to be applied in the SW coder directly. More precisely, to facilitate the selection of source coding rate with its corresponding optimized degree distributions, those 38 optimized degree distributions presented in [26] are able to be selected in different situations. Also, they are demonstrated in Appendix. Via the statistics of crossover probabilities in modeled BSC, a source coding rate is able to be chosen from those 38 ones to encode and decode the original data source without error for each bit plane. Moreover, we hope the selected source coding rate is at a lowest rate (when the gap between the conditional entropy and coding rate is smallest). In future work, an approximate rate estimation for each bit plane is ought to be drawn for a HFV dataset. Those HFV video signals are supposed to share some common properties, leading to the concentration of conditional entropy of its corresponding channel model. In addition, the compression at different spatial resolution and an improved coding scheme which exploits the dependency between two consecutive bit planes are developed as below.

3.3.1 Compression at the Original Spatial Resolution

The high frame rate video usually has high spatial resolution, containing abundant and complicated image details. Compressing the video frames at different scale of resolution is capable of realizing different compression requirements, such as the reconstruction image fidelity and the compression ratio.

In general, compression is implemented at the original image resolution, and it may achieve higher reconstruction fidelity. The developed SW coder has been presented in Fig. 3.1, where the data source and recovered side information have been transformed into the bit streams. Furthermore, the bit streams split into 8 pairs, each of them corresponding to the assembled bit sequences in a certain bit position, including $\boldsymbol{X}_k$ and $\boldsymbol{Y}_k$, and they are also supposed to be the k-th bit planes. Then there would be a BSC model being constructed to describe the correlation between each pair of bit streams (planes). The SI frame is predicted by motion estimation and motion compensation, and it is only known at the collaborative decoder. When it satisfies the inequality in (3.1), the SW decoder could recover $\hat{\boldsymbol{X}}_k$ without error probability when the length of sequence is sufficiently long. However, consider one certain video, the resolution of each frame is fixed, and the length of transformed bit streams is limited. It's obvious that it is not able to tend to infinity. Take the gap between conditional entropy and coding rate into consideration, we will increase the selected source coding rate to achieve the perfect decoding. Therefore, based on the statistical information of modeled BSC, an approximate selection of the source coding rate with its degree distribution is supposed to be selected from those 38 ones in [26]. This way can greatly reduce the computational cost in optimization for a better degree distribution. Note that, the decoder is capable of transmitting a feedback about the

predicted side information to the encoder, which may lead to the model construction between the data source and the SI.

3.3.2 Compression at the Reduced Spatial Resolution

Implementing the compression at the reduced spatial resolution usually achieves a higher compression ratio. Besides the selection of an approximate source coding rate, downsampling is also an effective scheme to reduce the bit datas coded for the image in transmission and storage.

A classic and simple compression-by-downsampling plan may contain a downsampler as an image encoder and an effective upsampler as its decoder, which is regarded as a lossy image compression method. With respect to HFV image signals, an appropriate downsampling method such as row/column binning and skipping techniques is applied during the acquisition of raw video data. Then the corresponding decoder has ability to reconstruct the original image by upsampling the received data. Usually, 'box' ,'bilinear' and 'bicubic' interpolations are the effective upsampling methods to be chosen. As the technology for image reconstrution develops, super-resolution (SR) also becomes a feasible scheme to decode the downsampled image/frame at high-fidelity, too.

Specially, bicubic interpolation is an extension of cubic interpolation for interpolating data points on a two dimensional regular grid. Observing the bicubic interpolation, the recovery may be smoother than ones reconstructed by nearest-neighbor or bilinear interpolation. In addition, bicubic interpolation can be implemented with either Lagrange polynomials, cubic splines, or cubic convolution algorithm.

In the practical implementation of compression at reduced resolution, the target

current frame is downsampled by a scale factor of Q before SW encoding. Suppose $Q = 2$, the original image is downsampled by the factor 2 in the horizontal and vertical directions. In consequence, a two-dimensional image signal will be reduced to 1/4 of original data size. As a result of such resizing, the total transmitted video data is able to be largely cut down. In this situation, at the collaborative decoder, the side information is also the downsampled image predicted by ME and MC. The main difference from the compression at original resolution case is that both data source and side information here are the downsampled image. Since some important image details may be lost during the resolution resizing, then some other scheme is necessary to be developed to improve the fidelity of the upsampled frame.

3.3.3 An Improved Coding Scheme

No matter what spatial resolution the image is, a generalized and efficient scheme is developed to reduce the source coding rates successfully . The previous two situations are studying the statistics on the conditional entropy for each bit plane, which is independent from other bit planes. However, the spatial correlation between two consecutive bit planes is able to be explored. Therefore, this improved coding scheme is proposed to reduce the spatial redundancy between consecutive bit planes.

According to the observation on the relevance of successive bit planes, we find that when the bit pairs $x_{k-1,i} \neq y_{k-1,i}$ satisfies in previous (k-1)-th bit plane, the probability $\Pr(x_{k,i} \neq y_{k,i})$ in k-th bit plane is much higher than the probability when $x_{k-1,i} = y_{k-1,i}$ holds. Therefore, we exploit the spatial dependency between the successive bit planes, and then construct two BSC models, respectively, for the situation when $X_{k-1} = Y_{k-1}$ and the situation when $X_{k-1} \neq Y_{k-1}$ illustrated in

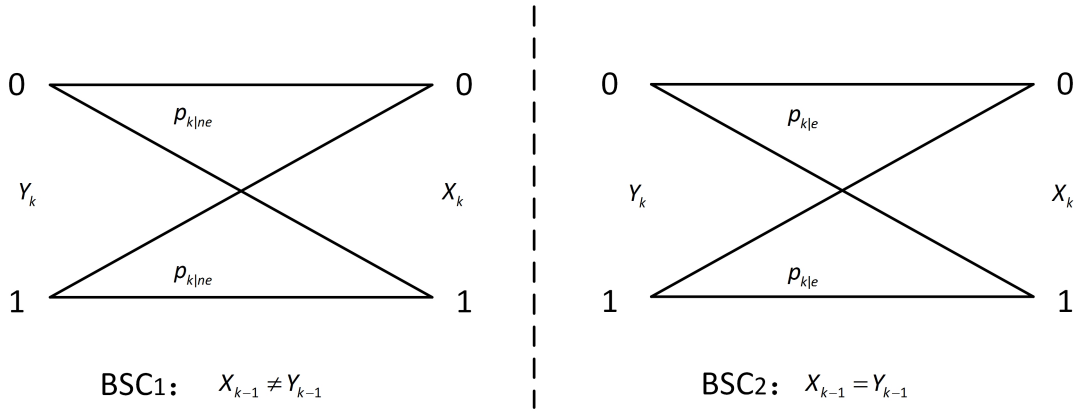


Figure 3.3: Two separate BSC models are constructed for the improved coding scheme which exploits the dependency between two consecutive bit planes.

Fig. 3.3. Note that, X_{k-1} and Y_{k-1} are the random variables, repsectively in the bit sequences $\boldsymbol{X}_{k-1}$ and $\boldsymbol{Y}_{k-1}$. In each one of two BSC models, there is a statistical crossover probability being detected, and then the conditional entropy for the k-th bit plane is able to be calculated by averaging the conditional entropies for those two cases. The computation of the average conditional entropy is defined as:

$$\begin{aligned} H(X_k|Y_k, U_k) = p_{k-1} \cdot H(X_k|Y_k, 0) \\ + (1 - p_{k-1}) \cdot H(X_k|Y_k, 1), \end{aligned} \tag{3.9}$$

and

$$U_k = \begin{cases} 0, & \text{if } X_{k-1} \neq Y_{k-1}, \\ 1, & \text{if } X_{k-1} = Y_{k-1}. \end{cases} \tag{3.10}$$

When $X_{k-1} \neq Y_{k-1}$, then $H(X_k|Y_k, 0)$ is the conditional entropy of the BSC model with crossover probability $p_{k|ne}$ for the k-th bit plane . Similarly, when $X_{k-1} = Y_{k-1}$, $H(X_k|Y_k, 1)$ is the condition entropy of the BSC model with crossover probability $p_{k|e}$ for k-th bit plane . Besides, p_{k-1} is the crossover probability of the BSC model for

(k-1)-th bit plane, when the improved coding scheme is not employed.

Then, after obtaining the statistical crossover probability (called CPE) when $X_{k-1} = Y_{k-1}$ and the statistical crossover probability (called CPNE) when $X_{k-1} \neq Y_{k-1}$, the averaged conditional entropy for each bit plane (except the 1^{st} plane) will be obtained via the computation (3.9). Finally, an appropriate selection of source coding rate is determined later. Then, the experimental results in chapter 5 will prove the validity of the proposed optimized encoding scheme.

The flexibility at different image resolution facilitates the system to achieve different goals on video compression. The trade-off between reconstruction frame fidelity and frame compression ratio should be taken into consideration. The improced coding scheme is capable of reducing the bits coded during the compression.

Chapter 4

Heavy-duty High-Fidelity Decoder

To realize the achievable high write speed from the high-speed camera sensor to the massive storage device, the light-duty encoder has transfered the burden of reconstructing the original frame to the collaborative decoder. At the receiver side, a relatively heavy-duty decoder is designed to obtain higher fidelity and better compression ratio. Motion estimation and compensation are employed twice at the heavy-duty decoder. Firstly, the SI frame is predicted by this technique. Then we implement the SW decoding with the received syndrome sequences and predicted SI images. With respect to the encoding developed based on the downsampled image, then a upsampler will utilize traditional upsampling methods such as 'box', 'bicubic' or SR. Last but not least, some important image details included in pixels or blocks of upsampled frame (which has been recovered to its original spatial resolution) need to be modified by ME and MC again. The diagram of implementing the developed heavy-duty decoder is illustrated as the Fig. 4.1.

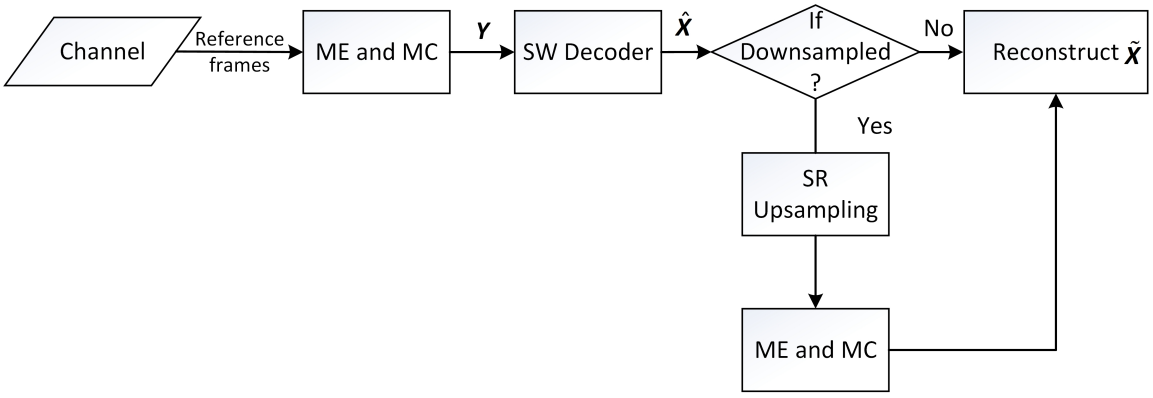


Figure 4.1: The heavy-duty decoder implementation

4.1 Side Information Reconstruction by First Motion Estimation and Compensation

Corresponding to the light-duty SW encoder, the proposed SW decoder implement motion estimation and motion compensation solely first, in order to predict the side informationm, which is illustrated in Fig. 4.2, which does not take the other decoding techniques into accounts.

Motion estimation exploits the temporal and spatial dependency between the current frames and reference frames to complete the video compression. By conventional or modified ME techniques, the video is able to be processed along the motion trajectories. The difference between motion estimation and motion compensation is that, the ME perceives the displacement of objects in the current frame compared with the reference frames. By subtracting the current blocks from the estimated blocks, the error with less data size will be obtained. Then, the MC could utilize the errors and movement vectors between frames to represent the current frame and then recover it in the receiver. Hence, the ME is the critical technique in the inter-frame coding.

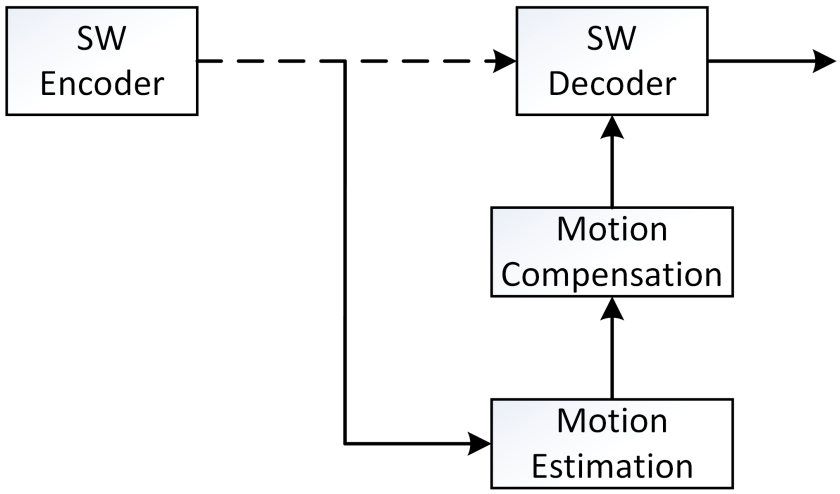


Figure 4.2: The SW coder with ME and MC in decoder

With the estimated motion vectors, the current/target frame could be reconstructed by one or more reference frames. When there is high correlation between the current frame and the reference frames , one could only code the difference between the consecutive frames, leading to a less amount of data size in transmission and storage.

Since there may be different changes among frames in a real video signal, such as a complex combination of illumination changes and movements, we mainly focus on the motions such as the translation and rotation. Such motion is difficult to be estimated, and it may require large amounts during processing. The classic motion estimation techniques include block-matching algorithm (BMA), optical flow, phase correlation and frequency domain methods, and pixel recursive algorithm. However, the pixel-based techniques involve more computational complexity and less regularity, and they are difficult to realize in hardware.

Block-matching algorithm is easy and successful to be applied in the motion estimation when there is translation motion or the motion is slow between current frame and reference frame. In this thesis, high frame rate video is capable of satisfying

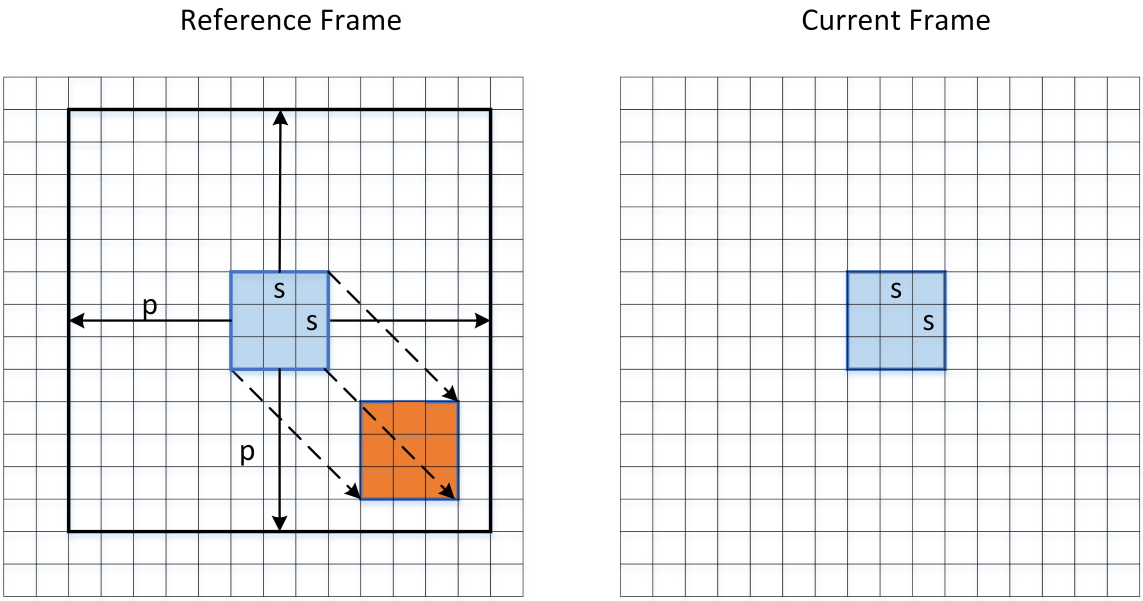


Figure 4.3: Block matching with the macro block size $s \times s$ pixels and a search parameter p.

that the slow and translation motion between near frame. Also, occlusion of one object by another, and uncovered background can be neglected. Consequently, the features - regularity and simplicity in block-matching is more suitable for software and hardware implementation.

BMA is a block-based estimation and compensation technique. The conventional BMA compress the current frame by referring to the reference frame. Illustrated in the Fig. 4.3, the estimation basis is a group of $s \times s$ pixels, called macro block. So one frame may be divided into a sequence of macro blocks, which are compared with the blocks in the previous or subsequent reference frames. The search region in the reference frames is a square window centered on the collocated position of the block in the current frame. Basically, the search window is extended to p pixels on all four sides, constituting a larger region with a size $(s+2p) \times (s+2p)$. The p represents the

searching parameter, and the larger p is, the more computational cost it produces.

The motion estimation is the process to find the best matching block and the motion vector based on a certain criterion. More precisely, one could apply different search algorithms to find the matches. Exhaustive search is the fundamental one, besides that, three-step, new three-step, four-step, and diamond search could also be utilized to improve the efficiency. Those other sub-optimal efficient algorithms must keep a trade-off between the computational efficiency and the estimation accuracy. Although ES is the most expensive one in computational cost, it is still being chosen as the most reliable to find the best match with a better quality.

There are several common cost functions such as Mean Absolute Difference (MAD), Mean Squared Error (MSE), Root MSE (RMSE) and so on. In general, the macro block in the reference frame which is found as the best match of the current block leads to the least cost. Since the conventional BMA subtract the best matching block from the current block, it then obtains the error signal and its motion vector. Consequently, the better its estimate, the smaller error signals are, resulting in less transmission bit rate.

In our developed decoder, ME and MC are implemented to reconstruct the SI image in current frame. The original image is not known at decoder, hence the previous and subsequent reference frames are applied to predict the image as the side information. Assuming that there are one previous frame f_{c-t} and one subsequent reference frame f_{c+t} used in the estimation and compensation for the current frame f_c, t is frame gap between them. Also, MAD is chosen to be the cost function to find

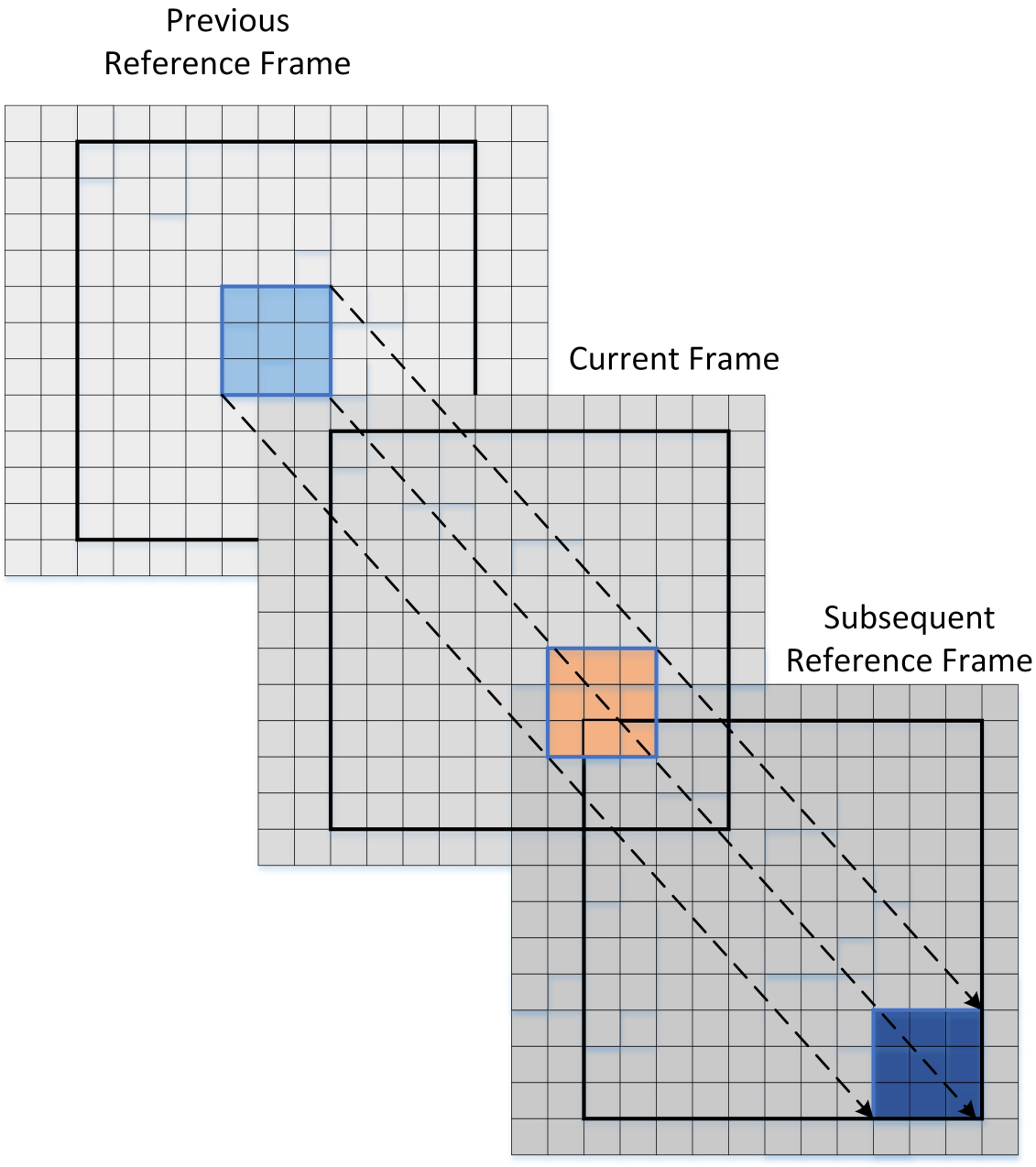


Figure 4.4: Motion estimation and compensation for the current frame with one previous and one subsequent reference frames.

the matching macro block. MAD is defined as:

$$MAD = \frac{1}{s^2} \sum_{i=0}^{s-1} \sum_{j=0}^{s-1} |I_{c-t}(i,j) - I_{c+t}(i,j)|, \tag{4.1}$$

where $I_{c-t}(i,j)$ and $I_{c+t}(i,j)$ are the RGB pixel values in macro blocks of the previous and subsequent frame, respectively. The block pair resulting in the least cost is the best match. Afterwards, a motion vector (dx, dy) is able to be obtained. The Fig. 4.4 assumes the movement trajectories along a translation direction among the three frames. Consequently, the predicted block in frame f_c is supposed to locate at the center of the motion trajectory. In other words, the motion vector between frame f_{c-t} and f_c is $(dx/2, dy/2)$. Furthermore, the pixels value in the reconstructed current block are averaged by:

$$\begin{gathered} I_c(i,j) = \frac{I_{c-t}(i,j) + I_{c+t}(i,j)}{2}, \\ i,j \in \{0,1,\ldots,s-1\}. \end{gathered} \tag{4.2}$$

By this means, the current frame is able to be recovered by ME and MC at different spatial resolutions.

4.2 Slepian-Wolf Decoding

With the side information recovered by the first motion estimation and motion compensation, the collaborative decoder is going to decode the received information - syndrome sequences to its target spatial resolution.

Similar to the definition of traditional Slepian-Wolf decoding method, the SW decoder in our proposed codec exploits the spatial correlation between adjacent bit

planes to improve the compression ratio. This improvement corresponds to the improced coding scheme.

With regard to the k-th bit plane decoding, it also starts from the variables passing messages to check nodes. Suppose the length of the syndrome sequence $\boldsymbol{S}_k$ is m. Firstly, the initial message is computed from the observed value of side information only. Therefore, referring to [36], at 0-th iteration, the initial messages emanating from variable node v_i to each of adjacent check nodes are redefined as:

$$M_{k,i}^{0} = \begin{cases} (1-2y_{k,i})log_2\dfrac{1-p_k}{p_k}, & \text{if } k=1, \\ (1-2y_{k,i})log_2\dfrac{1-p_{k|e}}{p_{k|e}}, & \text{if } k\neq 1,\ x_{k-1,i}=y_{k-1,i}, \\ (1-2y_{k,i})log_2\dfrac{1-p_{k|ne}}{p_{k|ne}}, & \text{if } k\neq 1,\ x_{k-1,i}\neq y_{k-1,i}, \end{cases} \tag{4.3}$$

It's obvious that the spatial dependency has been exploited except when $k=1$ for the 1^{st} bit plane (because there is no previous bit plane before it). In such sole case, p_k is the crossover probability of modeled BSC without the improved coding scheme. While $k\neq 1$, the statistical crossover probability will be separated to two cases. They are defined as $p_{k|e}$ when $X_{k-1}=Y_{k-1}$, and the $p_{k|ne}$ when $X_{k-1}\neq Y_{k-1}$. In the practical image coding, $x_{k-1,i}$ is the decoded bit for the (k-1)-th bit plane, which should be the same as original bit data when there is no error probability after SW decoding.

In each iteration, $M_{v_{k,i}\to c_{k,j}}^{(l)}$ denotes the message passing from variable node $v_{k,i}$ to the adjacent check node $c_{k,j}$. It is obtained as below:

$$M_{v_{k,i}\to c_{k,j}}^{(l)} = M_{k,i}^{0} + \sum_{c_{k,r}\in C_{v_{k,i}}\backslash\{c_{k,j}\}} N_{c_{k,r}\to v_{k,i}}^{(l)}. \tag{4.4}$$

In above equation, $v_{k,i}$ is in the variable nodes set V_k in k-th plane. $c_{k,j}$ composes check nodes set $\{c_{k,1}, c_{k,2}, \ldots, c_{k,m}\}$, representing the syndrome nodes $\boldsymbol{S}_k$. Moreover, $C_{v_{k,i}}$ is the set of all check nodes connected to variable $v_{k,i}$, and $c_{k,r}$ is the adjacent check node in set $C_{v_{k,i}}$ but except for $c_{k,j}$ itself.

To simplify the computation of the messages passing from check nodes to variable nodes, (2.8) could be redefined as the following equation, denoting the message emanating from $c_{k,i}$ to the connected variable $v_{k,j}$. Its explanation refers to (2.8).

$$N^{(l)}_{c_{k,i}\to v_{k,j}} = 2\ arctanh\{(1-2s_{k,i}) \prod_{v_{k,r}\in V_{c_{k,i}}\setminus v_{k,j}} tanh(\frac{M^{(l-1)}_{v_{k,r}\to c_{k,i}}}{2})\}. \tag{4.5}$$

Similar to (2.9), after each iteration completes, it is able to generate a recovered value $\hat{x}_{k,i}$ according to the judgment:

$$\hat{x}_{k,i} = \begin{cases} 0, & \text{if} \quad M^0_{k,i} + \sum_{c_{k,r}\in C_{v_{k,i}}} N^{(l)}_{c_{k,r}\to v_{k,i}} \geq 0, \\ 1, & \text{if} \quad M^0_{k,i} + \sum_{c_{k,r}\in C_{v_{k,i}}} N^{(l)}_{c_{k,r}\to v_{k,i}} < 0. \end{cases} \tag{4.6}$$

After the iteration l reaches a certain number L, the belief propagation will stop and then obtain the decoded bit sequences $\hat{\boldsymbol{X}}_k = \{\hat{x}_{k,1}, \hat{x}_{k,2}, \ldots, \hat{x}_{k,n}\}$ for the k-th bit plane. If the coding is implemented based at the original spatial resolution, then the $\hat{\boldsymbol{X}}_k$ recovered by SW decoding and the remaining bit planes may estimated by ME and MC constitute a reconstruction frame (here we need to retransform the reconstructed bit streams to RGB image $\hat{\boldsymbol{X}}$). In contrast with the compression at original resolution, if the coding is employed based on the downsampled frame, then $\hat{\boldsymbol{X}}$ after predicting and SW decoding are necessary to be upsampled to its original

resolution first.

4.3 Fast Super-Resolution

With respect to the downsampled version image, a certain technique is supposed to be applied to upsample the SW decoded image to its original spatial resolution. The compressions-by-downsampling method has a combination of downsampler as the encoder and a corresponding upsampler as the collaborative decoder. In this thesis, a reliable upsampling technique is employed to reconstruct the image after SW decoding. Conventional upsampling techniques contain the interpolations such as 'nearest', 'box', and 'bilinear' and 'bicubic'. Also, the advanced upsampling techniques like super-resolution (SR) is able to improve the recovery fidelity as well. More precisely, the state-of-the-art learning based SR technique: $A+$ is an effective and efficient method compared with other methods [47] and [48].

Comparison between the traditional upsampling interpolations and the advanced SR are completed based on the sample frames of experimental video. Downsample one extracted frame image firstly via 'bicubic', then at the decoder apply 'box', 'bilinear', 'bicubic' interpolations and SR respectively to upsample the downsampled version to its original resolution. Specially, in order to achieve higher upsampling fidelity, the advanced learning based SR: $A+$ is applied here to upsample the SW decoded image at reduced resolution to its original spatial resolution. In Table. 4.1, the reconstruction PSNRs by four upsampling schemes are presented as follows. It's evident that the SR: $A+$ is capable of achieving higher reconstruction image fidelity than the traditional interpolations such as 'box', 'bilinear', 'bicubic'.

Table 4.1: PSNR with comparing different upsampling techniques

Picture	'box'	'bilinear'	'bicubic'	SR: $A+$
No.1	29.57dB	29.49dB	30.26dB	30.62dB
No.2	30.19dB	30.34dB	31.41dB	32.42dB
No.3	31.49dB	32.07dB	33.26dB	34.78dB
No.4	36.77dB	39.51dB	42.38dB	43.62dB
No.5	35.65dB	39.04dB	42.04dB	43.38dB

4.4 Modification by Second Motion Estimation and Compensation

After the SW decoding and upsampling by super-resolution, some important image details may have been lost because some high frequency information are thrown away during the image resolution resizing. Hence, it is able to exploit the spatial correlation between the consecutive frames by motion estimation and compensation the second time, as the supplementary information.

The main process of the modification is the same as the scheme stated in the first utilization of ME and MC. Fix one macro block in the reconstructed current frame, and then searching is capable of being implemented in a $(s + 2p) \times (s + 2p)$ square window in reference frames, centered on the collocated position of the current block. To gain the relatively high quality of the modified image, exhaustive search is applied to find the best matches rather than the other sub-optimal efficient algorithms. The best matching block pair should be within the square region in the previous

and subsequent frame respectively, resulting in the least summation of MAD costs between f_{c-t}, f_c and f_c, f_{c+t}. More importantly, the three macro blocks should be moving along a straight trajectory, where the current block is located at the center.

Afterwards, the modification is able to be resolved by the following comparison. After SW decoding and upsampling, the reconstructed current frame is $\bar{\boldsymbol{X}}$. Then we compare the current macro block with the estimated matching block pair in reference frames. First, obtain the maximal and minimal pixel value of the reference block pair.

$$\begin{aligned} Max_{ij} &= \max\{I'_{c-t}(i,j), I'_{c+t}(i,j)\}, \\ Min_{ij} &= \min\{I'_{c-t}(i,j), I'_{c+t}(i,j)\}, \end{aligned} \tag{4.7}$$

in which $I'_{c-t}(i,j)$ and $I'_{c+t}(i,j)$ are the pixel values in the matching block pair, $i,j \in \{0,1,\ldots,s-1\}$. Afterwards, compare the pixel values $\bar{I}_c(i,j)$ in reconstructed current block, then the modification to the reconstructed current macro block is processed as below:

$$\tilde{I}_c(i,j) = \begin{cases} Max_{ij}, & \text{if } \bar{I}_c(i,j) > Max_{ij}, \\ Min_{ij}, & \text{if } \bar{I}_c(i,j) < Min_{ij}, \\ \bar{I}_c(i,j), & \text{if } Min_{ij} < \bar{I}_c(i,j) < Max_{ij}. \end{cases} \tag{4.8}$$

The modification to upsampled frame is presented as the developed comparison. Consequently, the upsampled pixel value in current frame is able to be modified, and finally an improved reconstruction of original frame is obtained as $\tilde{\boldsymbol{X}}$.

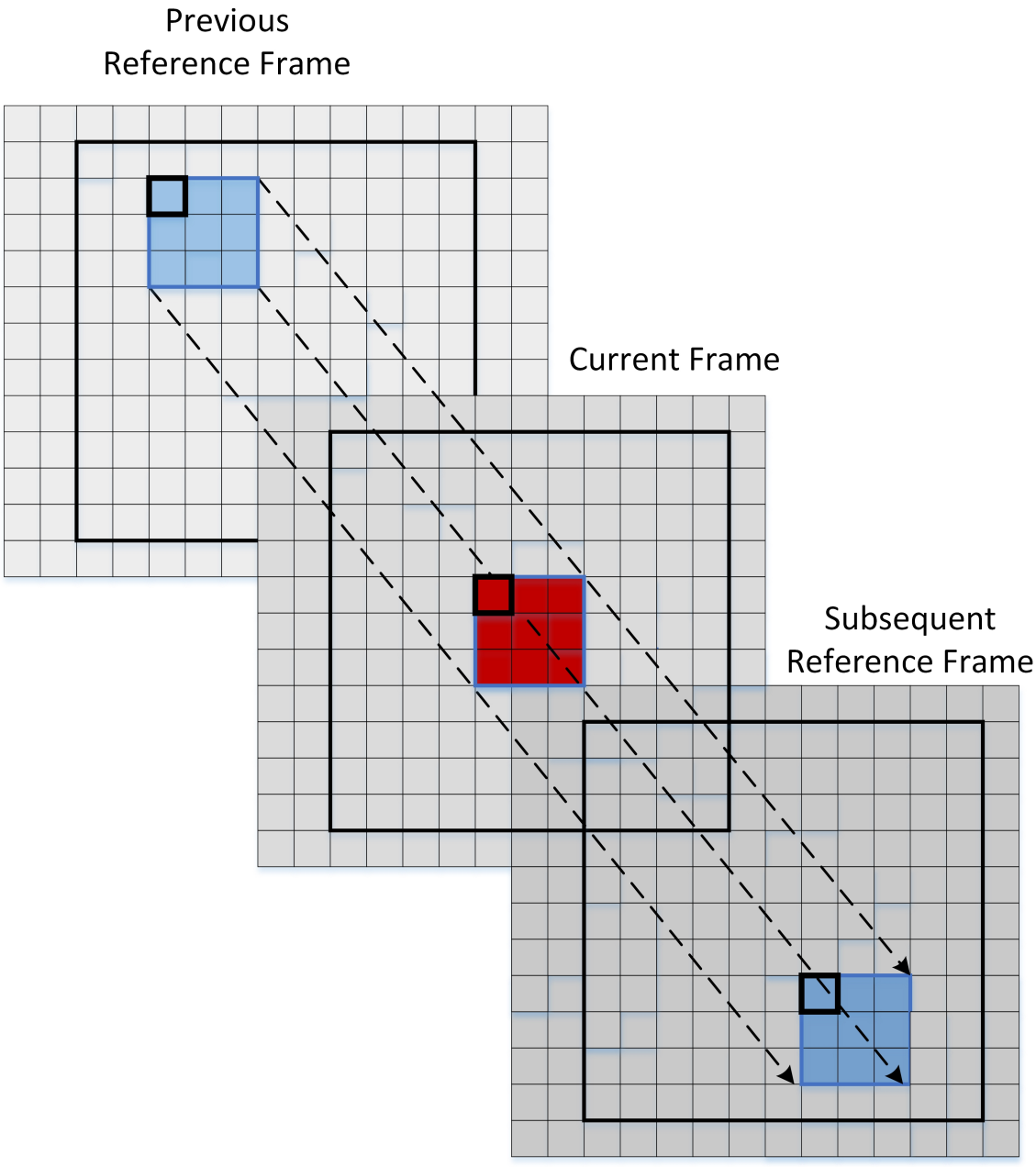


Figure 4.5: Modification by motion estimation and motion compensation to the up-sampled frame.

Chapter 5

Simulation and Results

This chapter mainly presents the experimental results of the simulation on our proposed compression techniques. There are two separate experiments. The first experiment is about the validation of the modification via ME and MC. With respect to the experiment on downsampled images, the performance of the modification via motion estimation and compensation after upsampling has been assessed. Also, the compared compression-by-downsampling method is selected. In addition, the second experiment presents the results of the entire codec with proposed compression techniques. The experiment 2 separates into two situations, compression at original spatial resolution and compression at reduced spatial resolution. Besides, the performance of the improved coding scheme which exploits the spatial dependency between two consecutive bit planes, is also analyzed.

In the experiment on implementing the BP algorithm, the sparsity of the parity-check matrix is fully exploited. When generating the sparse matrix, only the indexes of those connected variable(check) nodes are recorded. Otherwise, with regard to a LDPC code with dimension e.g. $(100000, 50000)$, if we record the value of each entry

in integer type, it is necessary to generate a matrix $\mathbf{H}$ occupying the stack memory of $100000 \times 50000 \div 1024^3 \approx 4.7GB$. By exploiting the sparsity property of LDPC codes, the effective and efficient method can greatly reduce the overheads in stack memory.

5.1 Experiment 1

First of all, a common compression-by-downsampling method is supposed to be chosen to compare with our developed compression techniques. To facilitate the transmission from high speed camera sensor to massive data storage device, a fast encoder with low computation cost is the solution to remove the obstacle for the high throughput. Hence, conventional idea to compress the extracted frame/image here is stated as follows. Downsample the image first via common DS interpolation techniques such as 'box', 'cubic', 'bicubic' and etc. Then decompressor upsample the received image to its original spatial resolution. In addition, the advanced upsampling techniques such as super-resolution may replace interpolations. In the experiment, the state-of-the-art learning based SR method: $A+$ is utilized to upsample the downsampled image. All of them are implemented with a resizing scale factor of $Q = 2$. Note that, those resizing schemes do not involve the information of reference frames.

Then Table. 5.1 shows the reconstruction performance - Peak Signal Noise to Ratio (PSNR) by different combinations of DS and US techniques.

Experiment is completed based on the HFV video source 'streetscape' which is captured using a sony 4K movie camera with 60Hz fps . According to the experimental results $PSNR_1$ which denotes the reconstruction fidelity by DS and US without other information, it is evident that $A+$ outperforms the other upsampling interpolations

Table 5.1: Simulation results on experiment 1

Downsample	Upsample	$PSNR_1$	Modify via ME/MC $PSNR_2$
'box'	'box'	31.50dB	35.51dB
'box'	'bicubic'	33.21dB	36.46dB
'bicubic'	'bicubic'	33.04dB	36.44dB
'box'	$A+$	34.80dB	37.32dB
'bicubic'	$A+$	35.50dB	37.70dB

such as 'box', 'bicubic'. Moreover, the combination of downsampling via 'bicubic' interpolation and upsampling via $A+$ has the highest reconstruction PSNR than other ones. Consequently, we choose the best compression-by-downsampling combination: 'bicubic' and $A+$ as the downsampling and upsampling techniques to be one of the compared methods. The value $PSNR = 35.50dB$ is the reconstruction PSNR value of the selected combination, which will be compared with our proposed techniques later.

Besides, another compared method called DSU is an embedded compression scheme proposed in [49]. It put forwards a new, simple infra-frame embedded technique based on downsampling and side-information aided upsampling. This method is able to When we set the compression ratio that is $16 : 6$, the $PSNR = 35.44dB$.

Furthermore, we assess the performance of the modification via motion estimation and compensation after upsampling, shown as $PSNR_2$. Obviously, the modification to the upsampled image with reference frames greatly promotes the reconstruction fidelity. The PSNR of reconstruction reaches a relatively high value $PSNR = 37.70dB$

which is higher than the compared combination in $PSNR_1$ and the image fidelity by DSU. Based on the results of $PSNR_2$, the modification via ME and MC is able to be applied in our developed codec, leading to the recovery of some important image details.

5.2 Experiment 2

The experiment 2 mainly analyzes the performance of the designed codec with the proposed compression techniques. As illustrated in Fig. 4.1, there are two different situations (compression at original resolution and reduced resolution) to verify the proposed schemes. They have different advantages in compression ratio and reconstruction fidelity. And the trade-off between them should be considered to satisfy different compression requirements.

5.2.1 The experimental results of compression at the original resolution

The first case is that implementing the compression based on the original spatial resolution. The same HFV 'streetscape' as the experimental video source have been extracted to a sequence of frames by 60Hz fps.

Since side information $\boldsymbol{Y}_k$ is predicted by the previous and subsequent frames, the dependency between the current frame and recovered SI frame may rely on the frame rate and the gap between the reference and current frame. For instance, we suppose that the current target image is the m-th frame, the frame gap is t, then the previous and subsequent reference images applied to ME are the (m-t)-th and

Table 5.2: Compression at original resolution, the statistics of crossover probability and conditional entropy of modeled BSC and a selection of source coding rate, frame gap $t = 5$.

Bit Plane	Crossover Probability	Conditional Entropy	Coding Rate
1^{st}	0.0203	0.1430	0.2000
2^{nd}	0.0639	0.3427	0.4010
3^{rd}	0.1053	0.4854	0.5765
4^{th}	0.1839	0.6885	0.7800
5^{th}	0.2668	0.8368	0.9300
6^{th}	0.3421	0.9268	$\times$
7^{th}	0.4044	0.9734	$\times$
8^{th}	0.4549	0.0041	$\times$

(m+t)-th frame, respectively. The common sense is that the smaller the frame gap , the higher the correlation between $\boldsymbol{X}_k$ and $\boldsymbol{Y}_k$ supposed to be. Note that, the compression system allows the decoder to send a feedback about the predicted SI to the encoder. Furthermore, there is a statistically modeled BSC to describe the dependency between the k-th pair of bit streams (planes). Then a statistical crossover probability and corresponding conditional entropy is capable of being computed by (3.1) for the BSC model. Consequently, a selection of source coding rate is able to be obtained by satisfying the inequality in (3.1). In consideration of the frame gap, a certain value $t = 5$ is setting in our experiment.

In Table. 5.2, the crossover probability and conditional entropy increases as bit plane shift from the most significant bit to the least significant bit. So as to have Slepian-Wolf coder decode the bit streams without error probability, an appropriate

source coding rate at lowest rate is able to be selected from those 38 optimized ones in Appendix of this thesis. Accordingly, the Table also shows the selections. More importantly, with regard to the compression at the original resolution, there are only the first 5 bits need be SW encoded and decoded. That's because we find that the more bits the SW coder compresses, the higher the PSNR and the lower the compression ratio. And when there are only the first 5 bits being SW coded, both compression ratio and image fidelity reaches relatively high values. Besides, the remaining bit planes (which do not involve the SW coding) can keep the predicted values by the first motion estimation and compensation. Consequently, when only the first 5 bit planes are SW coded, the reconstruction frame fidelity and compression ratio are obtained as follows.

$$PSNR = 39.31dB, \quad Bpp = 2.8875.$$

That is $3.81dB$ higher than the compared compression-by-downsampling and $3.87dB$ higher than DSU. And the Bit per Pixel (Bpp) demonstrates how many bits supposed to be used in coding, representing the compression ratio.

Besides, the improved coding scheme is applied to exploit the spatial dependency between the consecutive bit planes. Then construct two separate statistical BSC models respectively for the situation when $X_{k-1} = Y_{k-1}$ and the situation when $X_{k-1} \neq Y_{k-1}$. More precisely, the crossover probability of the BSC model when $X_{k-1} = Y_{k-1}$ is statistically calculated as CPE, and the crossover probability when $X_{k-1} \neq Y_{k-1}$ is statistically obtained as CPNE. With those two statistical CPs, the corresponding conditional entropy for the k-th bit plane is able to be computaed by averaging the entropies for those two cases (3.9).

Table 5.3: The improved coding scheme is applied to the original image , the statistics of crossover probabilities of two separate BSC models and averaged conditional entropy and a selection of source coding rate, frame gap $t = 5$.

Bit Plane	CPE	CPNE	Average CE	Coding rate
2^{nd}	0.0450	0.9789	0.2622	0.3277
3^{rd}	0.0474	0.9529	0.2753	0.3500
4^{th}	0.1046	0.8581	0.4945	0.5765
5^{th}	0.1652	0.7175	0.6856	0.7800
6^{th}	0.2313	0.6468	0.8220	$\times$
7^{th}	0.3063	0.5930	0.9182	$\times$
8^{th}	0.3828	0.5611	0.9718	$\times$

Table. 5.3 has presented the CPE and CPNE for two different BSC models. The conditional entropy for the 1^{st} bit plane and the p_{k-1} for (k-1)-th bit plane are the same as the information in Table. 5.2. According to the statistics of this Table, the averaged conditional entropy shows that the source coding rates is obviously less than the previous method without the improved coding scheme, resulting in the decrease of source coding rates effectively. At last, the reconstruction image fidelity and compression ratio are:

$$PSNR = 39.31dB, \quad Bpp = 2.2342.$$

Note that, the PSNR remains the same value. Therefore, the improved coding scheme primarily reduces the source coding rate, while keeping the same reconstruction frame fidelity.

5.2.2 The experimental results of compression at the reduced resolution

Another case is when compression is based on the downsampled spatial image. Downsample the original data source with a scale factor of $Q = 2$. By downsampling, the Bpp is able to be dramatically reduced, leading to a much higher compression ratio.

Similar to the compression at the original spatial resolution, the primary idea of SW encoder and SW decoder is the same as the previous one. The main difference is that the state-of-the-art learning based SR technique is employed after the SW decoding, in order to recover its original scale. Since there are some important image details being lost during SW coding and resolution scaling, a modification by ME and MC is developed the second time to improve the fidelity of the upsampled image. Comply with the setting in the first case , frame gap here is also setting as $t = 5$ in our experiment.

In Table. 5.4, the crossover probability and conditional entropy are indicated. Because SW coding is implemented on the downsampled image, the Bpp has been greatly reduced. There may be one more bit plane being SW coded, so as to achieve a higher reconstruction image fidelity. So the discussion about two selections that coding the first 5 bit planes and coding the first 6 bit planes makes sense. The trade-off between the PSNR and Bpp is necessary to be balanced. Finally, the reconstruction frame fidelity and compression ratio are obtained as

$$Coding\ on\ first\ 5\ bit : PSNR = 38.42dB, \quad Bpp = 0.6146.$$
$$Coding\ on\ first\ 6\ bit : PSNR = 38.98dB, \quad Bpp = 0.8646.$$

Table 5.4: Compression at reduced resolution, the statistics of crossover probability and conditional entropy of modeled BSC and a selection of source coding rate, frame gap $t = 5$.

Bit Plane	Crossover Probability	Conditional Entropy	Coding Rate
1^{st}	0.0118	0.0927	0.1500
2^{nd}	0.0442	0.2612	0.3277
3^{rd}	0.0748	0.3836	0.4508
4^{th}	0.1377	0.5782	0.6500
5^{th}	0.2255	0.7702	0.8800
6^{th}	0.3050	0.8873	1.0000
7^{th}	0.3776	0.9563	$\times$
8^{th}	0.4381	0.9889	$\times$

Coding the first 5 bits, that PSNR is around $3dB$ higher than the compared methods. Coding the first 6 bits, that PSNR is around $3.5dB$ higher than the compared methods. Both cases realize lower Bpp values.

When the improved coding scheme which exploits the spatial dependency between bit planes is applied, two separate BSC models could be constructed to describe the dependency based on the downsampled image. The statistical crossover probability CPE when $X_{k-1} = Y_{k-1}$ and CPNE when $X_{k-1} \neq Y_{k-1}$ have been presented in Table. 5.5. In addition, the remaining bit streams keep the predicted values by motion estimation and motion compensation, before upsampling and modification.

At last, the reconstruction image fidelity and compression ratio in Table. 5.5 are

Table 5.5: The improved coding scheme is applied to the downsampled image , the statistics of crossover probabilities of two separate BSC models and averaged conditional entropy and a selection of source coding rate, frame gap $t = 5$.

Bit Plane	CPE	CPNE	Average CE	Coding rate
2^{nd}	0.0331	0.9739	0.2091	0.2500
3^{rd}	0.0327	0.9843	0.2039	0.2500
4^{th}	0.0747	0.9175	0.3852	0.4754
5^{th}	0.1331	0.8047	0.5860	0.6800
6^{th}	0.1956	0.6807	0.7560	0.8500
7^{th}	0.2720	0.6181	0.8795	$\times$
8^{th}	0.3512	0.5814	0.9524	$\times$

obtained:

$$Coding\ on\ first\ 5\ bit : PSNR = 38.42dB, \quad Bpp = 0.4514.$$
$$Coding\ on\ first\ 6\ bit : PSNR = 38.98dB, \quad Bpp = 0.6639.$$

Thus, when the PSNRs remain the same values, the improved coding scheme is successful to a higher compression ratio.

Finally, Table. 5.6 summarizes the compression ratio and reconstruction image fidelity of those methods mentoned above .

As illustrated in the Table. 5.6, especially the performance of those methods (stated in bold) have demonstrated that the improved coding scheme is able to reduce the bit per pixel value when the PSNR remains the same value where the improved coding scheme is not applied. In addition, the compression at the reduced resolution is successful to promote the compression ratio greatly. No matter what resolution it

Table 5.6: Summarize the compression ratios and reconstruction PSNRs for different cases via different methods.

Method	PSNR	Bpp	Compression ratio
Orig	39.31dB	2.8875	2.77:1
DS 5	38.42dB	0.6146	13.02:1
DS 6	38.98dB	0.8646	9.25:1
Orig & Improved	**39.31dB**	**2.2342**	**3.58:1**
DS 5 & Improved	**38.42dB**	**0.4514**	**17.72:1**
DS 6 & Improved	**38.98dB**	**0.6639**	**12.05:1**
Comp-by-DS	35.50dB	2	4:1
DSU	35.44dB	3	2.67:1

is, the proposed methods outperforms the compared methods: best compression-by-downsampling and DSU.

Chapter 6

Conclusions and Future Work

6.1 Summary of the Thesis

In this thesis, we have developed a low-complexity compression codec for high frame rate video. The key technique of our proposed codec is the Slepian-Wolf coding. To facilitate the raw video signal's transmission from the high speed camera array sensor to the massive data storage device, SW encoding is able to realize a fast and light-duty encoder with lower computational cost. Compared with the conventional video encoding stated in the advanced video compression standard H.264, there are only simple inner product and XOR operations in our developed SW encoder. Furthermore, this thesis separates the frame resolution to two different cases: original and reduced spatial resolutions, satisfying different compression requirements. Regardless of the spatial resolution, every RGB pixel in the image is transformed into a 8-bits length bit sequence. We assemble the bits in the same bit position from the 1^{st} to the 8^{th}, composing 8 different bit streams. The 8 assembled bit streams are called bit planes as well. The corresponding SI image is predicted by motion estimation and

motion compensation at decoder. With respect to each bit plane, there is a dependency between each pair of bit streams original data $\boldsymbol{X}_k$ and SI $\boldsymbol{Y}_k$. Then a BSC model is constructed to describe the dependency. There will be a statistical crossover probability $p(X_k|Y_k)$ for the modeled BSC. Meanwhile, the conditional entropy of the BSC is obtained by (3.1). By exploiting the dependency between two consecutive bit planes, an improved scheme is then proposed to promote the encoding performance. Assuming that the current bit plane is only dependent on the previous bit plane and conditionally independent from the bit plane before the previous one, there will be two separate BSC modeled for the situation when $X_{k-1} = Y_{k-1}$ and the situation when $X_{k-1} \neq Y_{k-1}$ respectively. Consequently, the conditional entropy of k-th bit plane is obtained by the averaging in (3.9). In addition, after selecting the source coding rate, with which the parity check matrix $\mathbf{H}$ is produced, the transmitted data is obtained by (3.2), namely the syndrome sequence. Only the source coding rate is equal or larger than the conditional entropy of the modeled BSC, there will be no error in decoding when the data source length tends to infinity.

On the relatively heavy-duty decoder side, ME and MC predict the SI image first. The motion of objects is considered slow and translation, since the frame rate of experimental video is relatively high. Assuming that the movement is along a straight trajectory, block-matching algorithm is easy and efficient to find the best matching block pairs. The values of current macro blocks are computed by averaging the values of matching block pair in previous and subsequent reference frames (4.2). With the help of side information, SW decoder applies belief propagation algorithm to decode the received data. When it reaches a certain iteration number or the beliefs of variable nodes converge, the iterative decoding process is expected to stop,

obtaining a reconstruction of bit streams. With regard to the compression at original resolution, only the first 5 bit planes need to be encoded and decoded, while as for the compression at reduced resolution, the first 5 or 6 bit planes are chosen to be coded. Meanwhile, the remaining bit streams keep the predicted values by ME and MC. Then we retransform the decoded bit streams to RGB pixel values. If the compression is implemented based on the original image, the decompression is resolved after SW decoding. However, if it is based on the downsampled image, the next two designed procedures are supposed to be carried out. The first one is upsamling the SW decoded image to its original spatial resolution. Tradition techniques are the upsampling interpolations such as 'box','bilinear', and 'bicubic'. To achieve a higher reconstruction fidelity, the state-of-the-art learning based super-resolution technique: $A+$ is applied here. Although $A+$ outperforms the other traditional interpolations, there are still some important image details lost. That is because some high frequency information has been thrown away during the scale resizing. Thus, we utilize ME and MC again to modify the upsampled image. Fixing the macro block in current frame, we apply BMA to find the matched block pair in the reference frames. The modification for the current block is resolved by comparing the pixel values in the same collocated position within a macro block region. At last, the decompression on the downsampled image completes after that modification.

Besides, a good degree distribution plays an important role in reducing the source coding rate. EXIT charts is able to optimize the degree distributions by solving (3.7) and (3.8). In addition, those degree distributions generated by the EXIT charts techniques in [50] or obtained from LTHC database [46] are also able to be utilized

directly. Here, the 38 optimized degree distributions stated in [26], which were obtained from LTHC database [46], are employed here to facilitate the selection of source coding rates. Based on the simulation by those irregular degree distributions, those optimized ones realize outstanding performance in decoding.

According to the experimental results, it validates that the improved coding scheme cuts down the bit per pixel value while remaining the same reconstruction fidelity. Furthermore, compression at different spatial resolution can realize different compression goals. Especially, with regard to the compression on the downsampled image, the compression ratio is promoted remarkably. Thus, the validity of the developed codec is proved, and the proposed compression system outperforms the compared methods: compression-by-downsampling with 'bicubic' and $A+$, and DSU.

6.2 Future Work

In the demonstration above, the estimation of source coding rate is difficult to obtain before SW coding. Acquisition frame rate, motion speed, background, frame gap between the current and reference frames have different influences on the implementation of motion estimation and motion compensation. In this thesis, the statistical ranges of condition entropy are not obtained when high frame rate video varies. Therefore, it is necessary to send a feedback from the decoder to the encoder, by which the BSC model is able to be constructed to describe their dependency. This is a constraint of the proposed compression codec for real-time video coding. In the future, an effective estimation of source coding rates is supposed to be put forward to modify the proposed compression syetem .

Appendix A

Your Appendix

The optimized degree distributions for each code rates have been obtained from LTHC database [46], by optimization the algorithm in [50]. Especially, there are 38 optimized degree distributions from LTHC database, which are employed in Slepian-Wolf coding for the high frame rate video compression [26]. The coding rate designed here denotes the source coding rate, R_s.

1. The source coding rate $R_s = 0.0300$. Obtained from LTHC database.

$$\lambda(x) = \begin{cases} 0.0976872000x + 0.2084250000x^2 + .0239832000x^4 \\ + 0.0025877000x^5 + 0.0030756200x^6 + 0.2216020000x^7 \\ + 0.0791919000x^{15} + 0.0178188000x^{17} + 0.0707131000x^{21} \\ + 0.0032752600x^{30} + 0.0539584000x^{39} + 0.0959645000x^{40} \\ + 0.0373115000x^{49} + 0.0555949000x^{64} + 0.0288105000x^{66}, \end{cases} \tag{A.1}$$

$$\rho(x) = x^{199}.$$

2. The source coding rate $R_s = 0.0500$. Obtained from LTHC database.

$$\lambda(x) = \begin{cases} 0.1130550000x + 0.2223260000x^2 + 0.0217564000x^5 \\ + 0.1437610000x^6 + 0.0077757700x^7 + 0.0978175000x^8 \\ + 0.0282852000x^{13} + 0.0669393000x^{15} + 0.0730412000x^{26} \\ + 0.0149455000x^{34} + 0.2102970000x^{35}, \end{cases} \tag{A.2}$$

$$\rho(x) = x^{109}.$$

3. The source coding rate $R_s = 0.0800$. Obtained from LTHC database.

$$\lambda(x) = \begin{cases} 0.1172100000x + 0.2107930000x^2 + 0.1521100000x^6 \\ + 0.0845317000x^7 + 0.0236898000x^8 + 0.0049341200x^{15} \\ + 0.0493028000x^{16} + 0.1133230000x^{19} + 0.0147003000x^{23} \\ + 0.1190500000x^{40} + 0.0175400000x^{42} + 0.0928161000x^{46}, \end{cases} \tag{A.3}$$

$$\rho(x) = x^{69}.$$

4. The source coding rate $R_s = 0.1000$. Obtained from LTHC database.

$$\lambda(x) = \begin{cases} 0.1173140000x + 0.1997290000x^2 + 0.1935750000x^6 \\ + 0.0179234000x^7 + 0.0322564000x^8 + 0.0133904000x^{16} \\ + 0.1638470000x^{18} + 0.0222610000x^{20} + 0.1509760000x^{48} \\ + 0.0445984000x^{50} + 0.0131310000x^{51} + 0.0168516000x^{56} \\ + 0.0141470000x^{58}, \end{cases} \tag{A.4}$$

$$\rho(x) = 0.5x^{56} + 0.5x^{57}.$$

5. The source coding rate $R_s = 0.1300$. Obtained from LTHC database.

$$\lambda(x) = \begin{cases} 0.1154540000x + 0.1846430000x^2 + 0.1872730000x^6 \\ + 0.0107396000x^7 + 0.0107802000x^8 + 0.0298498000x^9 \\ + 0.0676952000x^{17} + 0.0713005000x^{21} + 0.0311166000x^{22} \\ + 0.0523218000x^{25} + 0.1940290000x^{65} + 0.0148834000x^{69} \\ + 0.0192438000x^{72} + 0.0020586100x^{89} + 0.0086120200x^{99}, \end{cases} \tag{A.5}$$
$$\rho(x) = 0.5x^{45} + 0.5x^{46}.$$

6. The source coding rate $R_s = 0.1500$. Obtained from LTHC database.

$$\lambda(x) = \begin{cases} 0.0903419000x + 0.1760760000x^2 + 0.3044350000x^6 \\ + 0.1356970000x^{15} + 0.0127703000x^{16} + 0.0764734000x^{22} \\ + 0.1680910000x^{27} + 0.0361156000x^{49}, \end{cases} \tag{A.6}$$
$$\rho(x) = x^{39}.$$

7. The source coding rate $R_s = 0.1800$. Obtained from LTHC database.

$$\lambda(x) = \begin{cases} 0.1267310000x + 0.1851360000x^2 + 0.1895540000x^6 \\ + 0.0406345000x^7 + 0.0170619000x^{15} + 0.0511614000x^{18} \\ + 0.0888360000x^{19} + 0.0299488000x^{20} + 0.0217271000x^{21} \\ + 0.0074947900x^{32} + 0.0092171200x^{39} + 0.0078997400x^{53} \\ + 0.0871835000x^{60} + 0.0412600000x^{62} + 0.0890379000x^{63} \\ + 0.0071160400x^{68}, \end{cases} \tag{A.7}$$
$$\rho(x) = 0.7x^{31} + 0.3x^{32}.$$

8. The source coding rate $R_s = 0.2000$. Obtained from LTHC database.

$$\begin{aligned} \lambda(x) = \Big\{ & 0.0815474000x + 0.1982150000x^2 + 0.7202380000x^{19}, \\ & \rho(x) = x^{34}. \end{aligned} \tag{A.8}$$

9. The source coding rate $R_s = 0.2281$. Obtained from EXIT chart based design.

$$\lambda(x) = \begin{cases} 0.0000696838x + 0.5645527721x^2 + 0.0006797545x^3 \\ + 0.0298558824x^4 + 0.0010561819x^5 + 0.0162886198x^6 \\ + 0.2852260980x^7 + 0.0945686623x^8 + 0.0003948341x^9 \\ + 0.0017553939x^{10} + 0.0010631110x^{11} + 0.0006613925x^{12} \\ + 0.0004469575x^{13} + 0.0003264797x^{14} + 0.0002535001x^{15} \\ + 0.0002065416x^{16} + 0.0001745563x^{17} + 0.0001515564x^{18} \\ + 0.0001343969x^{19} + 0.0001209905x^{20} + 0.0001102764x^{21} \\ + 0.0001014784x^{22} + 0.0000940540x^{23} + 0.0000877841x^{24} \\ + 0.0000822511x^{25} + 0.0000775660x^{26} + 0.0000733644x^{27} \\ + 0.0000697260x^{28} + 0.0000664825x^{29} + 0.0000636670x^{30} \\ + 0.0000611618x^{31} + 0.0000589626x^{32} + 0.0000570479x^{33} \\ + 0.0000554059x^{34} + 0.0000539797x^{35} + 0.0000527396x^{36} \\ + 0.0000517750x^{37} + 0.0000508221x^{38} + 0.0000500451x^{39} \\ + 0.0000494506x^{40} + 0.0000489172x^{41} + 0.0000484412x^{42} \\ + 0.0000481137x^{43} + 0.0000476051x^{44} + 0.0000472480x^{45} \\ + 0.0000467789x^{46} + 0.0000463338x^{47} + 0.0000458079x^{48} \\ + 0.0000453462x^{49} + 0.0000446644x^{50} + 0.0000441249x^{51} \\ + 0.0000436608x^{52} + 0.0000435132x^{53} + 0.0000440380x^{54}, \end{cases} \tag{A.9}$$

$$\rho(x) = x^{17}.$$

10. The source coding rate $R_s = 0.2500$. Obtained from LTHC database.

$$\lambda(x) = \begin{cases} 0.1118170000x + 0.1479280000x^2 + 0.0721407000x^5 \\ + 0.2464250000x^6 + 0.0021321100x^8 + 0.4195580000x^{29}, \end{cases} \tag{A.10}$$
$$\rho(x) = x^{23}.$$

11. The source coding rate $R_s = 0.2780$. Obtained from EXIT chart based design.

$$\lambda(x) = \begin{cases} 0.0365683268x + 0.4339804675x^2 + 0.0001299819x^3 \\ + 0.0171330361x^4 + 0.0023872502x^5 + 0.1923109345x^6 \\ + 0.0016539566x^7 + 0.0012354770x^8 + 0.0009357160x^9 \\ + 0.0010813697x^{10} + 0.0013693568x^{11} + 0.0024491406x^{12} \\ + 0.0001371963x^{13} + 0.2953988307x^{14} + 0.0072606502x^{15} \\ + 0.0021649586x^{16} + 0.0009936879x^{17} + 0.0005717216x^{18} \\ + 0.0003736405x^{19} + 0.0002657385x^{20} + 0.0002003367x^{21} \\ + 0.0001578227x^{22} + 0.0001285540x^{23} + 0.0001075570x^{24} \\ + 0.0000919092x^{25} + 0.0000799746x^{26} + 0.0000706532x^{27} \\ + 0.0000632326x^{28} + 0.0000571490x^{29} + 0.0000521729x^{30} \\ + 0.0000480070x^{31} + 0.0000444806x^{32} + 0.0000414675x^{33} \\ + 0.0000389182x^{34} + 0.0000366747x^{35} + 0.0000347150x^{36} \\ + 0.0000329633x^{37} + 0.0000314569x^{38} + 0.0000301197x^{39} \\ + 0.0000288897x^{40} + 0.0000278226x^{41} + 0.0000268317x^{42} \\ + 0.0000259364x^{43} + 0.0000251422x^{44} + 0.0000244178x^{45} \\ + 0.0000237279x^{46} + 0.0000231042x^{47} + 0.0000225359x^{48} \\ + 0.0000219885x^{49}, \end{cases} \tag{A.11}$$

$$\rho(x) = 0.3x^{15} + 0.7x^{16}.$$

12. The source coding rate $R_s = 0.3000$. Obtained from EXIT chart based design.

$$\lambda(x) = \begin{cases} 0.1392280000x + 0.2007590000x^2 + 0.2522010000x^6 \\ + 0.0134136000x^{11} + 0.1710390000x^{17} + 0.0424794000x^{31} \\ + 0.0855733000x^{41} + 0.0953074000x^{49}, \end{cases} \tag{A.12}$$

$$\rho(x) = 0.3x^{16} + 0.7x^{17}.$$

13. The source coding rate $R_s = 0.3277$. Obtained from LTHC database.

$$\lambda(x) = \begin{cases} 0.0530701457x + 0.3670818133x^2 + 0.0000467414x^3 \\ + 0.0001693434x^4 + 0.0001839202x^5 + 0.0069791251x^6 \\ + 0.2340869351x^7 + 0.0293543350x^8 + 0.0010512763x^9 \\ + 0.0005884899x^{10} + 0.0004336224x^{11} + 0.0004421017x^{12} \\ + 0.0003594770x^{13} + 0.0004299928x^{14} + 0.0004909475x^{15} \\ + 0.0005842977x^{16} + 0.0006793042x^{17} + 0.0007968070x^{18} \\ + 0.0012180480x^{19} + 0.0004888936x^{20} + 0.0233161791x^{21} \\ + 0.1332294395x^{22} + 0.1290474222x^{23} + 0.0085915359x^{24} \\ + 0.0029141588x^{25} + 0.0012946281x^{26} + 0.0007207962x^{27} \\ + 0.0004649800x^{28} + 0.0003297155x^{29} + 0.0002492324x^{30} \\ + 0.0001973730x^{31} + 0.0001619024x^{32} + 0.0001364856x^{33} \\ + 0.0001174977x^{34} + 0.0001029730x^{35} + 0.0000914730x^{36} \\ + 0.0000822134x^{37} + 0.0000744651x^{38} + 0.0000681241x^{39} \\ + 0.0000626681x^{40} + 0.0000580978x^{41} + 0.0000541896x^{42} \\ + 0.0000508900x^{43} + 0.0000479426x^{44}, \end{cases} \tag{A.13}$$

$$\rho(x) = 0.4x^{14} + 0.6x^{15}.$$

14. The source coding rate $R_s = 0.3500$. Obtained from LTHC database.

$$\lambda(x) = \begin{cases} 0.1577030000^x + 0.1991060000x^2 + 0.0324838000x^5 \\ + 0.1985480000x^6 + 0.0807045000x^7 + 0.2997880000x^{21} \\ + 0.0310813000x^{22} + 0.0005852400x^{33}, \end{cases} \tag{A.14}$$
$$\rho(x) = x^{13}.$$

15. The source coding rate $R_s = 0.3765$. Obtained from EXIT chart based design.

$$\lambda(x) = \begin{cases} 0.0749746514x + 0.3276397454x^2 + 0.0000133230x^3 \\ + 0.0000334288x^4 + 0.0000536346x^5 + 0.0153727931x^6 \\ + 0.0815021268x^7 + 0.2102399279x^8 + 0.0002396780x^9 \\ + 0.0001470809x^{10} + 0.0001101496x^{11} + 0.0000966061x^{12} \\ + 0.0000903891x^{13} + 0.0000890635x^{14} + 0.0000914754x^{15} \\ + 0.0000970680x^{16} + 0.0001067689x^{17} + 0.0001207840x^{18} \\ + 0.0001414627x^{19} + 0.0001706132x^{20} + 0.0002138222x^{21} \\ + 0.0002778502x^{22} + 0.0003801933x^{23} + 0.0005575559x^{24} \\ + 0.0009942277x^{25} + 0.0038202879x^{26} + 0.1061438593x^{27} \\ + 0.1680729114x^{28} + 0.0046663975x^{29} + 0.0013383072x^{30} \\ + 0.0007207862x^{31} + 0.0004404507x^{32} + 0.0002851821x^{33} \\ + 0.0002019419x^{34} + 0.0001557353x^{35} + 0.0001265960x^{36} \\ + 0.0001058190x^{37} + 0.0000899191x^{38} + 0.0000773867x^{39}, \end{cases} \tag{A.15}$$
$$\rho(x) = 0.2x^{12} + 0.8x^{13}.$$

16. The source coding rate $R_s = 0.4010$. Obtained from EXIT chart based design.

$$\lambda(x) = \begin{cases} 0.0665052292x + 0.3059480919x^2 + 0.0001148099x^3 \\ + 0.0008967976x^4 + 0.0004931189x^5 + 0.0885292799x^6 \\ + 0.0028298017x^7 + 0.0933425249x^8 + 0.0270974961x^9 \\ + 0.0288241934x^{10} + 0.0067485256x^{11} + 0.0000092266x^{12} \\ + 0.0622890112x^{13} + 0.0120946708x^{14} + 0.0041578038x^{15} \\ + 0.0023011096x^{16} + 0.0016328252x^{17} + 0.0015906365x^{18} \\ + 0.0014174658x^{19} + 0.0012243279x^{20} + 0.0010393924x^{21} \\ + 0.0008685978x^{22} + 0.0007279214x^{23} + 0.0006136080x^{24} \\ + 0.0005260355x^{25} + 0.0004590425x^{26} + 0.0004104392x^{27} \\ + 0.0003758855x^{28} + 0.0003539140x^{29} + 0.0003424701x^{30} \\ + 0.0003416727x^{31} + 0.0003517762x^{32} + 0.0003755456x^{33} \\ + 0.0004182788x^{34} + 0.0004931475x^{35} + 0.0006325812x^{36} \\ + 0.0009463712x^{37} + 0.0020643746x^{38} + 0.2806119994x^{39}, \end{cases} \tag{A.16}$$

$$\rho(x) = x^{13}.$$

17. The source coding rate $R_s = 0.4257$. Obtained from EXIT chart based design.

$$\lambda(x) = \begin{cases} 0.1242596346x + 0.3029532976x^2 + 0.0000000503x^3 \\ + 0.1130781949x^4 + 0.0000019471x^5 + 0.1030539620x^6 \\ + 0.0000011974x^7 + 0.0000000440x^8 + 0.0000002074x^9 \\ + 0.0000002709x^{10} + 0.0000003055x^{11} + 0.0000003467x^{12} \\ + 0.0000004646x^{13} + 0.3566500769x^{14}, \end{cases} \tag{A.17}$$

$$\rho(x) = 0.5x^9 + 0.5x^{10}.$$

18. The source coding rate $R_s = 0.4508$. Obtained from EXIT chart based design.

$$\lambda(x) = \begin{cases} 0.0999701013x + 0.2884808570x^2 + 0.0000004467x^3 \\ + 0.0000007193x^4 + 0.0000015285x^5 + 0.2323194203x^6 \\ + 0.0053583039x^7 + 0.0000044615x^8 + 0.0000014946x^9 \\ + 0.0000008772x^{10} + 0.0000006229x^{11} + 0.0000004958x^{12} \\ + 0.0000004363x^{13} + 0.0000004124x^{14} + 0.0000003973x^{15} \\ + 0.0000004041x^{16} + 0.0000004193x^{17} + 0.0000004649x^{18} \\ + 0.0000005231x^{19} + 0.0000006215x^{20} + 0.0000008004x^{21} \\ + 0.0000011595x^{22} + 0.0000022604x^{23} + 0.3738527719x^{24}, \end{cases} \tag{A.18}$$

$$\rho(x) = 0.6x^{10} + 0.4x^{11}.$$

19. The source coding rate $R_s = 0.4754$. Obtained from EXIT chart based design.

$$\lambda(x) = \begin{cases} 0.1153122367x + 0.2911329163x^2 + 0.0000013866x^3 \\ + 0.0000237308x^4 + 0.0000055074x^5 + 0.1563430715x^6 \\ + 0.1185591176x^7 + 0.0000123470x^8 + 0.0000039402x^9 \\ + 0.0000021293x^{10} + 0.0000016722x^{11} + 0.0000014129x^{12} \\ + 0.0000012628x^{13} + 0.0000011759x^{14} + 0.0000011534x^{15} \\ + 0.0000011760x^{16} + 0.0000012159x^{17} + 0.0000013357x^{18} \\ + 0.0000015028x^{19} + 0.0000017858x^{20} + 0.0000022500x^{21} \\ + 0.0000032496x^{22} + 0.0000062909x^{23} + 0.3185781326x^{24}, \end{cases} \tag{A.19}$$

$$\rho(x) = 0.7x^9 + 0.3x^{10}.$$

20. The source coding rate $R_s = 0.5000$. Obtained from LTHC database.

$$\lambda(x) = \begin{cases} 0.1527930000x + 0.2823500000x^2 + 0.0062193000x^3 \\ + 0.5586370000x^{19}, \end{cases} \tag{A.20}$$

$$\rho(x) = x^9.$$

21. The source coding rate $R_s = 0.5261$. Obtained from EXIT chart based design.

$$\lambda(x) = \begin{cases} 0.1460827952x + 0.2706599775x^2 + 0.0375150106x^4 \\ + 0.2419193856x^6 + 0.3038228311x^{21}, \end{cases} \tag{A.21}$$

$$\rho(x) = 0.3x^7 + 0.7x^8.$$

22. The source coding rate $R_s = 0.5515$. Obtained from EXIT chart based design.

$$\lambda(x) = \begin{cases} 0.1483309314x + 0.2284719333x^2 + 0.0000030096x^3 \\ + 0.0949413930x^4 + 0.0001039602x^5 + 0.1805360736x^6 \\ + 0.0000004930x^7 + 0.0000044589x^8 + 0.0000001607x^9 \\ + 0.0000056705x^{10} + 0.0000009369x^{11} + 0.0000017390x^{12} \\ + 0.0000020428x^{13} + 0.0000021972x^{14} + 0.0000024381x^{15} \\ + 0.0000027101x^{16} + 0.0000031452x^{17} + 0.0000038634x^{18} \\ + 0.0000048678x^{19} + 0.0000067093x^{20} + 0.0000100212x^{21} \\ + 0.0000167482x^{22} + 0.0000309110x^{23} + 0.3475135857x^{24}, \end{cases} \tag{A.22}$$

$$\rho(x) = 0.3x^7 + 0.7x^8.$$

23. The source coding rate $R_s = 0.5765$. Obtained from EXIT chart based design.

$$\lambda(x) = \begin{cases} 0.1603735738x + 0.2493334680x^2 + 0.0000000001x^3 \\ + 0.1398432549x^4 + 0.0009310009x^6 + 0.1919102356x^8 \\ + 0.2576084667x^{24}, \end{cases} \tag{A.23}$$

$$\rho(x) = 0.2x^6 + 0.8x^7.$$

24. The source coding rate $R_s = 0.6006$. Obtained from EXIT chart based design.

$$\lambda(x) = \begin{cases} 0.2365103129x + 0.0000003166x^2 + 0.3137349967x^3 \\ + 0.0714882030x^4 + 0.0000017984x^5 + 0.0000026206x^6 \\ + 0.0000020295x^7 + 0.0000013959x^8 + 0.0000009058x^9 \\ + 0.1706734148x^{10} + 0.0004347797x^{11} + 0.0000483848x^{12} \\ + 0.0000733639x^{13} + 0.0000728850x^{14} + 0.0000680328x^{15} \\ + 0.0000796862x^{16} + 0.0000905415x^{17} + 0.0001139106x^{18} \\ + 0.0001440668x^{19} + 0.0002042075x^{20} + 0.0003112035x^{21} \\ + 0.0006324201x^{22} + 0.0182989558x^{23} + 0.1854026561x^{24} \\ + 0.0016072092x^{25} + 0.0000017023x^{26}, \end{cases} \tag{A.24}$$

$$\rho(x) = 0.9x^6 + 0.1x^7.$$

25. The source coding rate $R_s = 0.6257$. Obtained from EXIT chart based design.

$$\lambda(x) = \begin{cases} 0.2345281333x + 0.2494103165x^3 + 0.0868495874x^4 \\ + 0.1048097509x^6 + 0.3244022119x^{23}, \end{cases} \tag{A.25}$$

$$\rho(x) = 0.9x^6 + 0.1x^7.$$

26. The source coding rate $R_s = 0.6500$. Obtained from LTHC database.

$$\lambda(x) = \begin{cases} 0.2454340000x + 0.1921240000x^2 + 0.1357320000x^5 \\ + 0.0838990000x^6 + 0.1116600000x^{12} + 0.0029827600x^{14} \\ + 0.0222593000x^{15} + 0.0742901000x^{28} + 0.1316190000x^{32}, \end{cases} \tag{A.26}$$

$$\rho(x) = 0.5x^5 + 0.5x^6.$$

27. The source coding rate $R_s = 0.6800$. Obtained from LTHC database.

$$\lambda(x) = \begin{cases} 0.2177260000x + 0.1634340000x^2 + 0.0001449710x^3 \\ + 0.0000070738x^4 + 0.0980647000x^5 + 0.1018190000x^6 \\ + 0.0537834000x^{13} + 0.0301359000x^{16} + 0.0566144000x^{20} \\ + 0.0109644000x^{26} + 0.0808932000x^{30} + 0.0000059471x^{97} \\ + 0.0001740400x^{98} + 0.1862340000x^{99}, \end{cases} \tag{A.27}$$

$$\rho(x) = 0.9x^6 + 0.1x^7.$$

28. The source coding rate $R_s = 0.7000$. Obtained from LTHC database.

$$\lambda(x) = \begin{cases} 0.2202400000x + 0.1604510000x^2 + 0.1219000000x^5 \\ + 0.0669837000x^6 + 0.0728829000x^{12} + 0.0056090100x^{19} \\ + 0.0223284000x^{21} + 0.0531729000x^{22} + 0.0496530000x^{25} \\ + 0.0222808000x^{26} + 0.2044980000x^{99}, \end{cases} \tag{A.28}$$

$$\rho(x) = 0.1x^5 + 0.9x^6.$$

29. The source coding rate $R_s = 0.7300$. Obtained from LTHC database.

$$\lambda(x) = \begin{cases} 0.2374900000x + 0.1643770000x^2 + 0.1482710000x^5 \\ + 0.0442728000x^6 + 0.0276623000x^{13} + 0.1174440000x^{15} \\ + 0.0384218000x^{30} + 0.0368359000x^{35} + 0.0319294000x^{37} \\ + 0.1532960000x^{99}, \end{cases} \tag{A.29}$$

$$\rho(x) = 0.7x^5 + 0.3x^6.$$

30. The source coding rate $R_s = 0.7500$. Obtained from LTHC database.

$$\lambda(x) = \begin{cases} 0.2911570000x + 0.1891740000x^2 + 0.0408389000x^4 \\ + 0.0873393000x^5 + 0.0074271800x^6 + 0.1125810000x^7 \\ + 0.0925954000x^{15} + 0.0186572000x^{20} + 0.1240640000x^{32} \\ + 0.0160020000x^{39} + 0.0201644000x^{44}, \end{cases} \tag{A.30}$$

$$\rho(x) = 0.8x^4 + 0.2x^5.$$

31. The source coding rate $R_s = 0.7800$. Obtained from LTHC database.

$$\lambda(x) = \begin{cases} 0.2547740000x + 0.1634760000x^2 + 0.0032539300x^4 \\ + 0.1524220000x^5 + 0.0331399000x^6 + 0.0038860400x^9 \\ + 0.0189110000x^{12} + 0.0998195000x^{14} + 0.0151103000x^{27} \\ + 0.0769337000x^{29} + 0.0218393000x^{32} + 0.1564350000x^{99}, \end{cases} \tag{A.31}$$

$$\rho(x) = 0.3x^4 + 0.7x^5.$$

32. The source coding rate $R_s = 0.8000$. Obtained from LTHC database.

$$\lambda(x) = \begin{cases} 0.2920250000x + 0.1739820000x^2 + 0.0523131000x^4 \\ + 0.0257749000x^5 + 0.1220460000x^6 + 0.0218315000x^8 \\ + 0.0209295000x^{10} + 0.0322251000x^{14} + 0.1127710000x^{23} \\ + 0.0001708020x^{25} + 0.0328124000x^{31} + 0.0274748000x^{44} \\ + 0.0048302000x^{53} + 0.0126282000x^{59} + 0.0681855000x^{99}, \end{cases} \tag{A.32}$$

$$\rho(x) = x^4.$$

33. The source coding rate $R_s = 0.8200$. Obtained from LTHC database.

$$\lambda(x) = \begin{cases} 0.3037920000x + 0.1731880000x^2 + 0.0671337000x^4 \\ + 0.0123568000x^5 + 0.1341320000x^6 + 0.0314767000x^{12} \\ + 0.0108393000x^{14} + 0.0256390000x^{16} + 0.0910351000x^{19} \\ + 0.0400076000x^{39} + 0.0000240473x^{45} + 0.0117242000x^{51} \\ + 0.0189157000x^{57} + 0.0112433000x^{62} + 0.0684922000x^{76}, \end{cases} \tag{A.33}$$
$$\rho(x) = 0.2x^3 + 0.8x^4.$$

34. The source coding rate $R_s = 0.8500$. Obtained from LTHC database.

$$\lambda(x) = \begin{cases} 0.3151270000x + 0.1902840000x^2 + 0.0449124000x^4 \\ + 0.1705930000x^6 + 0.1405970000x^{17} + 0.0081261000x^{37} \\ + 0.0440236000x^{41} + 0.0863369000x^{66}, \end{cases} \tag{A.34}$$
$$\rho(x) = 0.5x^3 + 0.5x^4.$$

35. The source coding rate $R_s = 0.8800$. Obtained from LTHC database.

$$\lambda(x) = \begin{cases} 0.3424730000x + 0.1650060000x^2 + 0.1203830000x^4 \\ + 0.0191956000x^5 + 0.0120714000x^6 + 0.1416920000x^{10} \\ + 0.0211997000x^{25} + 0.0201976000x^{26} + 0.0185881000x^{34} \\ + 0.0428897000x^{36} + 0.0133019000x^{38} + 0.0021735800x^{39} \\ + 0.0104203000x^{40} + 0.0704081000x^{99}, \end{cases} \tag{A.35}$$
$$\rho(x) = 0.8x^3 + 0.2x^4.$$

36. The source coding rate $R_s = 0.9000$. Obtained from LTHC database.

$$\lambda(x) = \begin{cases} 0.3585670000x + 0.1663620000x^2 + 0.0000299853x^3 \\ + 0.0487523000x^4 + 0.1205300000x^5 + 0.0004778820x^6 \\ + 0.0000422043x^7 + 0.0409013000x^{10} + 0.0744850000x^{13} \\ + 0.0339421000x^{25} + 0.0076194000x^{30} + 0.0564230000x^{34} \\ + 0.0918683000x^{99}, \end{cases} \tag{A.36}$$

$$\rho(x) = x^3.$$

37. The source coding rate $R_s = 0.9300$. Obtained from LTHC database.

$$\lambda(x) = \begin{cases} 0.4050180000x + 0.1716200000x^2 + 0.0995717000x^4 \\ + 0.0446767000x^5 + 0.0379776000x^6 + 0.0612300000x^{10} \\ + 0.0188277000x^{14} + 0.0332702000x^{16} + 0.0026478100x^{17} \\ + 0.0127722000x^{20} + 0.0435222000x^{28} + 0.0075207600x^{50} \\ + 0.0123120000x^{52} + 0.0258378000x^{62} + 0.0065513300x^{63} \\ + 0.0166443000x^{71}, \end{cases} \tag{A.37}$$

$$\rho(x) = 0.4x^2 + 0.6x^3.$$

38. The source coding rate $R_s = 0.9500$. Obtained from LTHC database.

$$\lambda(x) = \begin{cases} 0.4145410000x + 0.1667480000x^2 + 0.0971414000x^4 \\ + 0.0737392000x^5 + 0.0007658270x^6 + 0.0022987300x^8 \\ + 0.0118195000x^9 + 0.0751327000x^{11} + 0.0575786000x^{19} \\ + 0.0063649900x^{26} + 0.0046459300x^{35} + 0.0171996000x^{43} \\ + 0.0443262000x^{62} + 0.0111913000x^{82} + 0.0165064000x^{99}, \end{cases} \tag{A.38}$$

$$\rho(x) = 0.5x^2 + 0.5x^3.$$

Bibliography

[1] G. J. Sullivan and T. Wiegand, "Video compression - from concepts to the h.264/avc standard," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 18–31, 2005.

[2] B. Girod, E. Steinbach, and N. Faerber, "Comparison of the h. 263 and h. 261 video compression standards," in *Standards and Common Interfaces for Video Information Systems*, vol. 1, 1995, pp. 233–251.

[3] T. Von Roden, "H. 261 and mpeg1-a comparison," in *Computers and Communications, 1996., Conference Proceedings of the 1996 IEEE Fifteenth Annual International Phoenix Conference on.* IEEE, 1996, pp. 65–71.

[4] G. Ashraf and M. Chong, "Performance analysis of h. 261 and h. 263 video coding algorithms," in *Consumer Electronics, 1997. ISCE'97., Proceedings of 1997 IEEE International Symposium on.* IEEE, 1997, pp. 153–156.

[5] M. Nilsson and M. Naylor, "Comparison of h.263 and h.26l video compression performance with web-cams," *Electronics Letters*, vol. 39, no. 3, pp. 277–278, 2003.

[6] U. Horn and B. Girod, "Scalable video transmission for the internet," *Computer Networks and ISDN Systems*, vol. 29, no. 15, pp. 1833–1842, 1997.

[7] B. Erol, M. Gallant, G. Cote, and F. Kossentini, "The h.263+ video coding standard: complexity and performance," in *Data Compression Conference, 1998. DCC '98. Proceedings*, 1998, pp. 259–268.

[8] G. Raja and M. J. Mirza, "Performance comparison of advanced video coding h. 264 standard with baseline h. 263 and h. 263+ standards," in *Communications and Information Technology, 2004. ISCIT 2004. IEEE International Symposium on*, vol. 2. IEEE, 2004, pp. 743–746.

[9] O. J. Morris, "Mpeg-2: where did it come from and what is it?" in *MPEG-2 - What it is and What it isn't, IEE Colloquium on*, 1995, pp. 1/1–1/5.

[10] R. Schafer and T. Sikora, "Digital video coding standards and their role in video communications," *Proceedings of the IEEE*, vol. 83, no. 6, pp. 907–924, 1995.

[11] A. Puri and A. Eleftheriadis, "Mpeg-4: An object-based multimedia coding standard supporting mobile applications," *Mobile Networks and Applications*, vol. 3, no. 1, pp. 5–32, 1998.

[12] R. Koenen, "Overview of the mpeg-4 standard," *ISO/IEC JTC1/SC29/WG11 N*, vol. 1730, pp. 11–13, 2002.

[13] A. Puri, X. Chen, and A. Luthra, "Video coding using the h. 264/mpeg-4 avc compression standard," *Signal processing: Image communication*, vol. 19, no. 9, pp. 793–849, 2004.

[14] S.-F. Chang, T. Sikora, and A. Purl, "Overview of the mpeg-7 standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 6, pp. 688–695, 2001.

[15] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h.264/avc video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.

[16] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, "Video coding with h.264/avc: tools, performance, and complexity," *IEEE Circuits and Systems Magazine*, vol. 4, no. 1, pp. 7–28, 2004.

[17] S. Bachu and K. M. Chari, "A review on motion estimation in video compression," in *2015 International Conference on Signal Processing and Communication Engineering Systems*, 2015, pp. 250–256.

[18] M. Manikandan, P. Vijayakumar, and N. Ramadass, "Motion estimation method for video compression - an overview," in *2006 IFIP International Conference on Wireless and Optical Communications Networks*, 2006, pp. 5 pp.–5.

[19] Y.-L. Chan, W.-L. Hui, and W.-C. Siu, "A block motion vector estimation using pattern based pixel decimation," in *Circuits and Systems, 1997. ISCAS '97., Proceedings of 1997 IEEE International Symposium on*, vol. 2, 1997, pp. 1153–1156 vol.2.

[20] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 4, no. 4, pp. 438–442, 1994.

[21] L.-M. Po and W.-C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 313–317, 1996.

[22] S. Zhu and K.-K. Ma, "A new diamond search algorithm for fast block matching motion estimation," in *Proceedings of ICICS, 1997 International Conference on Information, Communications and Signal Processing. Theme: Trends in Information Systems Engineering and Wireless Multimedia Communications (Cat.*, vol. 1, 1997, pp. 292–296 vol.1.

[23] A. Barjatya, "Block matching algorithms for motion estimation," *IEEE Transactions Evolution Computation*, vol. 8, no. 3, pp. 225–239, 2004.

[24] N. Nazari, R. Shams, M. Mohrekesh, and S. Samavi, "Near-lossless compression for high frame rate videos," in *2013 21st Iranian Conference on Electrical Engineering (ICEE)*, 2013, pp. 1–6.

[25] S. Chuah, S. Dumitrescu, and X. Wu, "ℓ_2 optimized predictive image coding with ℓ_∞ bound," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 5271–5281, 2013.

[26] D. H. Schonberg, *Practical distributed source coding and its application to the compression of encrypted data*. University of California, Berkeley, 2007.

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

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

[29] R. P. Westerlaken, S. Borchert, R. K. Gunnewiek, and R. L. Lagendijk, "Dependency channel modeling for a ldpc-based wyner-ziv video compression scheme," in *2006 International Conference on Image Processing*, 2006, pp. 277–280.

[30] J. Ascenso, C. Brites, and F. Pereira, "Design and performance of a novel low-density parity-check code for distributed video coding," in *2008 15th IEEE International Conference on Image Processing*, 2008, pp. 1116–1119.

[31] A. Aaron, R. Zhang, and B. Girod, "Wyner-ziv coding of motion video," in *Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers, 2002.*, vol. 1, 2002, pp. 240–244 vol.1.

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

[33] R. Puri, A. Majumdar, and K. Ramchandran, "Prism: A video coding paradigm with motion estimation at the decoder," *IEEE Transactions on Image Processing*, vol. 16, no. 10, pp. 2436–2448, 2007.

[34] R. Puri, A. Majumdar, P. Ishwar, and K. Ramchandran, "Distributed video coding in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 23, no. 4, pp. 94–106, 2006.

[35] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, and M. Ouaret, "The discover codec: architecture, techniques and evaluation," in *Picture Coding Symposium (PCS" 07)*, no. MMSPL-CONF-2009-014, 2007.

[36] A. D. Liveris, Z. Xiong, and C. N. Georghiades, "Compression of binary sources with side information at the decoder using ldpc codes," *IEEE Communications Letters*, vol. 6, no. 10, pp. 440–442, 2002.

[37] R. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, 1962.

[38] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 619–637, 2001.

[39] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic information transfer functions: model and erasure channel properties," *IEEE Transactions on Information Theory*, vol. 50, no. 11, pp. 2657–2673, 2004.

[40] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Transactions on Communications*, vol. 52, no. 4, pp. 670–678, 2004.

[41] S. Ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Transactions on Communications*, vol. 49, no. 10, pp. 1727–1737, 2001.

[42] S. R. Kollu and H. Jafarkhani, "On the exit chart analysis of low-density parity-check codes," in *GLOBECOM '05. IEEE Global Telecommunications Conference, 2005.*, vol. 3, 2005, pp. 6 pp.–.

[43] X.-Y. Hu, E. Eleftheriou, and D. M. Arnold, "Progressive edge-growth tanner graphs," in *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE*, vol. 2, 2001, pp. 995–1001 vol.2.

[44] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, and V. Stemann, "Practical loss-resilient codes," in *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, 1997, pp. 150–159.

[45] S.-Y. Chung, "On the construction of some capacity-approaching coding schemes," Ph.D. dissertation, Citeseer, 2000.

[46] A. Amraoui and R. Urbanke, "Ldpcopt," *World Wide Web*, 2003.

[47] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

[48] R. Timofte, V. De Smet, and L. Van Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Asian Conference on Computer Vision*. Springer, 2014, pp. 111–126.

[49] Y. Shen and X. Wu, "Down-sampling based embedded compression in video systems," in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2015, pp. 2736–2739.

[50] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic information transfer functions: model and erasure channel properties," *IEEE Transactions on Information Theory*, vol. 50, no. 11, pp. 2657–2673, 2004.