Technische Universität München
Lehrstuhl für Medientechnik

Wyner-Ziv Bayer-pattern Video Coding

Hu Chen, M.Sc.

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:		Univ.-Prof. Dr.-Ing. Klaus Diepold
Prüfer der Dissertation:	1.	Univ.-Prof. Dr.-Ing. Eckehard Steinbach
	2.	Univ.-Prof. Dr.-Ing. André Kaup (Friedrich-Alexander-Universität Erlangen-Nürnberg)

Die Dissertation wurde am 17.04.2012 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 16.07.2012 angenommen.

To my beloved wife, parents and parents-in-law

Abstract

This thesis addresses the problem of Bayer-pattern video communications using Wyner-Ziv video coding (WZVC). It is studied how to take advantage of the low encoding complexity and the inherent error resiliency of WZVC for video compression and transmission. Moreover, it is investigated how to optimize the coding process specifically for Bayer-pattern video in order to improve the coding efficiency.

To begin with, a state-of-the-art Wyner-Ziv video codec using turbo codes is optimized and its functionality is extended. Firstly, the turbo decoding is optimized by making good use of error-free parity bits, which leads to a considerable improvement of coding efficiency. Secondly, a comparative study of bit-plane based coding using binary turbo codes and symbol based coding using trellis merging in turbo decoding is carried out. Experimental results show that the latter one is considerably more error-resilient than the former one when the transmission channel is quite noisy. Thirdly, motion estimation is distributed between encoder and decoder flexibly. In this way, motion estimation is made scalable at the encoder.

Moreover, error-resilient video transmission over an AWGN channel using WZVC is studied. This study involves statistical modeling of the sufficient coding rates, estimation of decoding failure probability, end-to-end distortion estimation and joint source-channel rate control using adaptive quantization. For joint source-channel coding using WZVC, two alternative solutions are compared. The first one is to use WZVC for both source and channel coding, while the second one confines WZVC to source coding and carries out channel coding at a separate stage. As a result, the former one is significantly more error-resilient than the latter one. Therefore, WZVC is a promising candidate solution for joint source-channel coding of video.

Furthermore, the color space transform, specifically the chroma subsampling, is optimized specifically for Bayer-pattern video. Two color space transform methods, namely B-4:2:0 and B-4:2:2 are proposed. Chroma sample positions are more reasonably chosen than the nominal sample positions in the H.264/AVC standard. Therefore, the proposed methods lead to a significant improvement of the reconstructed RGB video quality. Particularly, the method B-4:2:2 keeps only half of the luma pixels compared to the method B-4:2:2, and thus reduces

the coding complexity almost by half. The combination of the method B-4:2:2 and WZVC can lead to a solution of very low encoding complexity.

To sum up, this thesis ends up with a Wyner-Ziv Bayer-pattern video coding system with two good features: low encoding complexity and high error resiliency. On one hand, low encoding complexity can be enabled by partially or entirely moving the motion estimation to the decoder, at the expense of a coding efficiency loss. On the other hand, strong error resiliency is achievable, when the encoder has a high computational capacity for accurate motion estimation and sophisticated joint source-channel rate control.

Acknowledgements

First and foremost, i would like to express my sincere gratitude to my advisor, Prof. Eckehard Steinbach, who assigned me this interesting thesis topic, advised me with great patience and gave me much independence in research work. And I am thankful to my second examiner Prof. André Kaup for reading my thesis and giving pertinent comments. At the same time, I would like to thank Prof. Klaus Diepold for chairing my thesis defence.

Moreover, i wish to thank Deutsche Telekom Stiftung for financially supporting my research for three years. And i am very grateful to Prof. Tobias Kretschmer for his nice mentoring.

Furthermore, i am indebted to all my colleagues in our Institute for Media Technology. I would like to give special thanks to Dr. Wei Tu who helped me in all aspects when i started my work at our institute. I wish to thank Dr. Ingo Bauermann for helping me capture Bayer-pattern video sequences in our lab. And I am grateful to Yang Peng and Fan Zhang for sharing their expertise in conventional video coding and transmission with me. I also wish to thank Jianshu Chao for his help in C programming and Dr. Liang Zhou for his guidance in convex optimization.

Finally, i wish to thank my wife Liyun for her company throughout my thesis work. She has spent much time taking care of me and our family, even if she is quite occupied with her own doctoral research. And i would like to thank my father for asking about my work progress and pushing me to work hard from time to time. I wish to thank my mother for keeping my well-being in mind. Also many thanks to my parents-in-law and particularly for their enthusiasm in photography which stimulates my research interest in image and video processing. My thesis is dedicated to my beloved wife Liyun, my liberal parents and my kind parents-in-law.

Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Description	Definition
AWGN	Additive White Gaussian Noise	page 23
BER	Bit Error Rate	page 24
BPSK	Binary Phase Shift Keying	page 25
BSC	Binary Symmetric Channel	page 20
CCR	Channel Coding Rate	page 24
CRC	Cyclic Redundancy Check	page 13
CSI	Channel State Information	page 23
CSR	Channel-to-Source Rate Ratio	page 24
CST	Color Space Transform	page 24
DFP	Decoding Failure Probability	page 23
DSC	Distributed Source Coding	page 3
LDPC	Low-density Parity-check	page 15
MCE	Motion Compensated Extrapolation	page 10
MCI	Motion Compensated Interpolation	page 10
MCP	Motion Compensated Prediction	page 4
ME	Motion Estimation	page 21
MEC	Motion Estimation and Compensation	page 10
MSB	Most Significant Bit	page 11
QP	Quantization Parameter	page 40
RCPTC	Rate Compatible Punctured Turbo Codes	page 12
RD	Rate-Distortion	page 21
SER	Symbol Error Rate	page 24
SSD	Sum of Squared Differences	page 31
SW	Slepian-Wolf	page 3
WER	Word Error Rate	page 15
WZ	Wyner-Ziv	page 3
WZVC	Wyner-Ziv Video Coding	page 3
WZVC-B	Bit-plane Based WZVC	page 22
WZVC-S	Symbol Based WZVC	page 22

Chapter 1

Introduction

Most consumer digital cameras capture video using a single chip. Single chip cameras do not capture RGB triples for every pixel, but a subsampled version with only one color component per pixel, typically in Bayer pattern [Bay76] as shown in Fig. 1.1. Out of every two-by-two pixel array, two pixels are green, one is red and one is blue. A full resolution video is constructed from the Bayer pattern by interpolating the missing color components for every pixel from neighboring values, a process typically referred to as demosaicing. Thanks to the Bayer pattern, only one chip is required to capture a color video. Compared to broadcast cameras with three chips, each for one color component, single chip cameras can be up to five times smaller and lighter [TKA+90], and cost significantly less, while still being able to maintain a high video quality.

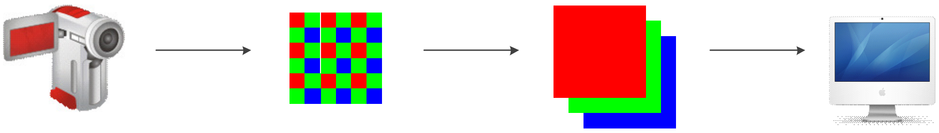


Figure 1.1: Bayer-pattern video capture, demosaicing and display

A typical Bayer-pattern video communications system is illustrated in Fig. 1.2. The encoder begins with demosaicing, converting Bayer-pattern RGB images to full color RGB images. Next, the full color images in the RGB domain are converted to the YUV domain via color space transform. Typically, the chroma samples U and V are subsampled. Different sampling modes and sampling rates lead to different YUV data formats. Then, the resulting YUV video data are forwarded to a video encoder for compression. Afterwards, the coded video data can either be saved in some storage devices or transmitted over a communication network to one or more receivers. At the receiver, the YUV data are decoded and converted back to full color RGB images via chroma upsampling and inverse color space transform. These reconstructed full color RGB images are ready to be displayed.

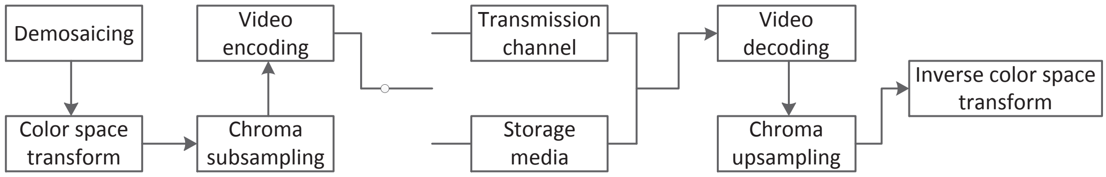


Figure 1.2: Bayer-pattern video communications

In this video communications system, video coding plays a central role. And the focus of this thesis is to investigate how to use Wyner-Ziv video coding, a fundamentally different video coding paradigm from conventional video coding standards, to compress and transmit Bayer-pattern video. The objective is to develop a Wyner-Ziv Bayer-pattern video coding system of high coding efficiency, strong error resiliency and low encoding complexity.

The remaining part of this chapter is organized as follows. To begin with, conventional video coding will be reviewed briefly in Section 1.1 and its limitations will be pointed out. Next, Wyner-Ziv video coding will be introduced in Section 1.2 and it will be discussed how this new coding paradigm can complement conventional video coding. After this, major contributions of this thesis will be summarized in Section 1.3. Finally, the organization of this thesis will be presented in Section 1.4.

1.1 Conventional video coding

Throughout the last two decades, video coding technology has been developing continuously, resulting in a series of standards, including H.261 [h2693], H.263 [h2605] and the currently most popular H.264/AVC [h2610]. The focus was compression efficiency. And one thumb rule was that the compression ratio almost doubled from one generation of video coding standard to the next. More specifically, H.263 could reduce the video data amount two times more than H.261 at the same target video quality, and H.264/AVC two times more than H.263. Such a significant increase of the compression efficiency in the evolution of video coding standards has played an important role in modern visual communications, in which compressed video data are transmitted over bandwidth-limited Internet and wireless networks.

Although conventional video coding, today typically H.264/AVC, has a significantly high coding efficiency, the following two issues might limit its performance in some specific scenarios.

- Complex encoding: To achieve the highest possible coding efficiency, H.264/AVC entails a sophisticated process of predictive coding. For both intra-frame and inter-frame prediction, there are quite a few different prediction modes. Normally, all of these prediction modes need to be tried out in encoding in order that the one with the highest coding efficiency can be determined. And this process requires a high computational complexity. Therefore, the application of H.264/AVC can be limited in some specific scenarios, in which a simple encoder is desirable.

- Error propagation: In conventional video coding such as H.264/AVC, predictive coding is carried out. For example in a P-frame, a predictor for a macroblock (MB) is generated, typically based on one of the previously decoded frames, and this frame is referred to as reference frame. Then, this predictor is subtracted from the original MB and it is the resulting residual that is coded and transmitted to the decoder. To reconstruct this MB, the decoder needs to decode the residual correctly and reproduce the predictor perfectly. The reference frame at the decoder, however, might have ended up with some errors due to noisy transmission channels. In this instance, the predictor state reproduced at the decoder will be erroneous. Consequently, the reconstructed MB will also be erroneous, even if the decoded residual is correct. Principally speaking, errors in one frame can propagate to the next one, and this propagation continues till the next I-frame. Intuitively, errors that show up at some time will remain visible for a certain duration, resulting in a significant degradation of video quality. Above all, H.264/AVC is very sensitive to transmission errors in video communications.

1.2 Wyner-Ziv video coding

Wyner-Ziv video coding (WZVC), a fundamentally different video coding paradigm, is regarded as a candidate solution to solve the two issues of conventional video coding mentioned in Section 1.1. The two most significant advantages of WZVC compared to conventional video coding are low encoding complexity and inherent error resiliency. These two features will be introduced in Section 1.2.1 and Section 1.2.2.

1.2.1 Low encoding complexity

To understand why and how WZVC can have a low encoding complexity, it is necessary to review the fundamental theory of distributed source coding (DSC), including the Slepian-Wolf (SW) theorem [SW73] and the Wyner-Ziv (WZ) theorem [WZ76]. Slepian and Wolf contributed to the lossless DSC, while Wyner and Ziv to the lossy DSC. Given that video coding is lossy in most cases, Wyner-Ziv video coding is generally used as the term for video coding based on the DSC theory. In comparison, Slepian-Wolf coding, at any rate in this thesis, refers to the lossless coding of the quantization indices based on the DSC theory.

According to the DSC theory, the fundamental scenario for WZVC is source coding with side information at the receiver, as shown in Fig. 1.3. Herein, X and Y are two correlated signals. X is the one to code and Y is considered as the side information of X. The Slepian-Wolf theorem is applicable when X and Y are two discretely distributed signals, whereas the

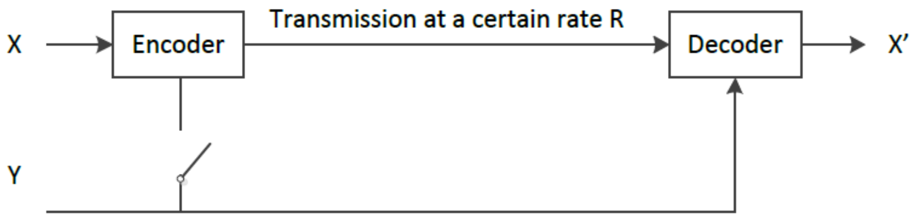


Figure 1.3: Source coding with side information at the receiver

Wyner-Ziv theorem is applicable when the two are continuously distributed. Conventionally, Y is available at both the transmitter and the receiver, and the coding of X can benefit from the side information Y, leading to a higher coding efficiency. In comparison, the Slepian-Wolf and Wyner-Ziv theorems suggest that the same coding efficiency can be achieved even if Y is only accessible at the receiver. The transmitter only needs to know the correlation of X and Y rather than the exact information of Y.

In the context of video coding, the side information Y in Fig 1.3 can be interpreted as a predictor for X, and the DSC theory means that the prediction, either the spatial prediction or the temporal prediction, can be moved from the encoder to the decoder. Particularly, the motion compensated prediction (MCP), which accounts for a significant part of the computational complexity of a conventional video encoder, can be performed at the Wyner-Ziv video decoder instead of at the Wyner-Ziv video encoder. Consequently, the encoding complexity is largely reduced. Therefore, WZVC is a candidate solution to specific applications, in which low encoding complexity is desirable. Typical applications include wireless video cameras, wireless low-power video surveillance, visual sensor networks, wireless capsule endoscopy and so on [PTG+08].

The low encoding complexity of WZVC, however, is achieved at the expense of a coding efficiency loss, because the MCP is significantly less efficient, if it is performed at the decoder rather than at the encoder. Therefore, the coding efficiency of WZVC is significantly worse when compared to that of conventional video coding, if the MCP is carried out only at the decoder. Nevertheless, motion compensated prediction can also be carried out at the Wyner-Ziv video encoder, and in this instance, the coding efficiency of WZVC can approach that of conventional video coding [PMR07].

1.2.2 Inherent error resiliency

A typical way of implementing DSC is to apply channel coding to source coding [BM01] and this prototype contributes to the inherent error resiliency of WZVC. An example with turbo coding is shown in Fig. 1.4. If the transmission channel is assumed to be error-free, only source coding is involved. An important concept is the so-called correlation channel assumed between the signal X and its side information Y. And the side information Y is generally interpreted as the sum of X and the noise over the correlation channel. The channel coding is used to resist the noise over the correlation channel and correct the difference between X and Y. In this manner, the signal X can be reconstructed. Moreover, if the real transmission channel is also noisy, a stronger protection of X using channel coding can resist the noise over both the correlation and transmission channel.

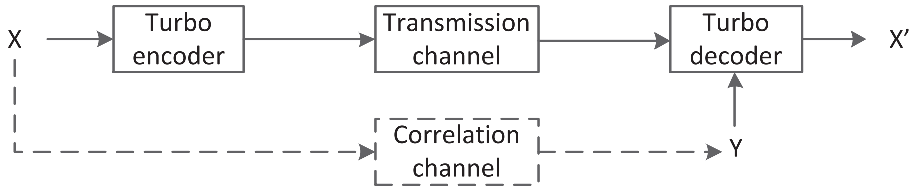


Figure 1.4: Implementation of distributed source coding using channel coding. In this instance, turbo coding is used.

Only the employment of channel coding for source coding in WZVC, however, does not guarantee that WZVC has a better error resiliency than conventional video coding, because channel coding can also be applied to conventionally coded video data to resist transmission errors. Yet the coding prototype of WZVC can attenuate error propagation and therefore the error resiliency of WZVC is superior to that of conventional video coding. For error resiliency, residual coding is not employed in WZVC. Instead, it is the original signal X that is coded and transmitted. When previously decoded frames are erroneous, these errors can also be reflected in the predictor, equivalently the side information, too. Fortunately, some or all of these errors can still be corrected if a stronger protection via channel coding is applied to X and thus will not propagate to the following video frames. Briefly speaking, if errors arise in some certain video frame, the error propagation can be attenuated simply by protecting the following frames in a stronger manner. In comparison, because residual coding is applied in conventional video coding, errors in the reference frame are very likely to propagate to the following frames, if there is no well-designed error-resilient scheme.

1.3 Major contributions

In this thesis, a Wyner-Ziv Bayer-pattern video communications system, as shown in Fig. 1.5, is designed, developed and optimized. This system begins with a novel color space transform, when Bayer-pattern video is converted to YUV video data. Next, the YUV video data are forwarded to a Wyner-Ziv video encoder. Then, the coded video stream is transmitted over an AWGN channel. Having received the transmitted video data, the receiver performs Wyner-Ziv video decoding and inverse color space transform. At this point, the Bayer-pattern video in the RGB domain has been reconstructed. Finally, demosaicing is carried out to obtain full color RGB video that can be correctly displayed. Research work of this thesis involves three aspects of this Wyner-Ziv Bayer-pattern video communications system and the research contributions are summarized as below.

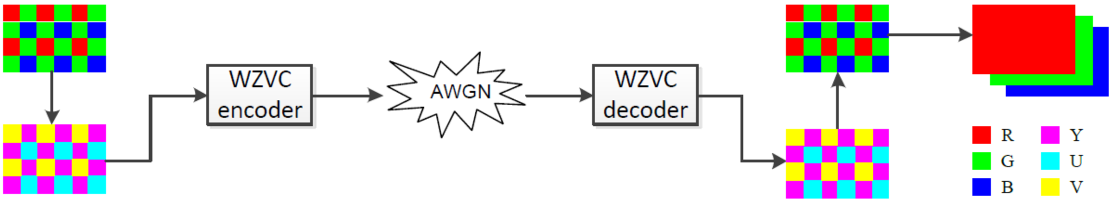


Figure 1.5: Wyner-Ziv Bayer-pattern video communications system

- **Wyner-Ziv video codec optimization:** To begin with, the turbo decoding, specifically the BCJR algorithm is optimized for WZVC, so that the coding efficiency can be improved. Moreover, bit-plane based WZVC using binary turbo codes and symbol based WZVC using trellis merging in turbo decoding are evaluated comparatively. Experiments are carried out to determine which scheme can lead to a better performance. In addition, a scheme is proposed to distribute the computation of motion estimation (ME) between encoder and decoder flexibly, which enables a good control of the encoding complexity, and the relationship between coding efficiency and encoding complexity is studied.

- **Error-resilient transmission:** A joint source-channel rate control scheme is proposed for error-resilient video transmission using WZVC. The noisy transmission channel is assumed to be AWGN. First, offline training is carried out to model the distribution of the sufficient coding rates for different AWGN channel states. Next, a method is proposed to estimate the mean end-to-end distortion of the reconstructed video. Finally, the joint source-channel rate control is carried out via adaptive quantization. And another specific issue addressed in this context of joint source-channel coding lies in whether to employ an additional channel coding scheme on top of WZVC to resist AWGN channel noise or to use the turbo codes embedded in the Wyner-Ziv video codec for both source coding and channel coding.

- **Bayer-pattern video coding:** The preprocessing step, specifically the color space transform (CST), is optimized specially for Bayer-pattern video. A novel CST scheme, which is referred to as B-4:2:0, is proposed and this scheme leads to a significantly higher video quality than the conventional scheme. Based upon the scheme B-4:2:0, a second CST scheme B-4:2:2 is proposed, and this scheme can reduce the encoding complexity by approximately 50% and thus can be used in a low complexity solution for Bayer-pattern video coding. Finally, the CST scheme B-4:2:2 and the Wyner-Ziv video codec are integrated into a Wyner-Ziv Bayer-pattern video compression system, and this system ends up with very low encoding complexity.

1.4 Organization

The remaining part of this thesis is organized as follows. In Chapter 2, the optimization of the Wyner-Ziv video codec is presented. To begin with, the structure of the codec is introduced. Next, the optimization of turbo decoding for WZVC is presented. Then, a framework is proposed for comparing the performance of bit-plane and symbol based coding. After this, it is discussed how motion estimation and compensation (MEC) at the encoder is made scalable to trade off encoding complexity and coding efficiency.

Chapter 3 refers to error-resilient video transmission using WZVC and this problem is formulated as a joint source-channel rate control problem. After a literature review, the challenges and assumptions of the specific scenario, namely real-time video transmission over a wireless error-prone channel are presented. Then, two variants of WZVC-based joint source-channel coding system, namely WZVC and WZVC+FEC, are presented and evaluated comparatively. Later, the joint source-channel rate control algorithm is described.

In Chapter 4, lossy compression of Bayer-pattern video using Wyner-Ziv video coding is investigated. Firstly, background knowledge of Bayer-pattern images as well as state-of-the-art demosaicing and compression schemes are introduced. Secondly, two variants of the proposed color space transform methods, namely B-4:2:0 and B-4:2:2 are presented. Thirdly, H.264/AVC based Bayer-pattern video coding systems using the proposed color space transform methods are shown. Finally, the Wyner-Ziv Bayer-pattern video coding system is presented.

Chapter 5 concludes this thesis. Major results are summarized and conclusions are drawn. Moreover, a couple of topics for future research are discussed.

Parts of this thesis work have been published in [CSC11a, CSC11b, RCS10, CS09, CS07b, CS07a, CSS09, CS08].

Chapter 2

Codec optimization

In this chapter, the optimization of the Wyner-Ziv video codec is presented. To begin with, the structure of the codec is introduced in Section 2.1. The components most relevant to this thesis, including mode selection, quantization, turbo codes, correlation channel modeling and rate control, are presented in detail.

Later, the major contributions of this thesis concerning optimizing the codec and extending its functionality are discussed. In Section 2.2, the optimization of turbo decoding is presented. In Section 2.3, a framework is proposed for comparing the error resiliency of bit-plane and symbol based WZVC in a noisy video transmission environment. In Section 2.4, it is discussed how motion estimation at the encoder is made scalable to trade off encoding complexity and coding efficiency.

Finally, all the functional components of the Wyner-Ziv video codec are summarized in two optional configurations. One of them aims at the low encoding complexity for source coding whereas the other targets thes error resiliency in noisy video transmission environments.

2.1 Codec structure

Initially, a Wyner-Ziv video codec, more specifically a pixel-domain Wyner-Ziv video codec, was developed based on the work in [GARRM05]. Later on, the functionality of the codec was extended based on [PMR07, XX06, AAD$^+$07]. The ultimate pixel-domain Wyner-Ziv video codec structure is shown in Fig. 2.1. Herein, video frames are classified into Intraframe (I-frame) and Wyner-Ziv frame (WZ-frame). The I-frame is identical to the one standardized in H.264/AVC [h2610] and thus an H.264/AVC encoder and decoder are deployed to code an I-frame. Any frame between two consecutive I-frames is a WZ-frame. WZ-frames are coded according to the DSC theory, more specificlaly the principle of source coding with side information at the decoder [SW73, WZ76].

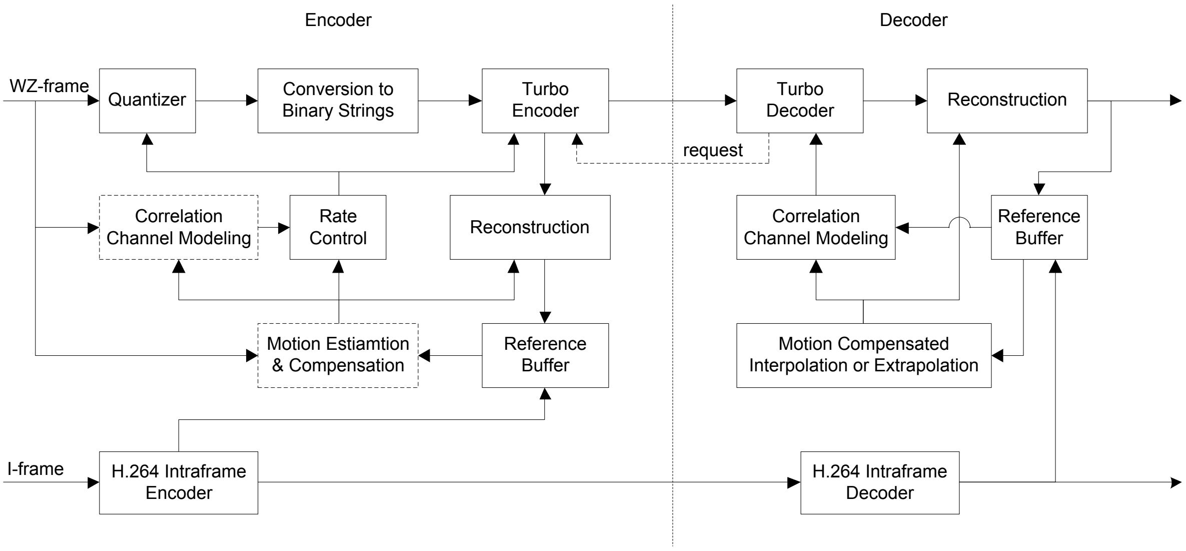


Figure 2.1: Pixel-domain Wyner-Ziv video coding using turbo codes

The encoding of a WZ-frame consists of three major steps. To begin with, image pixels are quantized. Next, the quantization indices are converted to binary strings. Finally, the turbo encoder generates parity bits and transmits part of them to the decoder. In addition, the component of rate control determines how strongly image pixels are quantized and how many parity bits the turbo encoder transmits to the decoder. There is much work on rate control [RCS10, MRPN$^+$08, AT05a, MFEK$^+$08, CWO08, BP07, SHG$^+$08].

At the decoder, previously decoded I-frames and WZ-frames, accessible in the reference buffer, are used to predict the current WZ-frame by motion compensated interpolation (MCI) [TA97, ABP05] or motion compensated extrapolation (MCE) [SK11a, SK11b, SK10]. The predicted WZ-frame is often called side information in the context of WZVC and can be refined iteratively during the decoding [AFWA06, AT05b]. The component of correlation channel modeling [BAP06, WGL06, BP10, PLC$^+$09, ZXSY09, WBGL06] estimates the correlation between the original WZ-frame and the side information based on the previously decoded WZ-frames. The turbo decoder makes use of this correlation model together with the received parity bits to recover the original quantization indices. Finally, the original pixel intensities are reconstructed using the quantization indices and the side information [GARRM05].

Motion estimation and compensation (MEC) is optional at the encoder. This component, due to its high computational complexity, is left out, if low encoding complexity is desirable. In this instance, the temporal correlation between successive video frames is exploited by motion compensated interpolation or extrapolation at the decoder. In some applications, however, the encoder is not complexity constrained and error resiliency is desired, MEC is activated at the encoder for higher coding efficiency. This component predicts a WZ-frame more precisely than motion compensated interpolation or extrapolation at the decoder, because the original

WZ-frame is available at the encoder. The resulting motion vectors are coded and transmitted to the decoder for generating precise side information.

Similarly, correlation channel modeling, another optional component at the encoder, is also more precise than its counterpart at the decoder, because the original WZ-frame can be exploited at the encoder. Parameters of the correlation channel are quantized to discrete levels [PLC$^+$09], coded and transmitted to the decoder.

So far, the structure of the Wyner-Ziv video codec as well as the coding process have been introduced briefly. In the following sections, five components most relevant to this thesis are presented in detail.

2.1.1 Mode selection

Macroblocks (MB), 16 by 16 pixel arrays, are the unit for mode selection. Each MB ends up with one of the three optional modes: intra mode, skip mode and WZ mode. In the following, the MBs in these three modes are called intra MBs, skip MBs and WZ MBs, respectively. The intra mode is developed based on the work in [LHJ$^+$08] while the skip mode is implemented based on the work in [HC08]. Similar to conventional video coding like H.264/AVC, an arbitrary number of intra and skip MBs in a video frame can be put together in a video slice and each slice is coded independently from other slices.

In the remaining part of this chapter, the focus will be on WZ MBs and the codec structure in Fig. 2.1 mainly refers to WZ MBs. WZ MBs are also arranged in slices. For the convenience of simulation, a WZ slice typically has eight WZ MBs in it. Moreover, in the slices for WZ MBs, however, intra and skip MBs are not allowed to be mixed in the codec used in this thesis, because Wyner-Ziv video coding is fundamentally different from conventional video coding. Nevertheless, it is possible in general to mix up MBs in different modes in a single slice.

2.1.2 Quantization

For WZ MBs, the slice is the unit for quantization. The same quantizer is applied to all WZ MBs in a slice and the quantizer can be different from slice to slice.

Four uniform scalar quantizers are supported to trade off rate and distortion: a one-bit, a two-bit, a three-bit and a four-bit one. One of the four optional quantizers needs to be chosen for each slice. Each quantizer leads to a different number of bit-planes coded for a slice. And the symbol N_{BP} is used to represent the number of bit-planes coded and identify the chosen quantizer. The value of N_{BP} can be 1, 2, 3 or 4. For instance, $N_{BP} = 1$ represents the one-bit quantizer and means that the first bit-plane, equivalently the most significant bit-plane (MSB) is coded. Similarly, $N_{BP} = 2$ represents the two-bit quantizer and means that the first two bit-planes are coded, and so on.

Of course, a five-bit, a six-bit, a seven-bit and an eight-bit quantizer should also be supported in general, because the pixel intensity of the test videos in this thesis has an eight-bit precision. These four quantizers, however, lead to very high quality of the reconstructed video, often beyond 40 dB in PSNR, which means near lossless video coding. Only lossy video coding is discussed in this thesis, and the target video quality ranges from 30 dB to 40 dB in PSNR. Therefore, these four quantizers associated with very high quality are rarely chosen in the experiments and thus excluded from the codec for simplicity.

2.1.3 Turbo codes

The SW coding is implemented using rate compatible punctured turbo codes (RCPTC), specifically the “Encoder C” in [RM00]. The mother code is of rate 1/3 and its generators are $(1, 33/31)_{\text{octal}} + (33/31)_{\text{octal}}$ and is punctured to 32 different sub rates for adaptation to the correlation and the transmission channel. The rate of 1/3 here accords with the conventional definition of rates of channel codes. This definition is inconvenient for SW coding, because SW coding turns out to be source coding, although channel coding is employed. Unfortunately, there has been no standard definition of rates for SW coding. In order that possible ambiguity of the rate definition is avoided, the rate R of the SW codes implemented using turbo codes in this thesis is defined in (2.1)

$$R = \frac{\text{the number of transmitted parity bits}}{\text{the number of information bits}}. \tag{2.1}$$

The mother code ends up with two parity bits for every information bit. Given that the puncturing period is 16, every 16 information bits result in at most 32 parity bits. These 32 parity bits underlie the 32 different rates of the SW codes. The minimum rate, for instance, is 1/16, which means that a single parity bit is transmitted for every 16 information bits. The source coding rate ranges from $1/16$ to $16/16 = 1$ with an increment of 1/16. If a noisy transmission channel is considered, higher rates are necessary to introduce redundancy and resist channel errors. In this circumstance, the rate could be larger than one, ranging from $17/16$ to $32/16 = 2$. Even higher rates could come out, if one or more of the 32 parity bits are transmitted more than once. Nevertheless, such high rates have never been called upon in the work presented in this thesis, so they will not be considered.

The turbo codes presented above are binary codes and are used for bit-plane based coding, referred to as WZVC-B. WZVC-B will be presented in Section 2.3.3. As a second option, symbol based coding is also implemented in the Wyner-Ziv video codec using turbo codes. WZVC using symbol based turbo codes is referred to as WZVC-S and will be presented in Section 2.3.2. In this instance, the encoding is carried out using the same binary turbo codes. Therefore, the definition of the rate remains the same. The decoding, however, is implemented in a symbol-based manner by trellis merging [ASW08, Ber98]. Symbol and bit-plane based coding are presented in detail and compared with each other in Section 2.3.

Although turbo codes are capable of correcting errors, they cannot detect errors alone. Therefore, cyclic redundancy check (CRC) is used to detect errors. At the encoder, the CRC bits are generated for the original information bits. When it comes to symbol based coding, the information symbols are converted to information bits before CRC bits can be generated. The CRC bits are transmitted to the decoder and used to check the correctness of its output. More specifically, some CRC bits are generated for the information bits or those converted from the information symbols output from the decoder and compared with the received CRC bits. If they are identical, the decoded result is supposed to be error-free and the decoding is regarded as successful; otherwise, the decoder output must be erroneous and thus the decoding fails.

2.1.4 Correlation channel modeling

The correlation channel model describes the statistical property of the difference between the original image data and their side information. This model is used for two purposes in this thesis. A basic usage lies in calculating the L-values of information bits which are required in turbo decoding. This usage is discussed in Section 2.2.2. Additionally, the correlation channel model is needed for calculating the conditional entropy of quantization indices given the side information, so that an accurate rate control can be achieved. How this conditional entropy is calculated based on the correlation model is presented in Section 3.3.1.

Among quite a few models for the correlation channel [BAP06, WGL06, BP10, PLC$^+$09, ZXSY09, WBGL06], the generalized Gaussian distribution with a shape parameter of 0.5 proves to have the best overall performance [WBGL06] and thus is used in this thesis. The correlation channel model basically appears in the form of the conditional probability density function (PDF) of a pixel intensity X given its side information Y and the pdf is formulated in the following:

$$f_{X|Y}(x|y) = \frac{1}{4\alpha} e^{-(\frac{|x-y|}{\alpha})^{0.5}}, \quad x \in (-\infty, \infty) \text{ and } y = 0, 1, 2, 3, ..., 255, \tag{2.2}$$

where $\alpha^2 = \sigma_N^2/120$ and $N = X - Y$ represents the correlation noise.

The only parameter to determine is α, which depends on σ_N^2, the variance of the correlation noise N. Because a video signal is in principle a non-stationary process, the correlation noise changes with time. Therefore, α is estimated for each individual WZ slice. Subtracting side information from original pixels in a video slice results in samples of correlation noise N, and the sample variance is calculated and used as an estimator for the statistical variance σ_N^2.

The PDF in (2.2) cannot be used in practice, because pixel intensities are discretely distributed, and more specifically integers ranging from 0 to 255. Therefore, the conditional PDF is converted to a conditional probability mass function (PMF) of image data given their side information in the following way:

$$p_{X|Y}(x|y) = \frac{\int_x^{x+1} f_{X|Y}(u|y)\,du}{\int_0^{256} f_{X|Y}(u|y)\,du}, \quad x, y = 0, 1, 2, 3, ..., 255. \tag{2.3}$$

The numerator of (2.3) is the integral of the conditional PDF in an interval of one and thus represents the conditional probability of the event that $X = x$ given $Y = y$. Because X is distributed between 0 and 255, the PDF outside this range will be neglected. In this instance, a normalization factor needs to be introduced, such that the basic PMF property $\sum_{x=0}^{255} p_{X|Y}(x|y) = 1$ is satisfied. The denominator of (2.3) plays the role of this normalization factor.

2.1.5 Rate control

The purpose of rate control in conventional video coding is to make the bit rate of the compressed video as close as possible to a certain target bit rate. And rate control is implemented by adaptive quantization. This purpose and this approach also hold for the rate control for WZVC.

An additional requirement for the rate control for WZVC lies in the fact that the sufficient rate needs to be estimated. In the context of WZVC, the sufficient source rate is defined as the minimum bit rate required for successful decoding. An accurate estimation of this rate at the encoder plays a significant role in achieving the optimal rate distortion performance. On one hand, the decoding fails and the video quality drops, if the encoder ends up with a rate lower than the sufficient source rate. On the other hand, overestimating the sufficient source rate means that the encoder transmits more bits than necessary and thus the rate-distortion performance is also away from its optimum.

Above all, rate control is a significant issue in WZVC. And when it comes to joint source-channel coding, the rate control is more important and more complicated. The state of the art for joint source-channel coding as well as the contribution of this thesis in joint source-channel rate control will be discussed in detail in Chapter 3.

2.2 Turbo decoding optimization

When turbo codes, originally designed for channel coding, are used for DSC, a significant issue lies in the role change of the parity bits. In the context of channel coding, the parity bits are transmitted along with the information bits over a noisy channel and thus undergo corruption to the same degree as the information bits. In the context of DSC, however, only the parity bits are transmitted to the decoder. Herein, an error robust transmission can be guaranteed by error resiliency mechanisms at lower layers, such as the channel coding at the physical layer and the retransmission at the transport layer, and the parity bits are assumed to be error-free. Therefore, a central issue in turbo decoding lies in how to make most of the error-free parity bits to improve the decoding performance.

There has been some relevant work in the literature. Bajcsy and Mitran [BM01] addressed the SW problem using turbo codes. Aaron and Girod did a similar work independently but with different turbo codes [AG02]. These two publications are the very initial work on SW coding using turbo codes. How to take advantage of the error-free parity bits, however, was explicitly addressed in none of them. Later, Hua and Chen studied both convolutional codes and turbo codes and proposed an improved Viterbi algorithm for decoding by making good use of the error-free parity bits [HC05].

Apart from turbo codes, low-density parity-check (LDPC) codes have also been used to implement SW coding. Liveris et al. used fix-rate LDPC codes and achieved a compression ratio close to the SW bound. Varodayan et al. proposed rate-adaptive LDPC Accumulate (LDPCA) codes and Sum LDPC Accumulate (SLDPCA) codes with flexible puncturing-based rate adaptation to source statistics [VAG05]. According to their experimental results, turbo codes are significantly less efficient than LDPCA and SLDPCA at medium or high rates, but compare to or even outperform the two at low rates. Given that low rates are predominant in video coding due to increasingly accurate prediction, it is still worth investigating turbo codes.

The contribution for turbo coding in this thesis lies in making most of error-free parity bits to improve the turbo decoding performance. And this is an extension of the work in [HC05] in two aspects. First, the BCJR decoding algorithm, generally more efficient than the Viterbi algorithm, is discussed. Second, only the calculation of L-values for parity bits needs to be updated and thus the interface and the internal structure of the decoder can remain the same.

In Section 2.2.1, the aspects of the BCJR decoding algorithm relevant to DSC are introduced. Next, some typical ways of calculating L-values for information bits are presented in Section 2.2.2. Then, in Section 2.2.3, the proposed approach to exploiting the error-free parity bits in turbo decoding using the BCJR algorithm is discussed in detail. Finally, experimental results are presented and analyzed in Section 2.2.4.

2.2.1 BCJR algorithm

The BCJR algorithm [LC04] is a decoding algorithm that can be applied to convolutional codes. The decoding performance using the BCJR algorithm fundamentally determines the turbo decoding performance, because a turbo coder consists of two convolutional coders in parallel. Therefore, it is sufficient to focus on the BCJR algorithm when it comes to the optimization of turbo decoding for DSC. Major notations for presenting the BCJR algorithm are listed up in Table 2.1.

Different from the Viterbi algorithm, which minimizes the word error rate (WER), the

Table 2.1: Notations for presenting the BCJR decoding algorithm

K	the total number of the information bits, including the tail bits
(s', s)	a transition from state s' to state s, which is equivalently a trellis branch
u_l	the original lth information bit
x_l	the lth information bit of the side information for decoding
$\mathbf{y}_l$	the received parity bit(s) associated with the lth information bit
$\mathbf{r}_l$	$\mathbf{r}_l = (x_l, \mathbf{y}_l)$
v_l^x	the information bit on the trellis branch (s', s) at time l
$\mathbf{v}_l^y$	the parity bit(s) on the trellis branch (s', s) at time l
$\mathbf{v}_l(s', s)$	$\mathbf{v}_l(s', s) = (v_l^x, \mathbf{v}_l^y)$
Σ_l	the set of all the states at time l
Σ_l^{-1}	the set of all the branches, on which $v_l^x = -1$, at time l
Σ_l^{+1}	the set of all the branches, on which $v_l^x = +1$, at time l
Σ_l	the set of all the branches at time l, $\Sigma_l = \Sigma_l^{-1} \cup \Sigma_l^{+1}$
$L(v_l^x)$	the log likelihood of the information bit on (s', s) at time l
$L_a(u_l)$	a priori log likelihood ratio for the information bit u_l
L_c	transmission channel reliability factor

BCJR algorithm minimizes the bit error rate (BER) and thus leads to a better decoding performance particularly in the following two circumstances. First, the BCJR algorithm outperforms the Viterbi algorithm when the information bits are not equally likely. Second, the BCJR algorithm gives the best performance if iterative decoding is employed and the *a priori* information changes from iteration to iteration, and this is what happens in turbo decoding.

The BCJR decoding algorithm is a trellis-based one and examples of trellis diagrams are available in [LC04]. Among several variants, the log-MAP algorithm, or the log domain BCJR algorithm, is the most popular one. For one thing, the log-MAP algorithm is easier to implement; for another, it provides higher numerical stability. The following log domain metrics play a central role in the decoding process:

$$\gamma_l^*(s', s) = \frac{v_l^x L_a(u_l)}{2} + L(v_l^x) + \frac{L_c}{2}\mathbf{y}_l \cdot \mathbf{v}_l^y, \quad l = 0, 1, \cdots, K-1, \tag{2.4}$$

$$\alpha_{l+1}^*(s) = max_{s' \in \sigma_l}^*[\gamma_l^*(s', s) + \alpha_l^*(s')], \quad l = 0, 1, \cdots, K-1, \tag{2.5}$$

$$\alpha_0^*(s) = \begin{cases} 0, & s = \mathbf{0} \\ -\infty, & s \neq \mathbf{0}, \end{cases} \tag{2.6}$$

$$\beta_l^*(s') = max_{s \in \sigma_{l+1}}^*[\gamma_l^*(s', s) + \beta_{l+1}^*(s)], \quad l = K-1, K-2, \cdots, 0, \tag{2.7}$$

$$\beta_K^*(s) = \begin{cases} 0, & s = \mathbf{0} \\ -\infty, & s \neq \mathbf{0}, \end{cases} \tag{2.8}$$

Once γ^* is determined, α^* and β^* can be calculated recursively using the max^* function.

The definition of the max^* function is in the following:

$$\begin{aligned} max^*(x, y) &= ln(e^x + e^y) = max(x, y) + ln(1 + e^{-|x-y|}), && (2.9) \\ max^*(x, y, z) &= ln(e^x + e^y + e^z) = max^*[max^*(x, y), z]. && (2.10) \end{aligned}$$

Finally, the a posteriori L-value of each individual information bit is calculated and can be expressed as

$$L(u_l) = max^*_{(s',s)\in\Sigma_l^+}[\beta^*_{l+1}(s)] + \gamma^*_l(s', s) + \alpha^*_l(s')] - max^*_{(s',s)\in\Sigma_l^-}[\beta^*_{l+1}(s)] + \gamma^*_l(s', s) + \alpha^*_l(s')], \tag{2.11}$$

and the decision on the original information bit u_l is made using the following criterion:

$$\hat{u}_l = \begin{cases} +1, & L(u_l) >= 0; \\ -1, & L(u_l) < 0. \end{cases} \tag{2.12}$$

One thing worth mentioning is why and how binary bits 0 and 1 are mapped to +1 and -1 for u_l. Originally, u_l is supposed to represent a binary information bit, which is either 0 or 1 in binary representation. When it comes to soft-value based BCJR decoding, the formulation of the algorithm and the numerical calculation are both more convenient, if 0 is mapped to $+1$ and 1 is mapped to -1, or if 0 is mapped to -1 and 1 is mapped to $+1$. The two ways of mapping are equivalent and do not make a difference in the decoded binary strings. In this thesis, the former way of mapping is taken.

2.2.2 Log domain metrics for information bits

The first two terms in (2.4) refer to the log domain metrics for information bits. The term $\frac{v_l^x L_a(u_l)}{2}$ represents the a priori log likelihood of the information bits. At the initial stage of the turbo decoding, this item equals zero, because +1 and -1 are assumed to be equally likely for an information bit. During the iterative decoding, the extrinsic information of one BCJR decoder is extracted and forwarded to the other decoder as a priori log likelihood. This practice is standard in turbo decoding.

The term $L(v_l^x)$ refers to the log likelihood of the information bit on the trellis branch (s', s) given the side information at the decoder. Calculation of this metric depends on how the correlation channel between the original image data and their side information is modeled. The generalized Gaussian distribution with a shape parameter of 0.5 proves to have the best overall performance [WBGL06] and thus is used in the Wyner-Ziv video codec to model the correlation channel. Once the correlation channel model is determined, there is a common method for calculating $L(v_l^x)$. Relevant discussions are available in [HM08, YFPP08].

2.2.3 Log domain metrics for parity bits

The log domain metrics for parity bits refer to the term $\frac{L_c}{2}\mathbf{y}_l \cdot \mathbf{v}_l^y$ in (2.4). Above all, the transmission channel reliability factor L_c needs to be determined. If the parity bits are transmitted over a noisy channel, for instance an AWGN channel, there is a standard way of calculating L_c. In comparison, if the discussion is confined to source coding and parity bits are assumed to be error-free, the transmission channel is completely reliable and thus L_c becomes infinity. Unfortunately, infinite values would not work out in a practical decoding algorithm. Therefore, the focus of this section is how to determine a reasonable value for L_c, which is finite but sufficiently large for achieving the optimum decoding performance.

Initially, an approximation of the max^* function needs to be introduced to simplify the derivation of a lower bound for L_c. The correction term $ln(1+e^{-|x-y|})$ in (2.9) is ignored and the approximation

$$max^*(x,y) \approx max(x,y) \tag{2.13}$$

is used instead. “The approximation is reasonably good whenever $max(x,y) \geq 7$.” [LC04]. Now that L_c is very large and thus γ^* in (2.5), (2.7) and (2.11) is much larger than 7, this approximation in (2.13) can be introduced. Under this approximation and after (2.4) is plugged in, (2.11) becomes

$$\begin{aligned} L(u_l) \quad = \quad & max_{(s',s)\in\Sigma_l^+}[\beta^*_{l+1}(s)] + \tfrac{v_l^x L_a(u_l)}{2} + L(v_l^x) + \tfrac{L_c}{2}\mathbf{y}_l \cdot \mathbf{v}_l^y + \alpha_l^*(s')] \\ & -max_{(s',s)\in\Sigma_l^-}[\beta^*_{l+1}(s)] + \tfrac{v_l^x L_a(u_l)}{2} + L(v_l^x) + \tfrac{L_c}{2}\mathbf{y}_l \cdot \mathbf{v}_l^y + \alpha_l^*(s')]. \end{aligned} \tag{2.14}$$

The first term on the right side of (2.14) suggests that decoding of the lth information bit involves selecting the trellis branch of the highest branch metric out of all the metric branches associated with an information bit +1. Similarly, the second term means selecting the trellis branch of the highest branch metric out of all the branches associated with an information bit -1. Ideally, the parity bits on the selected trellis branch should be identical to the received parity bits, because received parity bits are error-free. And this is just the proposed criterion to make sure that parity bits play a dominant role in decoding. A lower bound of L_c can be determined if this criterion is applied to the worst case in the BCJR decoding.

When decoding the lth information bit, the worst case consists of two aspects. First, all the information bits in the side information are erroneous. Second, all the parity bits before and after are punctured and there is only one parity bit on the lth trellis section. In this worst case, the decoding of the information bit on the lth trellis section is very difficult, and the best that the decoder can do is to select one of the trellis branches with a correct parity bit on them. Equivalently, the decoder can avoid all the trellis branches with a wrong parity bit in spite of all the information bit errors. This objective is ensured if the following condition is satisfied:

$$L_c > max_{(s',s)\in\Sigma_l}|\beta^*_{l+1}(s) + \frac{v_l^x L_a(u_l)}{2} + L(v_l^x) + \alpha_l^*(s')| \tag{2.15}$$

If the condition in (2.15) holds, all the trellis branches with a correct parity bit must end up with a positive a posteriori L-value, whereas those with an incorrect parity bit have a negative a posteriori L-value. Therefore, it is always true that the decoder will select a trellis branch with a correct parity bit ultimately.

In order that the derivation goes on, the right part of (2.15) needs to be upper bounded as follows:

$$\begin{array}{ll} & max_{(s',s)\in\Sigma_l}|\beta^*_{l+1}(s)+\frac{v^x_l L_a(u_l)}{2}+L(v^x_l)+\alpha^*_l(s')| \\ \leq & max_{s\in\sigma_{l+1}}|\beta^*_{l+1}(s)|+max_{(s',s)\in\Sigma_l}|\frac{v^x_l L_a(u_l)}{2}+L(v^x_l)|+max_{s'\in\sigma_l}|\alpha^*_l(s')| \end{array} \tag{2.16}$$

According to (2.5), (2.4) and the approximation in (2.13), $max_{(s',s)\in\Sigma_l}|\alpha^*_l(s')$ in (2.16) can be represented and upper bounded as follows:

$$\begin{array}{rcl} max_{s'\in\sigma_l}|\alpha^*_l(s')| & = & max_{s'\in\sigma_l}|max_{s''\in\sigma_{l-1}}[\gamma^*_{l-1}(s'',s')+\alpha^*_{l-1}(s'')]| \\ & \leq & max_{(s'',s')\in\Sigma_{l-1}}|[\gamma^*_{l-1}(s'',s')]|+max_{s''\in\sigma_{l-1}}|[\alpha^*_{l-1}(s'')]| \\ & \leq & max_{(s'',s')\in\Sigma_{l-1}}|\frac{u_{l-1}L_a(u_{l-1})}{2}+L^x_{l-1}|+max_{s''\in\sigma_{l-1}}|[\alpha^*_{l-1}(s'')]| \end{array} \tag{2.17}$$

The term $max_{s''\in\sigma_{l-1}}|[\alpha^*_{l-1}(s'')]|$ on the right side of (2.17) turns out to be a time shifted version of the term $max_{s'\in\sigma_l}|\alpha^*_l(s')|$ on the left side. Hence, applying (2.17) iteratively leads to the following result:

$$max_{s'\in\sigma_l}|\alpha^*_l(s')| \leq \sum_{i=0}^{l-1}|\frac{v^x_i L_a(u_i)}{2}+L(v^x_l)| \tag{2.18}$$

The $max_{s\in\sigma_{l+1}}|\beta^*_{l+1}(s)|$ in (2.16) can be processed in a similar manner and its upper bound is directly formulated in the following:

$$max_{s\in\sigma_{l+1}}|\beta^*_{l+1}(s)| \leq \sum_{i=l+1}^{K-1} max_{(s',s)\in\Sigma_i}|\frac{v^x_i L_a(u_i)}{2}+L(v^x_l)| \tag{2.19}$$

Plugging (2.18) and (2.19) into (2.16) and taking (2.15) into account leads to an ultimate lower bound for L_c:

$$L_c > \sum_{l=0}^{K-1} max_{(s',s)\in\Sigma_l}|\frac{v^x_l L_a(u_l)}{2}+L(v^x_l)|, \tag{2.20}$$

and this is a sufficient condition for achieving the optimal BCJR decoding performance with error-free parity bits.

The branch metric calculation for a parity bit according to the criterion in 2.20 consists of two steps. First, the maximum absolute value of information bit of all the branches for a trellis section is calculated. Next, the sum of the maximum for every trellis section is calculated and this value is just the absolute value of every parity bit. And the criterion in 2.20 is referred to as the "sum_max" criterion.

The "sum_max" criterion has two advantages. First, it makes most of the error-free parity bits for turbo decoding and thus the resulting source coding rate is closer to the SW bound. Additionally, this criterion involves only the branch metric calculation for parity bits and thus the interface as well as the internal algorithms of the standard BCJR decoder remain the same. Therefore, the practical application of the "sum_max" criterion is convenient.

One limitation of the "sum_max" criterion, however, lies in the significantly increased value range of parity bit L-values. The L-value of a parity based on this criterion is linearly proportional to the block length for turbo coding. This means that the turbo decoder needs to deal with larger numbers, which might require higher computational power and bring about challenges in hardware implementation for turbo decoding.

To manifest that the "sum_max" criterion does improve the turbo decoding performance by assigning sufficiently large L-values to error-free parity bits, it is necessary to look at the decoding performance when the L-values of parity bits are not large enough. For this purpose, an alternative criterion for calculating L-values for parity bits, referred to as the "max" criterion in this thesis, is formulated in the following:

$$L_c(l) > max_{(s',s)\in\Sigma_l}|\frac{v_l^x L_a(u_l)}{2} + L(v_l^x)|. \tag{2.21}$$

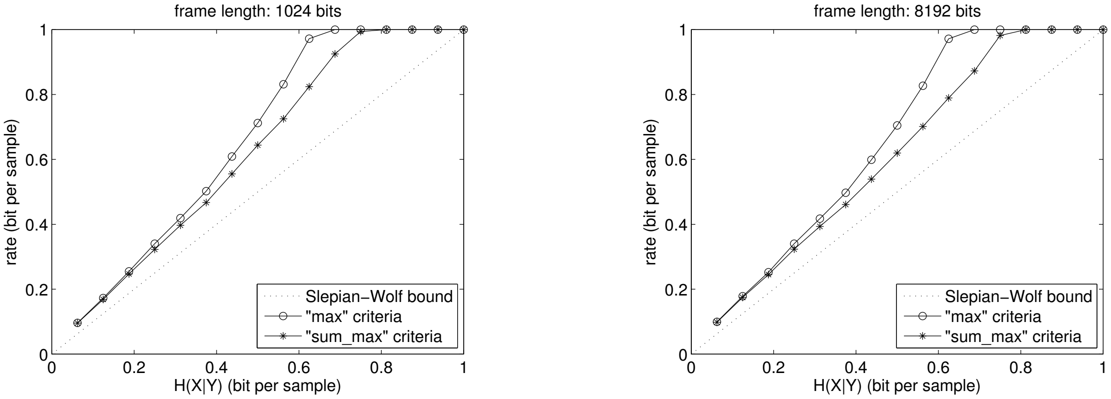
The "max" criterion ensures that the absolute L-value of a parity bit at the lth trellis section is larger than the absolute L-value of all the information bits at the same trellis section. Intuitively speaking, in the turbo decoding, a parity bit plays a dominant role in its local trellis section, but it is unable to resist the possible errors of the information bits in previous and following trellis sections. Compared with the "sum_max" criterion, the "max" criterion does not require increasing the L-value of parity bits with the block length for turbo coding, but the L-values of parity bits might be not sufficiently large, when it comes to a considerable number of information bit errors. Therefore, the "max" criterion is sub-optimal in turbo decoding performance. And this criterion is taken as a comparison scheme for the "sum_max" criterion.

2.2.4 Experimental results

Two experiments have been carried out to manifest that the "sum_max" criterion leads to a significant improvement in SW coding compared with the "max" criterion. The first experiment was based on random binary sources generated by a computer, while the second was based on videos. The turbo codes used in the experiments are presented in Section 2.1.3.

In the first experiment, the correlation channel between the source X and its side information Y is assumed to be a binary symmetric channel (BSC). A frame of information bits for X and an equal number of bits for Y are generated randomly. The crossover probability or the BSC between X and Y is controlled properly to obtain a certain conditional entropy $H(X|Y)$. The decoding is carried out for the "sum_max" criterion and the "max" criterion. The results are presented in Fig. 2.2. Each point in this figure is an average of 1000 trials.

Fig. 2.2(a) shows the turbo decoding performance for a short frame length of 1024 information bits while Fig. 2.2(b) for a longer frame length of 8192 information bits. Both figures



(a) SW coding for a frame length of 1024 information bits

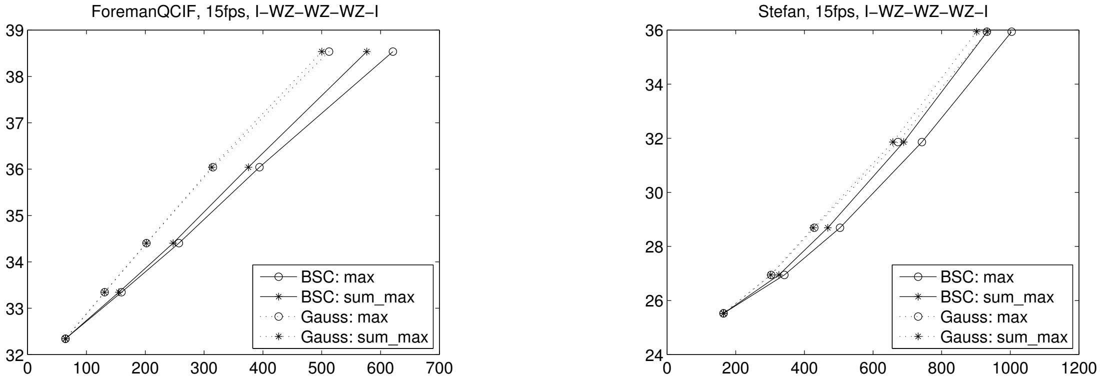
(b) SW coding for a frame length of 8192 information bits

Figure 2.2: Performance of SW coding using turbo codes for binary data

suggest that the decoding based on the "sum_max" criterion outperforms that based on the "max" criterion. Especially at high rates, the rate of the former one is significantly lower than the rate of the latter one and closer to the SW bound. At lower rates, however, the "sum_max" criterion and the "max" criterion have similar performance, because the parity bits are so sparsely distributed over the trellis that their influence in decoding dies away quickly after going through a few trellis branches, even if they have large absolute values in the "sum_max" criterion.

The second experiment involves video coding. Again, both criteria are applied to the decoding for comparison. Two correlation channel models, namely binary symmetric channel (BSC) and generalized Gaussian channel with a shape parameter of 0.5, are used in the simulation. The two test video sequences used in this experiment are the Foreman sequence and the Stefan sequence at a frame rate of 15 frames per second (fps). The GOP structure is configured to be I-WZ-WZ-WZ-I, and this configuration means that every fourth frame is an I-frame and there are three WZ-frames between two I-frames. Because the focus here is on the performance comparison of two variants of turbo decoding, it is simply assumed that an accurate correlation channel model is available. For this purpose, motion estimation (ME) with integer-pel accuracy is carried out at the encoder to generate the side information. Then, correlation channel parameters are computed based on this side information and signaled to the decoder.

The rate-distortion (RD) curves for the Foreman sequence are shown in Fig. 2.3(a) and those for the Stefan sequence in Fig. 2.3(b). The turbo decoding based on the "sum_max" criterion (BSC: sum_max) reduces the bit rate by up to 10% compared with that based on the "max" criterion (BSC: max), if the correlation channel is modeled as BSC. However, the two criteria (Gauss: sum_max and Gauss: max) perform equally well, if the correlation channel



(a) RD curves for the Foreman sequence

(b) RD curves for the Stefan sequence

Figure 2.3: Performance of SW coding using turbo codes for video sequences

is modeled as generalized Gaussian channel with a shape parameter of 0.5. The reason is that generalized Gaussian channel is a more sophisticated correlation channel model and thus leads to lower coding rates. According to the results in Fig. 2.2, there is hardly any difference between the two criteria at lower rates ranging from 0 to 0.3.

Above all, the results for both binary and video data agree on the conclusion that the "sum_max" criterion and the "max" criterion for turbo decoding perform equally well at lower rates, whereas the former one outperforms the latter one significantly at higher rates.

2.3 Symbol and bit-plane based coding

All the typical Wyner-Ziv video codecs [GARRM05, PMR07, AAD$^+$07] employ binary channel codes to implement bit-plane based WZVC (WZVC-B). An alternative, however, is to use symbol based coding (WZVC-S) with non-binary channel codes. Westerlaken et al. [WBGL07] analyzed theoretically the minimum achievable rates of both symbol based and bit-plane based coding and compared their efficiency via experiments based on low-density parity-check (LDPC) codes. They concluded that the bit-plane based coding, with a similar performance to that of its symbol based counterpart but a lower complexity, was more favorable. Zamani and Lahouti looked at this problem from a different perspective in [ZL08]. They compared the performance of symbol based and binary turbo decoding and found out the former one resulted in a smaller bit error rate in the scenario of wireless sensor networks.

In both these two publications the discussion is confined to source coding, and it is interesting to extend the investigation to joint source-channel coding and compare the performance of WZVC-B and WZVC-S for noisy video transmission environments. In this section, a framework is proposed for comparing the error resiliency of the two optional coding modes in the scenario of point-to-point wireless video transmission.

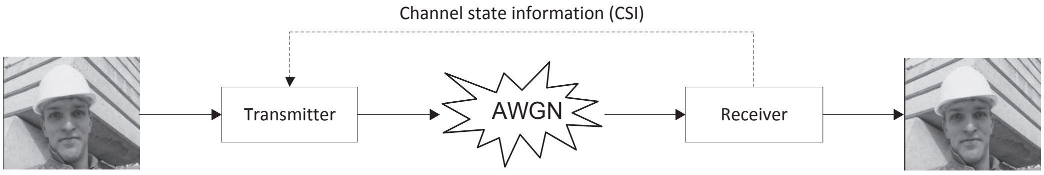


Figure 2.4: Point-to-point video transmission with feedback of CSI

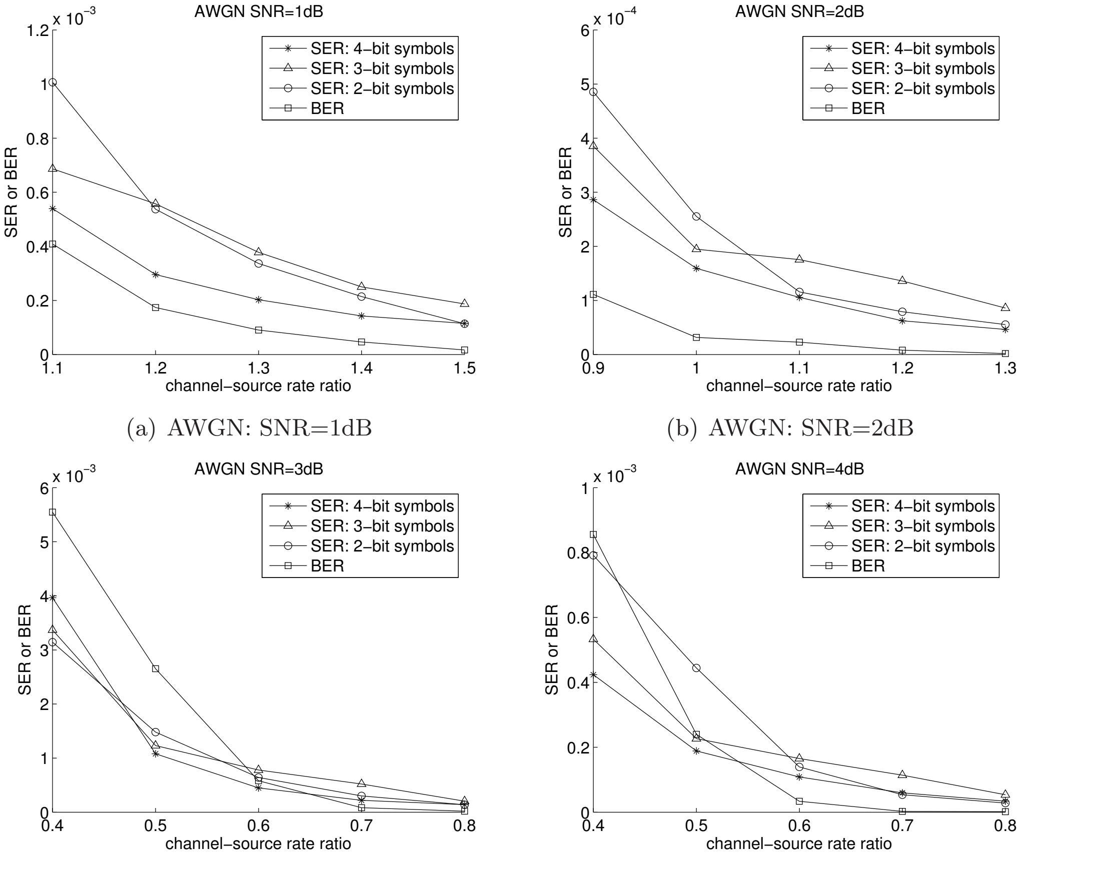
The scenario under discussion is illustrated in Fig. 2.4. The additive white Gaussian noise (AWGN) model is assumed for the transmission channel. Moreover, it is assumed that the encoder knows the estimated transmission channel state through feedback from the decoder. Therefore, the encoder can use this channel state information (CSI) to allocate rate between source coding and channel coding to achieve a high error resiliency.

To compare WZVC-S with WZVC-B effectively, a few additional assumptions are necessary. First, there is a strict delay constraint and thus retransmission is not allowed. Otherwise, all the data could finally reach the decoder error-free. It would seem as if there were no channel loss and thus the question would come back to source coding. Second, MEC at the encoder is activated for more precise prediction, which leads to a higher compression ratio. Given a certain constraint of the total bit rate, a higher compression ratio in source coding means more bits available for channel coding and thus higher error resiliency can be achieved. Third, the encoder is assumed to be capable of performing turbo decoding instantly so that the sufficient source coding rate can be determined precisely. This assumption also contributes to a higher coding efficiency, because the encoder is free of the impact of the uncertainty of the sufficient source coding rate. The second and the third assumption mean that the typically assumed low encoding complexity is disregarded, because the focus in this section is error resiliency.

For fair comparison, source-channel rate control is carried out for both WZVC-S and WZVC-B, constrained by the same bit budget. First, the performance of turbo coding is considered and offline training is carried out to obtain the bit error rate and symbol error rate of turbo decoding at different rates and different AWGN channel states. Next, a method for estimating the decoding failure probability (DFP) is proposed and the end-to-end distortion of a video slice is estimated based on DFP. Then, rate allocation between source and channel coding is carried out via adaptive quantization and the objective is to minimize the end-to-end distortion. Specifically for WZVC-B, rate allocation among different bit-planes is enabled so that the minimum end-to-end distortion can be achieved. Finally, simulations for transmitting the resulting video stream of WZVC-S and WZVC-B over an AWGN channel are performed and the quality of the decoded video for each scheme exhibits its error resiliency.

2.3.1 Performance of turbo coding

The turbo codes used for implementing the SW coding are responsible for source as well as channel coding and therefore the performance of turbo coding fundamentally determines the coding efficiency and the error resiliency of the Wyner-Ziv video codec. A good knowledge of the turbo codes' performance is a prerequisite for the joint source-channel rate control. For this purpose, training data are used to simulate the average bit error rate (BER) and the average symbol error rate (SER) for different channel coding rates and transmission channel states. Different symbol sizes lead to different symbol error rates. The symbol size here is equivalent to N_{BP}. The SER of one-bit symbols is equivalent to the BER for bit-plane based coding.



(a) AWGN: SNR=1dB

(b) AWGN: SNR=2dB

(c) AWGN: SNR=3dB

(d) AWGN: SNR=4dB

Figure 2.5: Performance of turbo codes at different AWGN SNRs (1dB, 2dB, 3dB and 4dB).

Herein, we define the channel-source rate ratio (CSR) in the following

$$\mathrm{CSR} = \frac{\text{the number of bits used for channel coding}}{\text{the number of bits used for source coding}}. \tag{2.22}$$

CSR is fundamentally the same as the conventionally defined channel coding rate (CCR) and can be uniquely converted from CCR like this: $CSR = (1 - CCR)/CCR$. The purpose of defining CSR and replacing CCR lies in two aspects. For one thing, CSR is linearly proportional to the number of bits used for channel coding and thus describes the degree of redundancy in a more straightforward manner; for another, the channel coding here is also involved in source coding and hence not to use CCR helps avoid ambiguity of rate definition for source and channel coding.

One assumption for comparing the performance of WZVC-B and WZVC-S is that the encoder knows the sufficient source coding rate. Hence, the number of bit used for source coding in this work refers to the sufficient source coding rate. For instance, suppose that the sufficient source coding rate of a WZ slice is 3/16. Now that the transmission channel is noisy, the encoder needs to add in redundancy by taking a higher rate, for instance, 5/16. The difference between the two can be regarded as the rate used for channel coding. According to the definition in (2.22), CSR in this instance equals to

$$\frac{5/16 - 3/16}{3/16} = \frac{2}{3}.$$

The performance of turbo codes for different symbol sizes and different SNRs of an AWGN channel is simulated using training data. The training data are video slices, each containing 8 MBs, extracted from the Salesman and the Stefan sequence in a QCIF (176×144) format. Every video slice is quantized using all the supported quantizers (N_{BP}=1, 2, 3 and 4) to one-bit, two-bit, three-bit and four-bit symbols. Herein, one-bit symbols are equivalent to bits.

The SER and BER of turbo codes are simulated at different CSRs and AWGN channel states. More specifically, CSR ranges from 0.1 to 2 with an increment of 0.1 and the SNR of the AWGN channel ranges from 1dB to 4dB with an increment of 1dB. Every AWGN channel state is simulated using hundred trials. Binary phase shift keying (BPSK) is taken as the modulation scheme. The results are shown in Fig. 2.5. These results are stored in the encoder and are used to estimate the decoding failure probability. Details are available in Section 2.3.2 for WZVC-S and Section 2.3.3 for WZVC-B.

For the convenience of presenting the joint source-channel rate control for WZVC-S and WZVC-B, the major acronyms and notations are listed in Table 2.2.

2.3.2 Symbol based WZVC

WZVC-S is illustrated in Fig. 2.6. This illustration takes a 3-bit quantizer as an example. First, all the pixels in a WZ slice are quantized and the resulting quantization indices, in the form of 3-bit symbols, are converted to binary strings according to natural codes. Next, these binary strings are put together one after another and forwarded to a binary turbo encoder. Then, the turbo encoder generates and transmits the parity bits to the decoder. After this, the turbo decoding carries out the decoding in a symbol based manner and recovers the original

Table 2.2: Acronyms and notations concerning joint source-channel rate control for bit-plane and symbol based WZVC

CSR	channel-source rate ratio
DFP	decoding failure probability
N_{BP}	the number of bit-planes coded; $N_{BP} = 1, 2, 3, \text{or } 4$
p^S	DFP of a MB in WZVC-S
p_i^B	DFP of the ith bit-plane of a MB in WZVC-B
D_0	distortion of a MB reconstructed using only the side information
D_i	distortion of a MB reconstructed using the side information and the first i bit-planes
D_e^S	mean end-to-end distortion for WZVC-S
D_e^B	mean end-to-end distortion for WZVC-B
ser	symbol error rate for WZVC-S
ber_i	bit error rate for the ith bit-plane in WZVC-B
R	bit budget for a slice
r_i	rate allocated to the ith bit-plane
K	the number of MBs in a slice

quantization indices in form of 3-bit symbols. Finally, the quantization symbols, together with the side information, are used to reconstruct the pixels of the WZ slice.

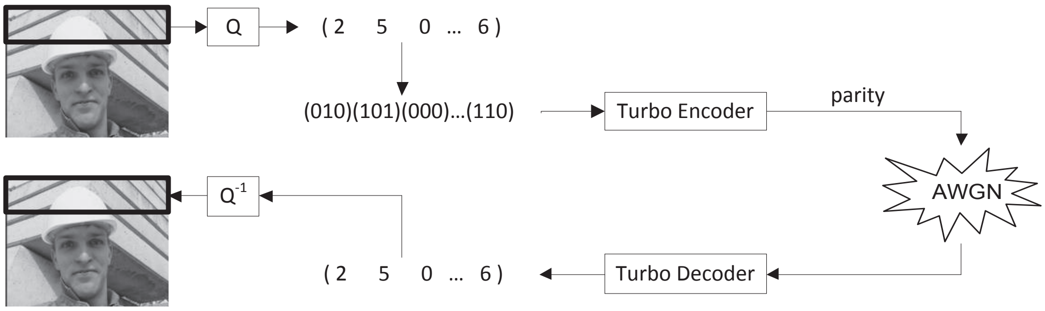


Figure 2.6: Symbol based WZVC using binary turbo encoding and symbol based turbo decoding

The joint source-channel rate control for WZVC-S lies in quantizing each slice adaptively to minimize the end-to-end distortion under the constraint of the bit budget R. More specifically, N_{BP} leaves to be determined adaptively for each slice. The estimation of the end-to-end distortion for a certain N_{BP} consists of the following major steps:

First, the CSR is calculated. The encoder is assumed to know the sufficient source coding rate and equivalently the number of bits necessary for source coding. Subtracting the number of bits for source coding from the bit budget R results in the number of bits for channel coding. Subsequently, we can calculate the CSR. A different value of N_{BP} results in a different source coding rate and thus a different CSR, too.

Second, the DFP is estimated. The relationship between SER and CSR, shown in Fig. 2.5, has been determined via training and stored in the encoder, and can be easily looked up. N_{BP} is equivalent to the symbol size in Fig. 2.5. The DFP of a MB, which contains 256 pixels, is calculated based on SER as follows:

$$p^S = 1 - (1 - ser)^{256}. \tag{2.23}$$

Third, the mean end-to-end distortion is estimated based on the DFP. If the decoding of a MB fails, this MB is reconstructed using only the side information and the resulting distortion is represented by D_0; otherwise, the reconstruction is carried out using both the decoded symbols and the side information, and the distortion is represented by $D_{N_{BP}}$. Consequently, the mean end-to-end distortion of a MB, for instance the kth one in a slice, can be formulated as

$$D_e^{(k)} = p^S \cdot D_0 + (1 - p^S) \cdot D_{N_{BP}}, \tag{2.24}$$

and averaging the mean end-to-end distortion of all the MBs in a slice results in the mean end-to-end distortion of the slice:

$$D_e^S = \sum_k D_e^{(k)}/K. \tag{2.25}$$

After the end-to-end distortion for all possible values of N_{BP} has been estimated, adaptive quantization is carried out and N_{BP} is determined such that the end-to-end distortion is minimized:

$$N_{BP}^* = \arg\min_{N_{BP}} D_e^S(N_{BP}). \tag{2.26}$$

2.3.3 Bit-plane based WZVC

WZVC-B is illustrated in Fig. 2.7. In this illustration, a 3-bit quantizer is used as an example. The coding process is similar to that of WZVC-S shown in Fig. 2.6. Two major differences, however, need to be made clear. First, the binary strings converted from the quantization indices are rearranged in three bit-planes. Specifically, the first bit, equivalently the MSB of every quantization index is extracted and all these bits are put together in the first bit-plane (BP 1). BP 2 and BP 3 are extracted in a similar way. All the bit-planes are encoded independently from the others.

A second difference between WZVC-B and WZVC-S lies in the decoding. The symbol based turbo decoder in WZVC-S directly outputs the recovered quantization indices, whereas a binary turbo decoder is employed in WZVC-B and the bit-planes are decoded one after another. The decoding of one-bit plane depends on all the previously decoded bit-planes. In this way, WZVC-B can reach a source coding efficiency as high as that of WZVC-S [WBGL07].

The dependency between bit-planes in decoding results in a “domino effect” of decoding failures. In the context of noisy video transmission, the decoding of a bit-plane might fail,

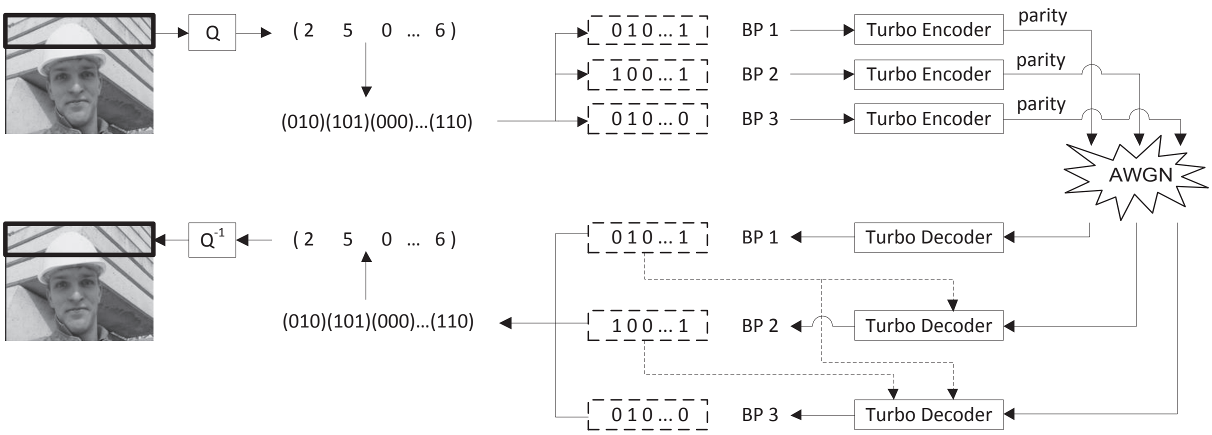


Figure 2.7: Bit-plane based WZVC using binary turbo coding

ending up with some bit errors. These errors are likely to cause decoding failures of all the following bit-planes, as observed in experiments. Therefore, if the decoding of a bit-plane fails, it is assumed that all the following bit-planes will end up with a decoding failure. This assumption is also reflected in the distortion estimation in (2.29). Above all, this "domino effect" of decoding failures in WZVC-B considerably limits the performance of WZVC-B in noisy transmission environment.

The joint source-channel rate control for WZVC-B, similar to that of WZVC-S, lies also in the adaptive decision on N_{BP} to minimize the mean end-to-end distortion of each slice. For each possible value of N_{BP}, however, rate allocation among different bit-planes is necessary to reach the minimum end-to-end distortion, and this is constrained by the same bit budget R as that in WZVC-S:

$$R = \sum_{i=1}^{N_{BP}} r_i. \tag{2.27}$$

For the ith bit-plane, the sufficient source coding rate is assumed to be known and subtracting the number of bits for source coding from the bit budget r_i results in the number of bits for channel coding. Subsequently, the CSR for this bit-plane is calculated and the ber_i is looked up in the precalculated data presented in Fig. 2.5. In general, the most significant bit-plane ends up with higher CSR and lower BER than the other bit-planes.

Now that ber_i is available, the DFP of the ith bit-plane can be calculated using the following equation:

$$p_i = 1 - (1 - ber_i)^{256}. \tag{2.28}$$

The estimation of the mean end-to-end distortion of a MB is based on the work in [RCS10] and appears in the following form:

$$D_e^{(k)} = \begin{cases} p_1 D_0 + (1-p_1)D_1, \quad if\ N_{BP} = 1; \\ p_1 D_0 + \sum_{i=1}^{N_{BP}-1}[\prod_{j=1}^{i}(1-p_j)p_{j+1}]D_j \\ +[\prod_{j=1}^{N_{BP}}(1-p_j)]D_{N_{BP}}, \quad if\ N_{BP} > 1. \end{cases} \tag{2.29}$$

Averaging the distortion of all the MBs in the slice leads to the mean end-to-end distortion of the slice:

$$D_e^B = \sum_k D_e^{(k)}/K. \tag{2.30}$$

For a certain N_{BP}, the bit budget is allocated between different bit-planes such that the end-to-end distortion of the slice is minimized:

$$r_{1:N_{BP}} = \arg\min D_e^B(r_{1:N_{BP}}) \text{ subject to } R = \sum_{i=1}^{N_{BP}} r_i, \tag{2.31}$$

and this problem is solved using the sub-gradient method [SKR85].

Finally, the mean end-to-end distortions for all candidate quantizers (N_{BP}=1, 2, 3 or 4) are compared with each other and the N_{BP}, which leads to the minimum end-to-end distortion, is chosen:

$$N_{BP}^* = \arg\min_{N_{BP}} D_e^B(N_{BP}). \tag{2.32}$$

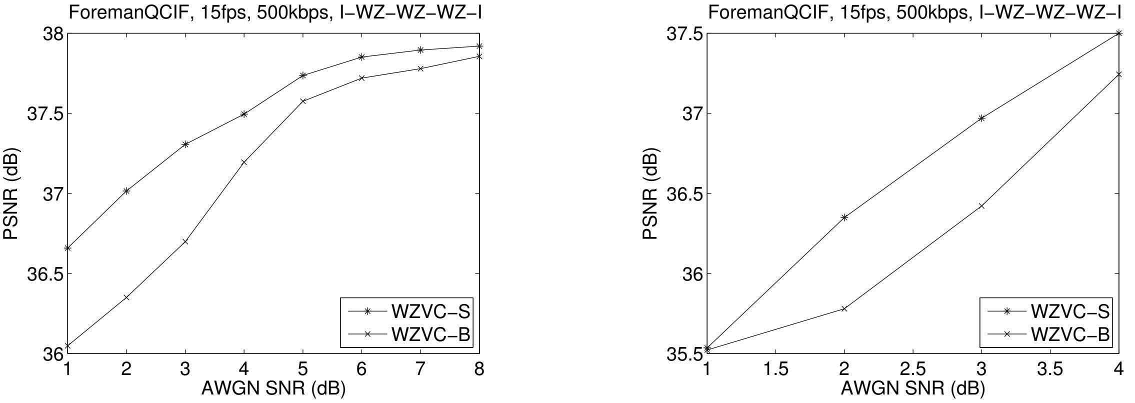
2.3.4 Experimental results

In this section, experimental results for WZVC-B and WZVC-S are presented comparatively. The first 100 frames of the Foreman sequence with a frame rate of 15 frames per second (fps) are used in the experiments. The GOP structure is set to I-WZ-WZ-WZ-I and this structure means that there are three WZ-frames between every two successive I-frames. The target bit rate for the joint source-channel rate control is 500 kbps. The simulation is performed for different AWGN channel states and every channel state is simulated using ten trials.

In a first experiment, it is assumed that the encoder has perfect knowledge of the AWGN channel state and is able to determine N_{BP} optimally for each slice. Encoding and transmission are simulated for an AWGN channel of different SNRs. The quality of the decoded video is shown in Fig. 2.8(a).

WZVC-S and WZVC-B perform equally well, when the channel state is pretty good (SNR=8 dB). Under this circumstance, there is hardly any channel-induced loss, so the question comes back to source coding. And this result agrees with the conclusion in [WBGL07]. WZVC-S exhibits a significantly better error resiliency when the channel state turns worse and its performance gain over WZVC-B is up to 0.7 dB.

In practice, however, the channel state estimation cannot always be perfect. Therefore, in a second experiment, channel mismatch is considered. More specifically, the joint source-channel



(a) End-to-end distortion with accurately estimated AWGN transmission channel state

(b) End-to-end distortion when channel mismatch occurs

Figure 2.8: End-to-end distortion comparison for WZVC-S and WZVC-B.

rate control at the encoder is carried out under the assumption that the AWGN channel state is pretty good (SNR=4 dB). Next, the transmission of the encoded video stream is simulated under different channel states (SNR=1, 2, 3 and 4 dB). The quality of the decoded video is shown in Fig. 2.8(b).

First, WZVC-S outperforms WZVC-B slightly under a good channel state (SNR=4 dB). Next, the gain increases to over 0.5 dB as the real channel state (SNR=2 or 3 dB) deviates from the assumed one. Finally, the two schemes perform equally bad when the channel state (SNR=1 dB) deviates too much from the assumed one.

In both experiments, WZVC-S exhibits better error resiliency than WZVC-B. Therefore, it can be concluded that WZVC-S is more robust than WZVC-B, if turbo codes presented in 2.1.3 are used. The conclusion might be different if other codes, for instance LDPC codes, are employed. Nevertheless, the proposed framework enables a fair comparison between symbol based and bit-plane based coding in the context of noisy video transmission.

2.4 Scalability of encoding complexity

In conventional video coding standards, such as H.264/AVC, the decoder is fairly simple. The encoder has to compute intensively, largely due to motion estimation and mode selection, to achieve the highest coding efficiency. This architecture results in an unbalanced distribution of computational complexity between encoder and decoder. Thus, the application of these video coding standards can be limited under certain circumstances, in which a low complexity encoder is on demand, take video surveillance for example.

For a significantly lower encoding complexity than that of conventional video coding, motion estimation is shifted completely to the decoder in typical Wyner-Ziv video codecs [GARRM05, PMR07, AAD$^+$07]. However, the computational complexity of the encoder and the decoder all together does not decrease at all and its distribution is still unbalanced. Apart from this, the RD performance is considerably worse than that of conventional video coding, because motion estimation at the decoder is significantly less efficient.

In order to improve the coding efficiency of WZVC, people have introduced low-complexity motion estimation to the Wyner-Ziv video encoder. A typical example was the encoder-side bit-plane motion estimation [CMCS07]. In addition, the 2D logarithmic motion search for Wyner-Ziv video coding was discussed in [CS07a].

In this section, a more flexible architecture is presented. Encoder and decoder can cooperate with one another for motion estimation. This cooperation has two aspects. First, the encoder, according to its available computation power, selects only a percentage of MBs, carries out motion estimation for them and transmits the motion vectors to the decoder. Motion estimation for the remaining MBs are handed over to the decoder. It is discussed in Section 2.4.1 how the encoder decides on image MBs to carry out motion estimation for. Second, the fast 2D logarithmic motion search is performed for the selected MBs and results in motion vectors of limited accuracy, when the decoder is ready to refine these motion vectors. The refined motion estimation at the decoder is presented in Section 2.4.2.

2.4.1 Fast motion estimation at the encoder

If the computational power of the Wyner-Ziv video encoder is limited and thus can perform fast motion estimation (ME), specifically the 2D logarithmic motion search, for only a certain percentage of MBs in a frame, these MBs need to be reasonably selected for the purpose of high RD performance. Herein, the criterion for this selection lies in how large the motion of a MB is. More specifically, MBs in high motion areas have priority and will be selected preferentially for motion estimation. For brevity, the MBs, to which motion estimation is applied to at the encoder, are referred to as ME-MBs in the following.

The reason that the motion is taken as the criterion largely lies in the property of side information generation at the decoder. In general, MCI and MCE at the decoder is more difficult than MEC at the encoder, because the original WZ-frame is available only at the encoder but not at the decoder. For low motion areas, the Wyner-Ziv video decoder is still capable of generating quite accurate side information. For MBs with large motion, however, it is rather difficult to estimate the motion accurately at the decoder. Therefore, motion estimation is carried out for these MBs at the encoder and MVs are transmitted to the decoder to help generate accurate side information.

The proposed measure for evaluating the motion of a MB is the sum of squared difference (SSD) between this MB and its co-located MB in the previous frame:

$$SSD = \sum_{i=1}^{16} \sum_{j=1}^{16} [MB_n(i,j) - MB_{n-1}(i,j)]^2, \tag{2.33}$$

where MB_n represents a block in the current frame and MB_{n-1} its co-located block in the previous frame. Heuristically, it is fine to assume that a large SSD value implies large motion, though this assumption is not always true. Given this quantitative measure, all the MBs in a WZ-frame can be sorted according to how large their motion is. MBs with higher motion are selected preferentially at the encoder for motion estimation.

Now that sorting according to SSD is required in the proposed approach for selecting the ME-MBs, it is referred to as the sorting-based selection. To verify the necessity of this sorting process, random selection is taken as the comparison scheme. And experimental results in Section 2.4.3 show that the sorting-based selection of ME-MBs leads to a higher coding efficiency than the random selection.

The major advantage of the proposed measure in (2.33) for selecting ME-MBs lies in its simplicity. The encoder needs to calculate the SSD for a MB only once before it can determine whether motion estimation should be carried out for this MB. Nonetheless, the coding efficiency would be further improved, if the encoder could spend more computation power to evaluate the motion of MBs more accurately.

Another thing worth mentioning is the block size used for motion estimation. The size of a MB is 16×16 pixels, and the block size for motion estimation can be either 16×16, 8×8 or 4×4, similar to the H.264/AVC standard. For simplicity, the discussion in the remaining part of this section is confined to a block size of 8×8 for motion estimation. And this configuration means that each MB ends up with four motion vectors after motion estimation.

2.4.2 Refined motion estimation at the decoder

Fig. 2.9 illustrates how encoder and decoder work together for accurate motion estimation. This example involves fast motion estimation, specifically the 2D logarithmic search [JJ81a], at the encoder, and motion compensated interpolation at the decoder. The motion search range at the encoder, represented by the larger dashed square in Fig. 2.9, is just normal, whereas the refined motion search at the decoder is confined to a smaller area bounded by the smaller dashed squares in I-frame 1 and I-frame 2. The discussion is based on a GOP structure of I-WZ-I, which means that there is a single WZ frame between two I-frames. The block size for motion estimation is 8×8. Nonetheless, the discussion can be extended to different GOP structures, motion compensated extrapolation and smaller block sizes for motion estimation without loss of generality.

The motion vector of every 8×8 block in a ME-MB is transmitted to the decoder. This motion vector ends up with limited accuracy, given that the 2D logarithmic search does not perform as well as full search, and thus leaves to be refined by the decoder. The motivation of fast motion estimation, however, lies in keeping the encoding complexity at a low level.

With the motion vector of a 8×8 block, the decoder first finds the location in the previous

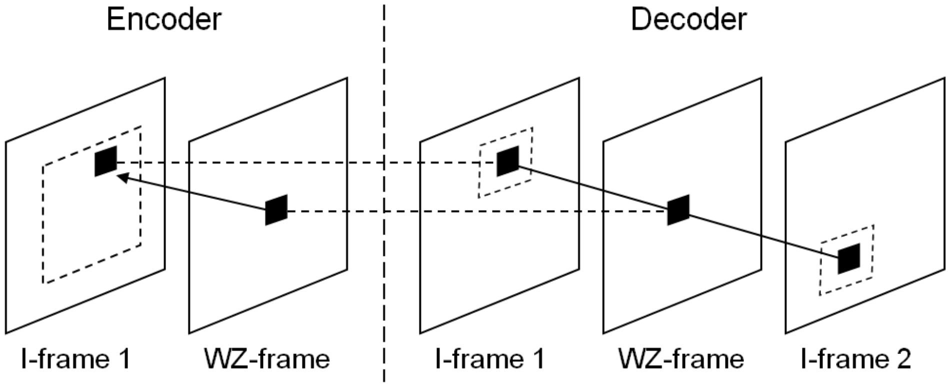


Figure 2.9: Cooperation of the encoder and the decoder for motion estimation

frame which the motion vector points at. Then it searches in a small neighborhood for a more accurate motion vector, assuming that the motion vector from the encoder is already very close to the optimal one. This refined motion estimation at the decoder is based on the previous and the following I-frames. Herein, the motion search algorithm is the bilateral motion estimation [CHKK07] combined with the spatial motion smoothing in [ABP05].

The teamwork of encoder and decoder for motion estimation has two benefits. Above all, the motion vectors obtained by fast motion estimation at the encoder can help improve the accuracy of motion estimation at the decoder. Moreover, the computation power saved via limiting the motion search range at the decoder is considerably larger than that required for fast motion estimation at the encoder, and therefore the total amount of computational complexity is reduced via the teamwork of the encoder and the decoder.

2.4.3 Experimental results

The experiments are carried out based on a GOP structure of I-WZ-I, which means that every second frame is an I-frame and between every two I-frame there is a WZ-frame. The I-frames are compressed by an H.264/AVC intraframe coder, while the WZ-frames are compressed using the pixel-domain Wyner-Ziv video codec illustrated in Fig. 2.1. For the Foreman sequence with a frame rate of 15 fps, five rate distortion (RD) points are simulated. Rate control is enabled to help generate compressed video streams at five different bit rates: 500 kbps, 400 kbps, 300 kbps, 200 kbps and 100 kbps. The results shown in this section are for the Foreman sequence, and other sequences lead to similar results.

2D logarithmic search is used as the fast motion estimation algorithm at the encoder. The motion vectors are compressed by Huffman coding before being sent to the decoder. Predictive coding for motion vectors leads to higher compression ratio at larger ME-MB ratios like 75% and 100%, but hardly helps at lower ME-MB ratios. Therefore, predictive coding for motion vectors is excluded for simplicity. The block size for motion estimation is 8×8 and this practice means that motion estimation is applied to each of the four 8×8 blocks within a MB. The simulation is carried out for five different percentages of ME-MBs at the Wyner-Ziv video encoder: 0%, 25%, 50%, 75% and 100%. ME-MBs denote the MBs to which fast motion estimation is applied to at the encoder. At one extreme, when the percentage of ME-MBs is 0%, no motion estimation is performed at the encoder at all, and the decoder takes over the entire job for motion estimation. At the other extreme, when the percentage of ME-MBs is 100%, motion estimation is performed for all the MBs at the encoder and the decoder needs to refine the motion estimation only. In between, the encoder has to select a certain percentage of ME-MBs. At this point, the proposed sorting-based selection is compared with the random selection.

As far as the Foreman sequence is concerned, the motion search range for ME-MBs at the encoder is ±8 pixels, while that for the refined motion estimation at the decoder is reduced to ±2 pixels. For other MBs, motion estimation is completely taken over by the decoder and the motion search range is ±8 pixels.

To show the scalability of motion estimation at the encoder, the experimental results are presented in three parts, including the quality of the side information, the rate-distortion performance, and the comparison of the sorting-based and the random selection of ME-MBs.

Firstly, the quality of side information along with the average number of block comparisons are presented in Table 2.3 to evaluate the teamwork of the encoder and decoder for motion estimation. The PSNR of the side information for five different percentages of ME-MBs as well as for four different bit rates are shown. Herein, the side information refers to the one finally generated, involving both the MVs from the encoder and the refined motion estimation at the decoder.

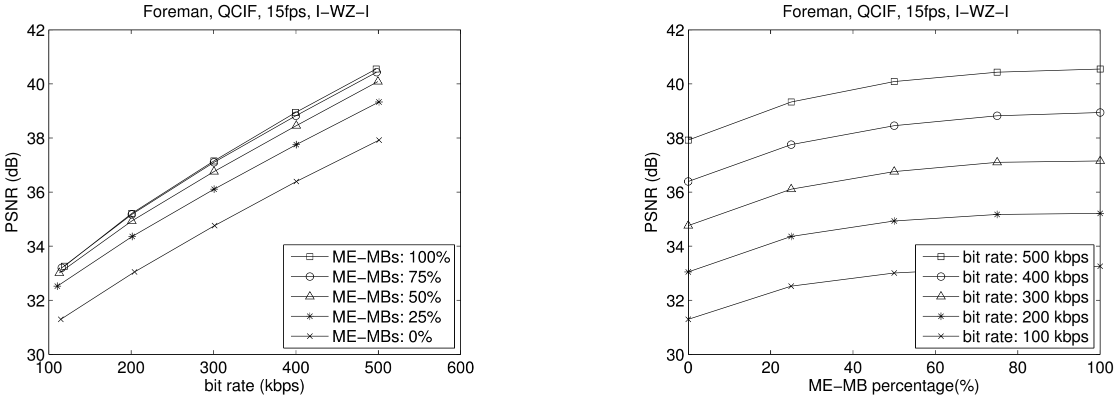
The number of block comparisons is used as a measure for the computational complexity of motion estimation. A block comparison refers to the calculation of the SSD between a block and one of its candidate blocks in the reference frame. To avoid ambiguity, the block size is always 8×8 in this section, when it comes to the concept of block comparison. The presented number of block comparisons, both at the encoder and the decoder, is an average. The average number of block comparisons is calculated by dividing the total number of block comparisons for all the MBs by the total number of non-overlapping 8×8 blocks in the video sequence. At the encoder, 2D logarithmic search is applied to each 8×8 block in ME-MBs and thus the number of block comparison is smaller than that of full search. At the decoder, full search is used for motion estimation. The search range for each 8×8 block in ME-MBs is ±2 pixel and thus 25 block comparisons are needed. For other MBs, the search range is ±8 pixels and thus 289 block comparisons are needed. The number of block comparisons at the encoder and the decoder are added up and the sum is presented in the last column of Table 2.3. This sum measures how much computational power the encoder and the decoder consume for motion estimation altogether.

Table 2.3: Quality of the side information and the number of block comparisons for different percentages of ME-MBs

Percentage	PSNR of side information (dB)				Average number of block comparisons		
of ME-MBs	500 kbps	400 kbps	300kbps	200 kbps	Encoder	Decoder	Total
0%	30.33	30.14	29.80	29.36	0	289	289
25%	33.87	33.60	33.19	32.57	3.3	223	226.3
50%	35.44	35.09	34.52	33.76	6.6	157	163.6
75%	36.05	35.65	35.05	34.20	9.7	91	100.7
100%	36.19	35.77	35.15	34.31	13.1	25	38.1

Table 2.3 shows that the quality of the side information generated at the decoder increases with the percentage of ME-MBs. The reason lies in the fact that the motion vectors transmitted from the encoder can help the decoder estimate the motion more accurately. In addition, because the fast motion search algorithm is used at the encoder, the increase of the number of block comparisons at the encoder is limited. In comparison, the number of block comparisons at the decoder is largely reduced, because the search range for the refined motion search at the decoder is made much smaller. Above all, motion estimation at the encoder improves the quality of the side information considerably and the team work of encoder and decoder for motion estimation can reduce the total computational complexity.

Secondly, the rate-distortion performance and video quality for different percentages of ME-MBs are shown in Fig. 2.10 to illustrate the scalability of motion estimation at the encoder and its influence on the coding efficiency and reconstruction quality of the video.



(a) Rate distortion curves for different percentages of ME-MBs

(b) Video quality vs. percentage of ME-MBs at different bit rates

Figure 2.10: Scalability of motion estimation at the Wyner-Ziv video encoder

RD curves for the Foreman sequence for different percentages of ME-MBs are put together for comparison in Fig. 2.10(a). When fast motion estimation is applied to all the MBs at the encoder, the best RD performance is achieved. When no motion estimation is performed at the encoder at all, the RD performance is the worst. The gap between the two extremes is around 2 dB over the entire bit rate range. The RD curves for other percentages of ME-MBs lie in between and the higher the percentage is, the better the RD performance. To sum up, the motion estimation at the encoder improves the coding efficiency.

Fig. 2.10(b) presents the same results but from a different perspective. Herein, the focus lies in how much the reconstruction quality improves as the percentage of ME-MB increases. And the results at different bit rates are presented. It can be observed that the PSNR improvement is not proportional to the ME-MB percentage. Specifically, the gain is up to 1dB if the ME-MB percentage goes from 0% to 25%, whereas the improvement is little if it goes from 75% to 100%. The fundamental reason lies in the proposed sorting-based selection of ME-MBs. More specifically, the first 25% of ME-MBs are selected in priority because their motion is estimated to be large and they need motion estimation the most. In contrast, the last 25% ME-MBs have little motion and tend to end up with the skip mode, and therefore applying motion estimation to them leads to little improvement in reconstruction quality. Above all, it is recommended to apply motion estimation to some of the MBs but not all, and it is meaningful to select these ME-MBs carefully.

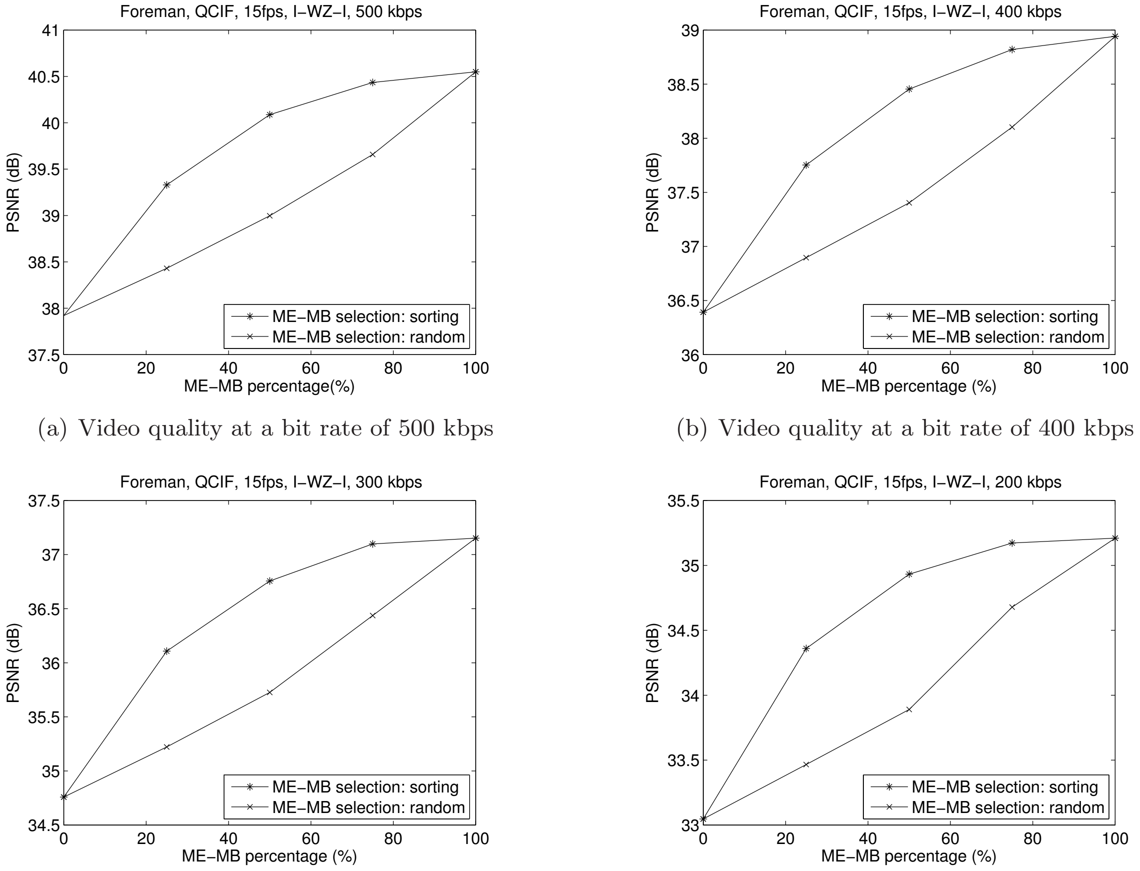
In Fig 2.11, the proposed sorting-based selection of ME-MBs is compared with the random selection in terms of the resulting video quality. Results at different bit rates are presented separately. A percentage of 0% and 100% does not involve selecting ME-MBs, thus the curves for the two schemes overlap at these points. At other percentages of ME-MBs, the proposed sorting-based selection outperforms the random selection by up to 1 dB. This means the proposed sorting-based selection of ME-MBs, though simple, can be quite effective.

To sum up, this section refers to the scalability of motion estimation at the Wyner-Ziv video encoder. This scalability is realized by applying motion estimation to different percentages of ME-MBs. In general, the more the encoder is engaged in motion estimation, the higher the coding efficiency is. It is recommended to perform fast motion estimation to part of the MBs at the encoder, typically 25% or 50%, where a good trade-off between high coding efficiency and low encoding complexity can be achieved.

2.5 Chapter summary

This chapter concerns the development and the optimization of the Wyner-Ziv video codec. There are three major contributions including the optimization of the turbo decoding, the comparison of bit-plane and symbol based coding and the scalability of the encoder's computational complexity.

To begin with, discussion is confined to source coding and error-free parity bits are available for turbo decoding. The "sum_max" criterion is proposed for calculating the L-values for the parity bits and improving the decoding performance. This practice can achieve the optimal decoding performance while the standard decoding algorithm does not need to be changed.



(a) Video quality at a bit rate of 500 kbps

(b) Video quality at a bit rate of 400 kbps

(c) Video quality at a bit rate of 300 kbps

(d) Video quality at a bit rate of 200 kbps

Figure 2.11: Comparison between sorting-based and random selection of MBs for motion estimation

Then, the discussion is extended to video transmission over error-prone channels. Bit-plane and symbol based WZVC are compared in terms of error resiliency. The conclusion is that the symbol based WZVC leads to better error resiliency than the bit-plane based WZVC, if turbo codes are used for WZVC and the transmission channel is assumed to be an AWGN channel.

Finally, an extended functionality of the Wyner-Ziv video codec lies in introducing a scalable motion estimation to the Wyner-Ziv video encoder. The encoder can do without motion estimation to achieve the lowest possible encoding complexity, or perform motion estimation for an arbitrary percentage of the MBs to trade off low encoding complexity and high coding efficiency flexibly.

Chapter 3

Joint source-channel rate control

The focus of this chapter lies in the error-resilient video transmission using WZVC. The objective is to achieve the highest possible quality of the decoded video at a certain target bit rate. Therefore, this problem is formulated as a joint source-channel rate control problem. The relevant state-of-the-art work is discussed in Section 3.1.

The addressed scenario is real-time video transmission over a wireless error-prone channel. This scenario, together with its challenging issues and principal assumptions, is presented in Section 3.2. Additionally, two variants of WZVC-based joint source-channel coding, namely WZVC and WZVC+FEC, are presented in Section 3.2.1 and Section 3.2.2, respectively. It is a contribution of this chapter to compare WZVC and WZVC+FEC in terms of error resiliency.

The wireless error-prone channel is modeled as an AWGN channel and a significant issue lies in the uncertainty of sufficient rates, both for source and channel coding, and an approach addressing this issue is presented in Section 3.3. This approach is a major contribution presented in this chapter.

Section 3.4 refers to joint source-channel rate control. It is explained how different components, including estimation of decoding failure probability, estimation of end-to-end distortion and slice-wise adaptive quantization, are put together to generate an error-resilient Wyner-Ziv video stream at a target bit rate.

Experimental results for WZVC and WZVC+FEC are presented comparatively in Section 3.5. These results support the conclusion that WZVC leads to better error-resiliency than WZVC+FEC in real-time video transmission over a wireless error-prone channel. In the end, a summary is made for this chapter in Section 3.6.

3.1 State of the art

In this section, the state-of-the-art joint source-channel coding schemes are presented. In Section 3.1.1, relevant work on the joint source-channel coding for image and video transmission is presented. This section summarizes typical work ranging from the classic JPEG image codec to the current popular H.264/AVC video codec. The focus of Section 3.1.2 is WZVC-based error resilient video transmission systems. Based upon this set of work, the contribution of the thesis in this chapter is evaluated.

3.1.1 JSCC for image and video

Goldsmith and Effros discuss extensively the joint design of source and channel codes [GE98]. They propose three design schemes and test them thoroughly using image data. Their conclusion is that the channel codes remove most of the channel errors at the optimal bit allocation and thus the three schemes are roughly equivalent. Moreover, the three schemes have a comparable performance, even if optimal bit allocation is not possible due to channel mismatch.

Sherwood and Zeger trade off the available transmission rate between source coding and channel coding based on JPEG [SZ97]. Similarly, Banister et al. perform such optimization based on JPEG2000 [BBF02]. Li and Chen propose an unequal error protection scheme (UEP) based on JPEG 2000 and optimize the channel coding rate for individual bit-planes [LC01]. What all these works have in common lies in the separation of source and channel coding.

There has been extensive work on channel adaptive rate control for video transmission. Typically, He et al. propose a rate distortion model for joint source-channel rate control, reducing significantly the end-to-end distortion [HCC02]. In a most recent work, Yang et al. use H.264/AVC video coding and determine dynamically the quantization parameter (QP) to adapt the source rate to the condition of the transmission channel [PZS10].

3.1.2 Error-resilient WZVC

In quite a few practical Wyner-Ziv video codecs [RBG08, PMR07, XX06, AAD$^+$07], channel codes are employed to achieve data compression. Now that channel coding is adopted in WZVC, it is quite straightforward to increase the coding rate to resist against the noise of the transmission channel. In this way, channel codes are employed for the dual purpose of both source coding and channel coding. This approach is referred to as WZVC in the following. In comparison, a conventional way of transmitting videos over noisy channels lies in the separation of source coding and channel coding. Specifically speaking, source coding is performed first and forward error correction (FEC) is then applied independently to resist channel noise. This scheme is referred to as WZVC+FEC. Please note that FEC is a synonym for channel coding in telecommunications.

Both options, WZVC and WZVC+FEC, have proved more robust than conventional video coding like H.264/AVC. The WZVC option is taken in PRISM [PMR07] and the DISCOVER codec [AAD$^+$07]. In these two schemes, the Wyner-Ziv video stream is transmitted directly over the error-prone channel without further FEC protection. A more complicated variant in this class is SLEP [RBG08]. Herein, there is in some sense a layered structure. For each slice of video, a redundant slice is generated via stronger quantization and coded in the WZVC manner. These Wyner-Ziv slices are transmitted together with the original slices and no additional FEC is employed. All of the three schemes are more robust than H.264/AVC protected using FEC at a considerably large packet loss rate or symbol error rate.

The option WZVC+FEC is used in layered Wyner-Ziv video coding [XX06] and also exhibits better error-resiliency than H.264/AVC protected using FEC. In this work, there is an explicit layered structure: a base layer and an enhancement layer. The base layer is coded using H.264/AVC codec whereas the enhancement layer using WZVC. The coded video stream is further protected using FEC for transmission over an error-prone channel.

Now that WZVC and WZVC+FEC are both more error-resilient than H.264/AVC protected using FEC, the question arises which of the two options is more error-resilient, WZVC or WZVC+FEC? The first work addressing this issue in the context of distributed source coding (DSC) is in [HCC07], in which Hua et al. compare the robustness of the two schemes, namely DSC without protection and DSC protected using FEC, confining the discussion to binary data. They conclude that the former scheme is more robust. This discussion is extended to video data in this chapter. More specifically, WZVC is compared with WZVC+FEC in terms of error-resiliency for a noisy video transmission environment.

Compared with the state-of-the-art, the contribution of this chapter lies in the joint source-channel rate control using a stand-alone Wyner-Ziv video codec. The end-to-end distortion is estimated based upon the transmission channel state and rate allocation between source and channel coding is carried out via adaptive quantizaton. The work in [PMR07, XX06, AAD$^+$07], however, does not explicitly adapt the quantization to the channel state. In comparison, SLEP [RBG08] involves a quite accurate end-to-end estimation. Moreover, the channel state is used to allocate rate between original video slices and their redundant slices, such that error-resiliency is optimized. One limitation of this work, however, lies in the high system complexity due to its layered structure, involving original and redundant video slices. The former one requires an H.264/AVC video codec, while the latter one a Wyner-Ziv video codec. This structure increases the difficulty of practical implementation.

3.2 Real-time wireless video transmission

The work in this chapter targets real-time error-resilient video transmission over a noisy wireless channel, as shown in Fig. 3.1. The objective is to maximize the quality of the video decoded at the receiver. The transmitter is supposed to compress a video in a real-time manner under the constraint of a certain target rate, modulate the resulting bit stream using BPSK and then transmit the modulated bits over a noisy wireless channel. The transmission channel is assumed to be an AWGN channel and might corrupt the bits transmitted over it. The receiver decodes the video and checks whether the decoded video is error-free via cyclic redundancy check (CRC). Error concealment is carried out if decoding errors are identified.

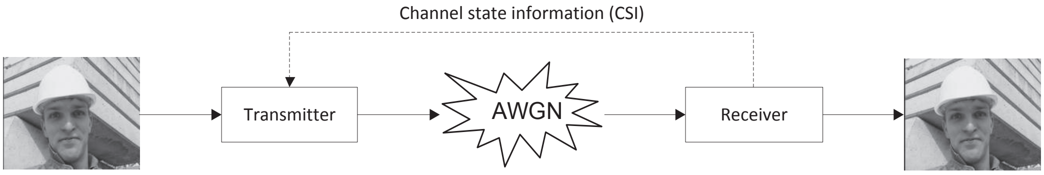


Figure 3.1: Real-time point-to-point video transmission over a noisy wireless channel

Now that the target is real-time application, there is a rather strict delay constraint, which leads to two necessary assumptions. First, rate control in a “decode and request” manner [GARRM05] via a feedback communication channel cannot be adopted, because there might be several rounds of interaction between encoder and decoder and these round trips would add to the end-to-end delay. Second, decoding is not allowed at the encoder, because the decoding in the context of WZVC is no other than decoding of channel coding and is relatively time consuming. The delay would be enlarged significantly. The consequence of these two consumptions lies in the uncertainty of sufficient rates, which will be discussed in detail in Section 3.3. Simply speaking, it is not deterministic how many bits the encoder has to transmit to the decoder so as to ensure successful decoding. And this issue is challenging in joint source-channel coding for WZVC.

A third assumption is that the function of retransmission, typically adopted in standard communication protocols such as the transmission control protocol (TCP), is not allowed. The reason is that retransmission would result in higher delay and higher complexity of system design for video transmission. Moreover, the lossy transmission channel would seem lossless, because all the data could finally reach the receiver error-free via retransmission or hybrid-ARQ. In this instance, the problem is not error-resilient transmission but narrows down to source coding efficiency. It remains an open question, whether WZVC or H.264/AVC would lead to higher video quality at the same bit rate, if retransmission were enabled. This issue is out of the scope of this thesis.

Additionally, it is assumed that the state of the transmission channel is estimated and known to the encoder. Now that an AWGN channel and BPSK modulation are assumed, signal-to-noise ratio (SNR) is used to represent the state of the transmission channel state. A larger SNR value means that the channel is relatively good, whereas a smaller SNR implies stronger noise on the channel. In practice, a few pilot bits, agreed on by the transmitter and the receiver in advance, can be transmitted via the lossy channel to the receiver. Subsequently, the receiver can estimate the SNR of the AWGN channel by evaluating how much the channel noise has corrupted the pilot bits. Finally, the SNR of the AWGN channel is signaled back to the encoder. The joint source-channel rate control will be adapted to the transmission channel state.

Under all the above mentioned assumptions, a specific issue investigated in this chapter lies in the comparison of two variants, namely WZVC and WZVC+FEC, in terms of error resiliency. WZVC just refers to the basic codec architecture presented in Section 2.1. A single turbo coder is involved. This turbo coder is dedicated to source coding if error-free transmission can be ensured. However, if the transmission channel is noisy, this turbo coder is employed for a dual purpose of source and channel coding and the coding rate is increased to resist the channel noise. The system of the WZVC scheme is presented in detail in Section 3.2.1.

In comparison, WZVC+FEC, presented in detail in Section 3.2.2, involves two turbo coders. The first one is dedicated to source coding only, whereas the second one, concatenated with the first one, is dedicated to channel coding. The coding rate of the first turbo coder is determined by the source statistics, while that of the second one is determined according to the SNR of the AWGN channel. Above all, source and channel coding are implemented using two turbo coders and in two separate stages and therefore this instance is referred to as WZVC+FEC.

For a fair comparison between WZVC and WZVC+FEC, joint source-channel coding is applied to both schemes at the same bit budget. Herein, the bit budget refers to the total bit rate for source and channel coding altogether. The joint source-channel rate control for WZVC as well as WZVC+FEC is discussed in Section 3.4. The video transmission is simulated and the error-resilience of each scheme is evaluated with the quality of the decoded videos.

For the readers' convenience, the frequently used acronyms and notations in this chapter are listed up in Table 3.1. To clarify the fundamental difference between WZVC and WZVC+FEC, the system diagram of the two are presented in Section 3.2.1 and Section 3.2.2, respectively. The underlying mode decision, presented in Section 2.1.1, is not mentioned here, and the discussion primarily refers to the WZ MBs or slices. In fact, the mode decision for WZVC is identical to that for WZVC+FEC, and the processing of intra MBs and skip MBs is also the same for both schemes. This practice excludes the side effects of intra MBs and skip MBs and enables a fair comparison of WZVC and WZVC+FEC for WZ MBs.

3.2.1 Robust video transmission using WZVC

The WZVC scheme, as illustrated in Figure 3.2, involves only one turbo coder, which is used for the dual purpose of source and channel coding. The quantization indices of a WZ slice are coded by the turbo encoder. Next, the resulting parity bits are modulated using

Table 3.1: Acronyms and notations concerning joint source-channel rate control for WZVC and WZVC+FEC

$H(X\|Y)$	conditional entropy
S	sufficient source coding rate
Γ	sufficient channel-source rate ratio
CSR	channel-source rate ratio
p_{wzvc}	decoding failure probability for the WZVC scheme
p_{fec}	decoding failure probability for the WZVC+FEC scheme
N_{BP}	number of bit planes, $N_{BP} = 1$, 2, 3 or 4
D_0	distortion of a MB reconstructed using only the side information
$D_{N_{BP}}$	distortion of a MB reconstructed using the side information and the first N_{BP} bit-planes
$D_e(k)$	mean end-to-end distortion of the kth MB in a slice
D_e	mean end-to-end distortion of a slice
BB	bit budget
R	rate allocated to source coding for the WZVC+FEC scheme
R^*	optimal source rate for the WZVC+FEC scheme

BPSK and transmitted through an AWGN channel to the decoder, whereas the systematic bits are discarded. At the receiver, turbo decoding is carried out based on the received parity bits and the side information. A CRC is applied to the quantization indices of each MB at the reconstruction stage. If the CRC of a MB is fine, the decoded quantization indices are regarded as error-free and thus used for reconstruction together with the side information of this MB. If the CRC of a MB fails, the decoded quantization indices are discarded and the side information is taken as the reconstructed MB.

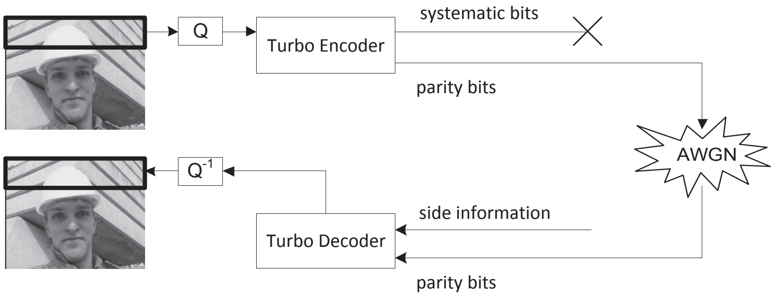


Figure 3.2: Error resilient video transmission using WZVC

The turbo coding is implemented in a symbol based manner [ASW08, Ber98], as presented in Section 2.3.2. The advantage of symbol based coding over bit-plane based coding has been identified in Section 2.3.

Motion estimation, limited to a single reference frame, is performed at the encoder. A full search with up to quarter pel accuracy is implemented for high coding efficiency. Motion vectors are entropy coded and transmitted to the decoder. The transmission is assumed to be error-free. The decoder generates side information by performing motion compensation based on the reference frame and the motion vectors.

3.2.2 WZVC protected using FEC

The WZVC+FEC scheme, as shown in Figure 3.3, employs two turbo coders. Source coding of the quantization indices of a WZ slice is performed using turbo encoder 1. Systematic bits are discarded and parity bits are forwarded to turbo encoder 2. Herein, the parity bits for WZVC are treated as systematic bits for turbo encoder 2 and parity bits are generated for them. Then, both the systematic bits and the parity bits output from turbo encoder 2 are modulated using BPSK and transmitted over an AWGN channel to the decoder. At the receiver, turbo decoder 2 performs channel decoding and forwards the output to turbo decoder 1. Turbo decoder 1 tries to decode the quantization indices using the output from turbo decoder 2 and the side information available at the decoder.

For a fair comparison with the WZVC scheme, the turbo encoder 1 and turbo decoder 1 carry out symbol based coding [ASW08, Ber98], as presented in Section 2.3.2. The turbo encoder 2 and turbo decoder 2 are implemented using binary turbo codes, because the input of turbo encoder 2 are originally binary data.

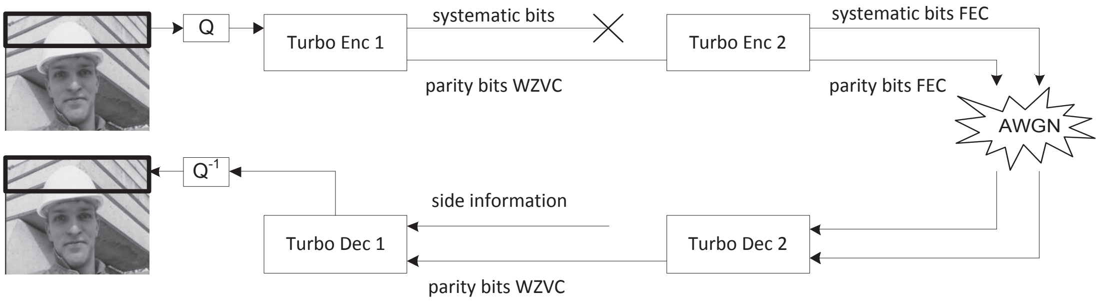


Figure 3.3: Error resilient video transmission using WZVC+FEC

The other parts of the system are identical to their counterparts in the scheme of WZVC, including motion estimation, side information generation and the reconstruction of MBs based on CRC.

3.3 Uncertainty of sufficient rates

This section concerns the most challenging issue in joint source-channel rate control for WZVC. This challenging issue lies in the uncertainty of sufficient source coding [RCS10] and channel coding rate [CSC11b]. In other words, it is difficult to determine how many bits the encoder needs to transmit for source coding and for channel coding to ensure successful decoding of the quantization indices. Experimental results, however, exhibit some statistical properties of the sufficient rates. And the sufficient rate for source and channel coding are modeled as random variables in this thesis. Details are presented in Section 3.3.1 and Section 3.3.2.

Because of the uncertainty of sufficient rates, it is often probable that the decoding fails. Now that the sufficient rates are modeled as random variables, the decoding failure probability can be estimated. The estimation of the decoding failure probability is presented in Section 3.4.3.

3.3.1 Sufficient source coding rate

The sufficient source coding rate is defined as the minimum source coding rate that allows successful source decoding. Precisely speaking, the source coding rate here means the SW rate and conforms to the definition in (2.1). According to the SW theorem, the conditional entropy of a discretely distributed random source given its side information is the lower bound of the coding rate. In practice, however, this lower bound cannot be achieved. Nevertheless, the conditional entropy is a significant measure for estimating the sufficient source coding rate. And its value depends on the correlation channel model presented in Section 2.1.4. How the conditional entropy is calculated in this thesis is presented in the following:

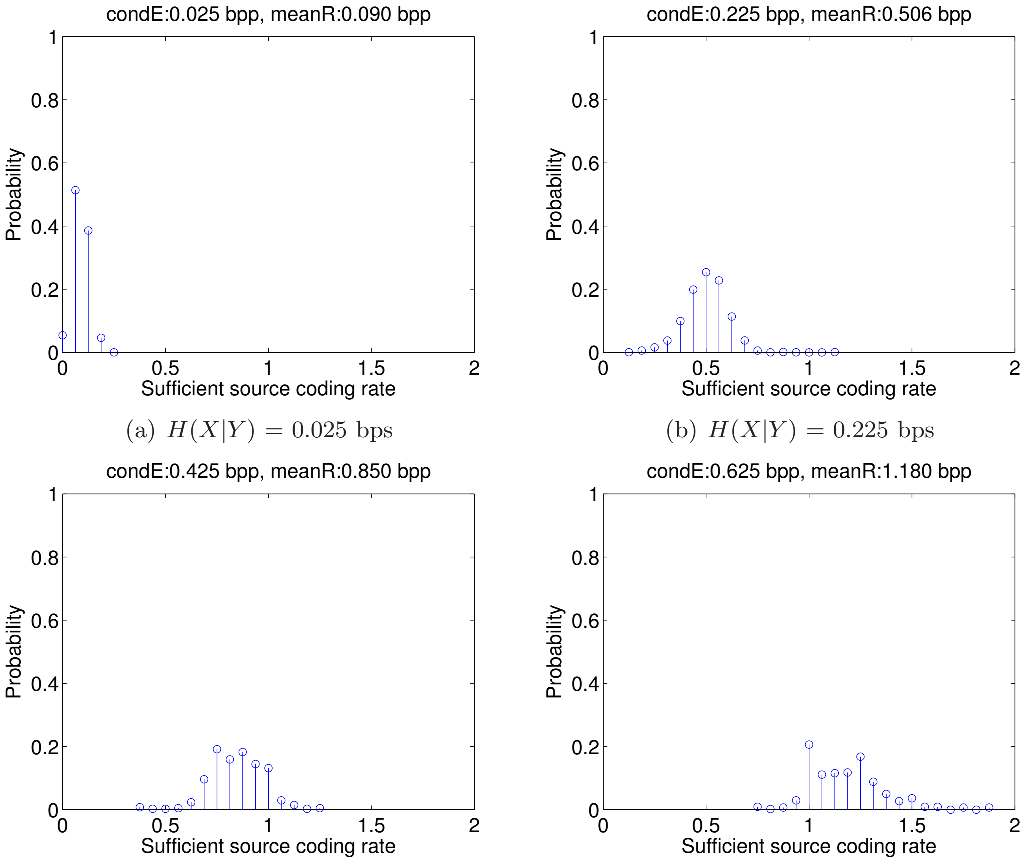
$$H(X|Y) = \sum_{y=0}^{255} p_Y(y) \sum_{x=0}^{2^{N_{BP}}-1} p_{X|Y}(x|y) \cdot \log_2 p_{X|Y}(x|y), \tag{3.1}$$

where $p_Y(y)$ represents the probability mass function (PMF) of the side information Y, and $p_{X|Y}(x|y)$ the conditional PMF of the quantization index X given the side information Y. The value of Y is an integer ranging from 0 to 255, while the quantization index X ranges from 0 to $2^{N_{BP}} - 1$.

Strictly speaking, the conditional entropy of a WZ slice should be expressed as the conditional entropy of the quantization indices of a WZ slice. For simplicity, the former one is used throughout this chapter, because it is straightforward that the SW coder codes the quantization indices. Above all, whenever the conditional entropy of a WZ slice is discussed, it refers to the quantization indices.

Hopefully, the conditional entropy of a source given its side information can determine the sufficient source coding rate. Unfortunately, this is not true. To find out some regularity connecting the conditional entropy and the sufficient source coding rate, an experiment is conducted. The Salesman sequence and the Stefan sequence, both in the QCIF format (176×144), are used in this experiment. They are coded in the WZ manner. For each WZ slice, its conditional entropy and sufficient source coding rate are recorded. This is referred to as a sample vector. The WZ video coding is simulated using different quantizers so that a wide range of conditional entropy and sufficient source coding rate is covered and a large number of sample vectors are collected.

The resulting sample vectors in this experiment are clustered according to the value of the conditional entropy and analyzed to discover the relationship between the conditional entropy and the sufficient source coding rate. In the experiment, the range of conditional entropy ended up from 0 to 1 in a unit of bit per sample (bps), and this range is divided to 20 non-overlapping bins centered at 0.025 bps, 0.075 bps, 0.125 bps ... , 0.925 bps. Each sample vector collected in the experiment falls into one of the 20 bins. The distribution of sufficient source coding rate of all the sample vectors in each bin is calculated. The distribution of sufficient source coding rate in four bins centered at 0.025 bps, 0.225 bps, 0.425 bps and 0.625 bps are shown as examples in Fig. 3.4.



(a) $H(X|Y) = 0.025$ bps

(b) $H(X|Y) = 0.225$ bps

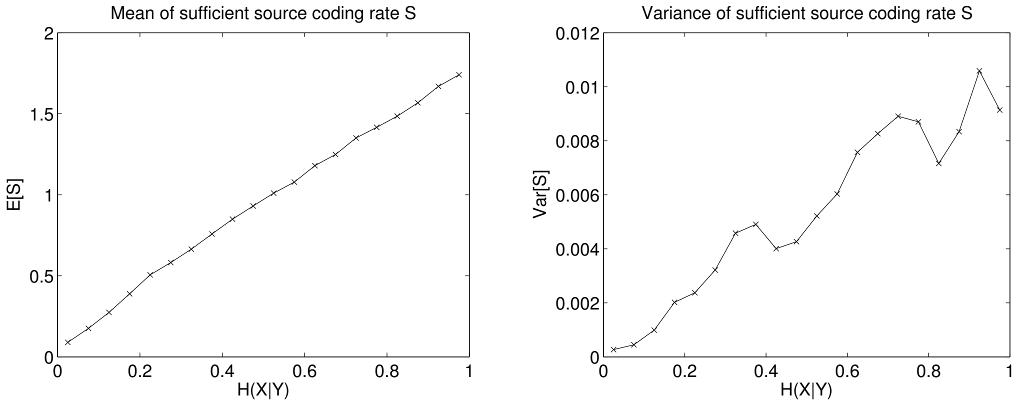
(c) $H(X|Y) = 0.425$ bps

(d) $H(X|Y) = 0.625$ bps

Figure 3.4: Distribution of the sufficient source coding rate S for different conditional entropy $H(X|Y)$.

Fig. 3.4 shows the uncertainty of the sufficient source coding rate. Given a specific value of conditional entropy, the sufficient source coding rate is indeterministic and follows a certain distribution that looks symmetric. The mean of the distribution increases with the conditional entropy. Equivalently, the larger the conditional entropy, the higher the averaged sufficient source coding rate. Moreover, it can be observed that the distribution is peaky for small conditional entropy and flat for large conditional entropy. This phenomenon indicates that the variance as well as the uncertainty of the distribution increases with the conditional entropy.

How the mean and variance of the distribution exactly changes with the conditional entropy is presented in Fig. 3.5. Herein, the sample mean and sample variance of the distribution of the sufficient source coding rate for each of the 20 bins of conditional entropy is presented. All these data are saved in a lookup table at the Wyner-Ziv video encoder. Whenever the conditional entropy of a WZ slice is known, the mean and variance can be extracted from the lookup table.



(a) Mean of sufficient source coding rate (b) Variance of sufficient source coding rate

Figure 3.5: Mean and variance of the sufficient source coding rate S for different conditional entropy $H(X|Y)$.

Finally, the sufficient source coding rate S is modeled as a Gaussian random variable. After the mean and variance of S is determined according to the conditional entropy, the probability density function f_S can be uniquely specified as a Gaussian distribution. The purpose of this modeling lies in simplifying the representation of the sufficient source coding rate. For one thing, it is enough to save the data for mean and variance in Fig. 3.5 in the encoder, and the storage of the original data for the distribution shown in Fig. 3.4 can be spared; for another, the numerical calculation of the decoding failure probability, presented in Section 3.4.3, is more convenient, when the distribution is Gaussian.

To sum up, the conditional entropy $H(X|Y)$ fundamentally influences the sufficient source coding rate. Nevertheless, the sufficient source coding rate is indeterministic but follows a seemingly symmetric distribution. This distribution is modeled as a Gaussian distribution, and its mean as well as its variance for different values of $H(X|Y)$ are estimated using a large set of training data. The distribution of the sufficient source coding rate is the fundamental for the calculation of the decoding failure probability presented in Section 3.4.3.

3.3.2 Sufficient channel-source rate ratio

Apart from the uncertainty of the sufficient source coding rate presented in Section 3.3.1, another challenging issue in joint source-channel coding using WZVC lies in the uncertainty of the channel coding rate. To avoid ambiguity, it is necessary to point out that this uncertainty does not refer to the transmission channel. It is true that the transmission channel can change so rapidly with time that it is hard to estimate and thus appears uncertain, but this uncertainty of the transmission channel state is not the topic in this section. Instead, it is assumed that the transmission channel, modeled as an AWGN channel throughout this chapter, is stable and its SNR can be estimated quite accurately. The uncertainty of the channel coding rate means that it is indeterministic how many redundant bits are needed for FEC to correct all the transmission errors caused by the AWGN channel noise, even if the SNR of the AWGN channel is determined.

To describe how much redundancy needs to be introduced in FEC to resist the transmission channel noise, the sufficient channel-source rate ratio Γ is defined as the minimum channel-source rate ratio that ensure successful FEC decoding, or in other words, that ensure the correction of all the channel errors. The definition of the channel-source rate ratio is identical to the one in Section 3.4.3. The sufficient channel-source rate ratio characterizes how strong the source needs to be protected by FEC. In fact, the channel-source rate ratio can be uniquely mapped to a conventionally defined channel coding rate. The concept of channel-source rate ratio is introduced here, because it is linearly proportional to the number of bits used for FEC, thus tends to be convenient in the context of bit allocation between source and channel coding.

Theoretically speaking, the sufficient channel-source rate ratio is $(1 - C)/C$, given a discrete channel with channel capacity C, so that the decoding error probability approaches zero. Experimental results show, however, especially with short codewords, that the sufficient channel-source rate ratio is not deterministic but exhibits statistical properties, even if the SNR of the AWGN channel state is determined. This is why Γ is treated as a random variable. The parameters of its distribution, namely mean and variance, fundamentally depend on the transmission channel state.

To find out the relationship between the distribution of Γ and the SNR of the AWGN channel, an experiment is carried out using the Salesman and the Stefan sequences, both in the QCIF format (176×144), as training data. For each WZ slice, following steps are necessary before its sufficient channel-source rate ratio can be determined:

Step 1: The WZ slice is encoded at the sufficient source coding rate S. This rate ensures a successful decoding provided that the transmission channel is error-free.

Step 2: Initialize the channel-source rate ratio (CSR) as 0.

Step 3: Increase the CSR by 0.1.

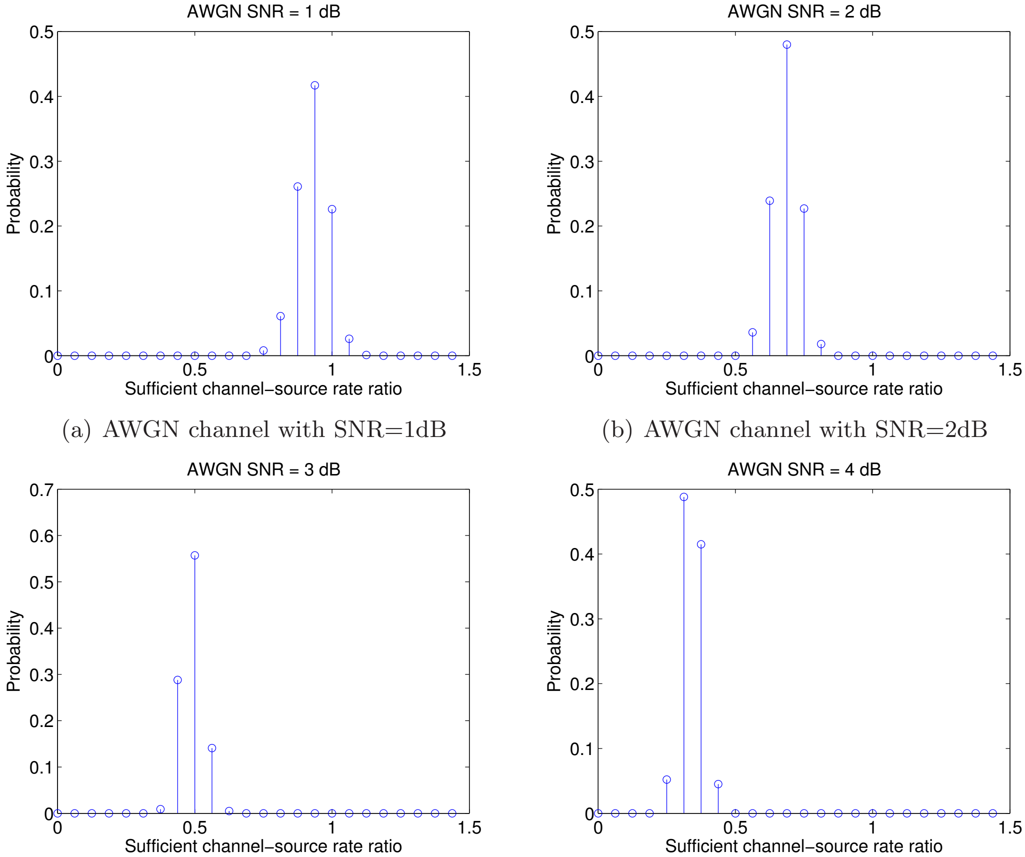
Step 4: Calculate the number of FEC bits, that is, $S \times CSR$, and apply FEC to the source bit stream.

Step 5: Simulate the transmission of the WZ slice, including both source and FEC bits, over an AWGN channel (SNR = 1, 2, 3, or 4 dB).

Step 6: Perform FEC decoding and WZ decoding and calculate SER between the original WZ slice and the decoded one.

Step 7: If SER equals zero, the current CSR is taken as the sufficient channel-source rate ratio for this slice; if SER is larger than zero, go to Step 3.

The experimental results are presented in Fig. 3.6. This figure shows the distribution of the sufficient channel-source rate ratio for an AWGN channel of different SNR (1, 2, 3 or 4 dB).



(a) AWGN channel with SNR=1dB

(b) AWGN channel with SNR=2dB

(c) AWGN channel with SNR=3dB

(d) AWGN channel with SNR=4dB

Figure 3.6: Distribution of the sufficient channel-source rate ratio Γ for different conditional entropy $H(X|Y)$.

Similar to the sufficient source coding rate S in Section 3.3.1, the data of mean and variance of the sufficient channel-source rate ratio Γ shown in Fig. 3.6 are stored in a lookup table in the encoder. If the AWGN channel state is known, mean and variance of Γ can be obtained by looking up in the table. Additionally, Γ is modeled as a Gaussian random variable and thus its probability density function $f_\Gamma(\gamma)$ is uniquely specified by its mean and variance.

One issue worth mentioning is the definition of the sufficient channel-source rate ratio for the WZVC scheme. This is not straightforward, because there is no explicit FEC. Therefore, it is an issue how to define the channel-source rate ratio. In the experiment, the total rate that ensures error-free decoding under a certain AWGN channel state is trained at the first stage. Then, the difference between the total rate and the sufficient source coding rate is taken as the equivalent rate for FEC. The ratio between this rate and the sufficient source coding rate is defined as the sufficient channel-source rate ratio equivalently. Experimental results show that the distribution of CSR for the WZVC scheme is similar to that for the WZVC+FEC scheme. Therefore, they are not differentiated in this thesis and they are represented by a single notation Γ.

To sum up, the uncertainty of the sufficient source coding rate S and the sufficient channel-source rate ratio Γ is discussed in Section 3.3. Both of them are modeled as a Gaussian random variable. The distribution, namely the mean and the variance, of the former one depends on the conditional entropy $H(X|Y)$, while that of the latter one depends on the SNR of the AWGN channel. Once the $H(X|Y)$ and the SNR of the AWGN channel are known, the probability density functions $f_S(s)$ and $f_\Gamma(\gamma)$ can be determined. How they contribute to the estimation of decoding failure probability for a WZ slice is presented in Section 3.4.3.

3.4 Joint source-channel rate control

The objective of joint source-channel rate control lies in minimizing the end-to-end distortion by allocating a target bit rate between source and channel coding. In this chapter, the focus is the WZ MBs, and therefore it is assumed that I-frames, Intra MBs, and the signaling of skip mode are error-free by all means. A certain number of WZ MBs, typically eight, are put together in a WZ slice, and a WZ slice is the smallest unit for joint source-channel rate control. For each WZ slice, a bit budget BB is specified, and BB is supposed to be allocated between source and channel coding for this slice. In practice, this rate allocation is realized through adaptive quantization.

3.4.1 Bit budget for a WZ slice

To begin with, the target bit rate in a unit of kilo bit per second (kbps) is converted to a bit budget in a unit of bit per pixel (bpp) for each GOP given a certain frame rate. Throughout this chapter, the frame rate is 15 frames per second (fps). The GOP structure is I-WZ-WZ-WZ-I and this structure means that each GOP starts with an I-frame and has three WZ-frames in the following. Next, the rate allocated to the I-frame is determined heuristically under the principle that the PSNR difference between an I-frame and its following WZ-frame is smaller than 2 dB. Then, the number of bits used for the I-frame is subtracted from the bit budget for the GOP and the residual is averaged over the three WZ frames in the GOP. In this way, the bit budget for each WZ-frame is determined. After this, mode decision is performed for each WZ-frame to identify intra MBs. The rate for signaling skip MBs is ignored. The rate for intra MBs is subtracted from the bit budget for the WZ-frame and the residual is averaged over all the WZ MBs in the frame. This practice leads to a bit budget for each WZ MB. Finally, this bit budget for each WZ MB multiplied by the number of MBs in each slice results in the bit budget BB for each slice.

It is true that BB is specified for each WZ slice before the joint source-channel coding is performed for this slice, but it can be updated during the coding process and thus can be different from slice to slice eventually. For instance, if one slice has very accurate side information and the transmission channel is pretty good, the bit budget BB cannot be used up. In this circumstance, the remaining bits are handed over to the next slice. In other words, the BB of the next slice will be higher than the one specified beforehand. This handover of the bit budget goes through all the WZ slices one after another until the end of each GOP.

3.4.2 Adaptive quantization

Given the BB for a WZ slice, rate allocation between source and channel coding is realized via adaptive quantization. As presented in Section 2.1.2, four quantizers are available (N_{BP} = 1, 2, 3, or 4). And the encoder needs to select one of these four candidate quantizers for each WZ slice. The quantizer can be different from slice to slice.

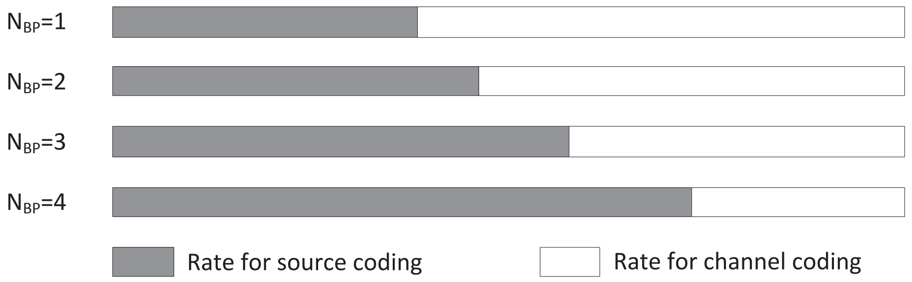


Figure 3.7: Rate allocation between source and channel coding using adaptive quantization.

Adaptive quantization is illustrated in Fig. 3.7. If a coarse quantizer, for example N_{BP} = 1, is selected, the quantization distortion is large. At the same time, the sufficient source coding rate is small. This means that many bits are available for channel coding, the bit stream is robust against channel noise and the resulting channel-induced distortion is small. In contrast, if a fine quantizer, for example $N_{BP} = 4$, is selected, the quantization distortion is small. The sufficient source coding rate, however, tends to be large. This means that fewer bits can be used for channel coding and thus the channel-induced distortion might be larger. Adaptive quantization aims at selecting the quantizer that leads to the optimal trade-off between quantization distortion and channel induced distortion, so that the end-to-end distortion D_e is minimized. The problem of adaptive quantization is formulated in (3.2). Herein, an underlying constraint is the BB for each slice.

$$N_{BP}^* = \underset{N_{BP} \in \{1,2,3,4\}}{\arg\min} D_e(N_{BP}). \tag{3.2}$$

N_{BP} is connected to the end-to-end distortion D_e via the decoding failure probability. More specifically, N_{BP} influences the conditional entropy of the WZ slice. This is reflected in (3.1). Moreover, the conditional entropy influences the distribution of the sufficient source coding rate S. Furthermore, this distribution $f_S(s)$ is needed to estimate the decoding failure probability. It is discussed in Section 3.4.3 how to estimate the decoding failure probability for a given N_{BP}. Afterwards, how decoding failure probability influences the end-to-end distortion for a given N_{BP} is presented in Section 3.4.4.

3.4.3 Decoding failure probability

The decoding failure probability for a WZ slice X can be calculated, if the conditional entropy $H(X|Y)$, the AWGN channel state and the bit budget BB are given. The conditional entropy $H(X|Y)$ depends on N_{BP}. For a given N_{BP}, the $H(X|Y)$ is calculated using (3.1). Based upon the $H(X|Y)$, the distribution of the sufficient source coding rate $f_S(s)$ can be determined, as presented in Section 3.3.1. In parallel, $f_\Gamma(\gamma)$ can be determined according to the SNR of the AWGN channel, as presented in Section 3.3.2. Finally, $f_S(s)$, $f_\Gamma(\gamma)$ and BB are directly used to estimate the decoding failure probability.

In the WZVC+FEC scheme, source coding and channel coding are carried out at two separate stages. The decoding is successful, if both the FEC decoding and the Wyner-Ziv video decoding are successful. The decoding failure probability is calculated using the following equation:

$$p_{fec} = 1 - \int_0^R f_S(s)ds \cdot \int_0^{\frac{BB-R}{R}} f_\Gamma(\gamma)d\gamma. \tag{3.3}$$

In (3.3), R represents the rate allocated for source coding and thus $\frac{BB-R}{R}$ is the channel-source rate ratio. Because source and channel coding are separate, the source coding rate R needs to be determined explicitly. The optimal source coding rate R^* is supposed to minimize the decoding failure probability, as formulated in (3.4).

$$R^* = \arg\min_R p_{fec}(R). \tag{3.4}$$

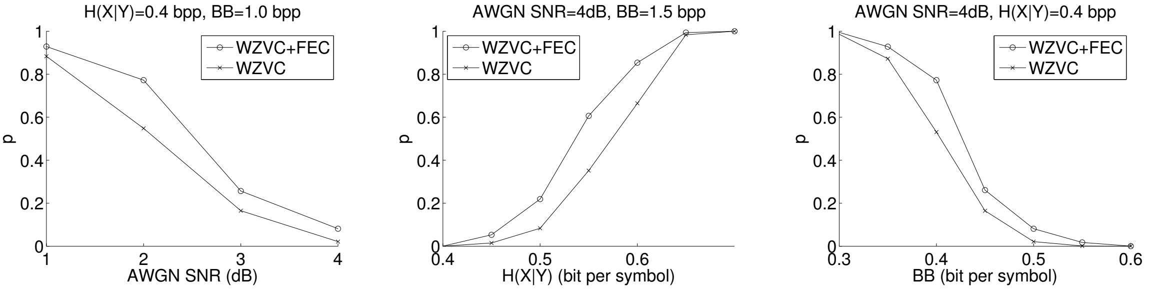
In comparison, it is not necessary to specify the source coding rate explicitly in the WZVC scheme, because source coding and channel coding are carried out at a single stage. Therefore, the decoding failure probability only depends on BB:

$$p_{wzvc} = 1 - \int_0^{BB} (\int_0^{\frac{BB-s}{s}} f_\Gamma(\gamma)d\gamma)\cdot f_S(s)ds. \tag{3.5}$$

By comparing (3.3) and (3.5), p_{fec} proves larger than or equal to p_{wzvc}, because

$$\begin{aligned} & \textstyle\int_0^R f_S(s)ds\cdot \int_0^{\frac{BB-R}{R}} f_\Gamma(\gamma)d\gamma \\ \leq & \textstyle\int_0^R (\int_0^{\frac{BB-s}{s}} f_\Gamma(\gamma)d\gamma)\cdot f_S(s)ds \\ \leq & \textstyle\int_0^{BB} (\int_0^{\frac{BB-s}{s}} f_\Gamma(\gamma)d\gamma)\cdot f_S(s)ds, \end{aligned} \tag{3.6}$$

and this inequality fundamentally justifies why WZVC is more robust than WZVC+FEC. Intuitively speaking, WZVC+FEC requires an explicit decision upon the source rate, which might be inadequate, thus ends up with a higher decoding failure probability. Of course, an underlying assumption is that the distribution of the channel-source rate ratio for the WZVC scheme is identical to that of the WZVC+FEC scheme. It remains a question whether this assumption is always true, but the experimental results obtained using a large set of training data, presented in Section 3.3.2, indicate that this assumption is true for the Wyner-Ziv video codec and the AWGN channel model discussed in this thesis.



(a) decoding failure probability vs. AWGN channel state

(b) decoding failure probability vs. conditional entropy

(c) decoding failure probability vs. bit budget

Figure 3.8: Decoding failure probability

Figure 3.8 shows some typical examples of decoding failure probability. The decoding failure probability depends on the AWGN channel state, $H(X|Y)$ and BB. Figure 3.8(a) shows that the decoding failure probability decreases when the SNR of the AWGN channel increases. According to Figure 3.8(b), the larger the conditional entropy $H(X|Y)$, the higher the decoding failure probability. It can be observed in Figure 3.8(c) that the decoding failure probability drops if BB becomes larger. These properties are true for both WZVC and WZVC+FEC. In comparison, however, the decoding failure probability in WZVC is always smaller than or equal to that in WZVC+FEC.

Above all, for the purpose of selecting the optimal quantizer, the encoder needs to estimate the decoding failure probability for each optional quantizer (N_{BP} = 1, 2, 3 or 4). The decoding failure probability plays an important role in estimating the end-to-end distortion. The estimation of the end-to-end distortion is discussed in detail in Section 3.4.4.

3.4.4 End-to-end distortion estimation

The mean end-to-end distortion of a WZ slice involves two parts, namely quantization distortion and channel-induced distortion. In the WZVC scheme as well as the WZVC+FEC scheme, a CRC is applied to the decoded quantization indices of each MB in the slice. If the CRC is fine, the decoded quantization indices are assumed to be correct and thus used to reconstruct each MB together with its side information. The mean square error (MSE) between the original and the reconstructed MB is referred to as the quantization distortion $D_{N_{BP}}$ of this MB. Herein, N_{BP} identifies the quantizer. If the CRC fails, the decoded quantization indices have errors and thus are discarded. Side information of the MB is directly taken as the reconstructed MB. In this instance, the distortion is the MSE between the original MB and its side information. This distortion is referred to as the channel-induced distortion D_0.

The process of estimating the end-to-end distortion for the WZVC scheme is very similar to that for the WZVC+FEC scheme, only the decoding failure probability of the two schemes might be different. The mean end-to-end distortion D_e of a WZ slice for a given N_{BP} is estimated in the following steps:

Firstly, the decoding failure probability p is estimated based on the BB. For the WZVC scheme, p represents p_{wzvc}, and is calculated according to (3.5) in Section 3.4.3; for the WZVC+FEC scheme, p represents p_{fec}, and is calculated according to (3.3) in Section 3.4.3.

Secondly, the mean end-to-end distortion of every MB in the slice is estimated. For instance the kth one in the slice, has the end-to-end distortion in the following form:

$$D_e^{(k)}(N_{BP}) = p \cdot D_0 + (1-p) \cdot D_{N_{BP}}, \tag{3.7}$$

Finally, averaging the mean end-to-end distortion of all the MBs in a slice results in the mean end-to-end distortion of the slice for a given :

$$D_e(N_{BP}) = \sum_k D_e^{(k)}(N_{BP})/K. \tag{3.8}$$

Above all, the mean end-to-end distortion estimation for a given N_{BP} is presented in this section. The encoder needs to estimate the mean end-to-end distortion for all the candidate quantizers ($N_{BP} = 1$, 2, 3 or 4) before carrying out the adaptive quantization presented in Section 3.4.2.

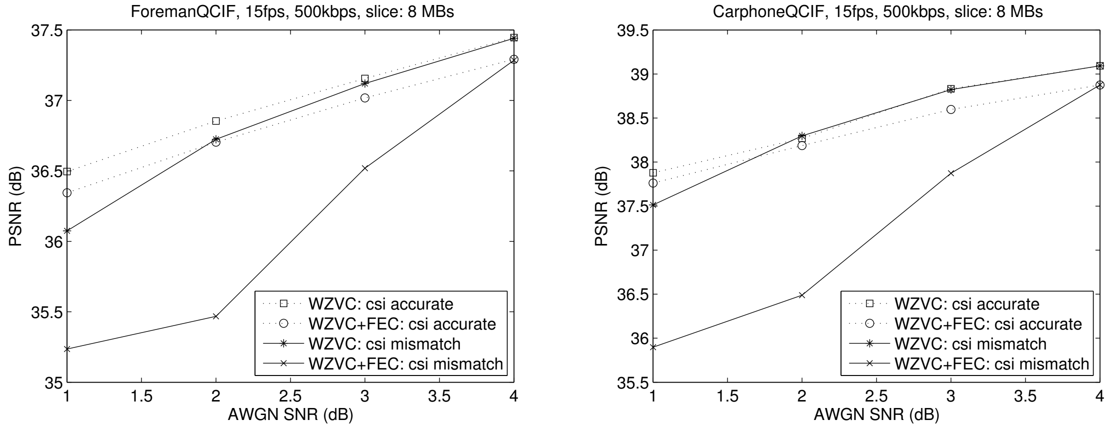
3.5 Experimental results

WZVC and WZVC+FEC are compared in terms of error-resilience in two circumstances, so that exhaustive knowledge about their performance can be acquired. First, they are compared under the assumption that the estimation the transmission channel state, namely the SNR of the AWGN channel, is completely precise. In this case, both schemes can achieve their optimal error-resiliency and it is their optimal performances that are compared. In the second case, it is assumed that the transmission channel state is worse than estimated and thus the performance of both schemes deteriorates. The objective of the comparison in this case is to find out which scheme has a more graceful performance degradation.

The Foreman sequence and the Carphone sequence, both in the QCIF format (176×144) and with a frame rate of 15 fps, are used for the simulation. To achieve a reasonable target quality, the joint source-channel rate control targets at a bit rate of 500 kbps for both video sequences. This target bit rate includes the rate for both source and channel coding. How this target rate is eventually allocated to each individual WZ slice has been discussed in Section 3.4.1.

3.5.1 End-to-end distortion with accurate channel state information

In this experiment, it is assumed that the encoder knows perfectly the channel state information (CSI), thus can perform adaptive quantization optimally for each slice. The reconstructed video quality of the two schemes ("WZVC: csi accurate" and "WZVC+FEC: csi accurate") are presented in Figure 3.9. They perform almost equally well. Nevertheless, the WZVC+FEC is always a bit worse. After all, it has a higher decoding failure probability, thus ends up with a lower video quality.



(a) Foreman (b) Carphone

Figure 3.9: Comparison of the end-to-end distortion for WZVC and WZVC+FEC at the same target bit rate.

3.5.2 End-to-end distortion with channel mismatch

In this simulation, the encoder performs rate control assuming that the SNR of the AWGN channel is 4 dB, when the resulting bit stream is transmitted under worse channel states. The performance comparison between the two schemes ("WZVC: csi mismatch" and "WZVC+FEC: csi mismatch") is presented in Figure 3.9. The video quality in both schemes drops if the real channel condition is worse than the encoder expects. Comparatively speaking, WZVC is significantly more robust and outperforms WZVC+FEC by up to 1 dB.

3.6 Chapter summary

This chapter refers to error resilient video transmission using WZVC. The objective is to minimize the mean end-to-end distortion at a target bit rate. Therefore, this problem is formulated as a joint source-channel rate control problem and the rate control is implemented using slice-wise adaptive quantization.

One challenging issue in this joint source-channel rate control problem lies in the uncertainty of the sufficient rates. Therefore, sufficient source coding rate and channel-source rate ratio are modeled as Gaussian random variables. This practice is a major contribution in this chapter and enables the estimation of the decoding failure probability and thus the mean end-to-end distortion of each slice.

Another contribution presented in this chapter lies in the comparison of WZVC and WZVC+FEC in terms of error resiliency. Experimental results and theoretical analysis lead to the conclusion that WZVC is significantly more error-resilient than WZVC+FEC.

Chapter 4

Wyner-Ziv Bayer-pattern video coding

This chapter refers to lossy compression of Bayer-pattern video using Wyner-Ziv video coding. The motivation is to achieve low encoding complexity. Firstly, Bayer-pattern video will be introduced in Section 4.1. The state-of-the-art demosaicing methods and compression schemes will also be presented in this section.

Next, a novel color space transform (CST) scheme, the major contribution of this chapter, will be presented in Section 4.2. As a matter of fact, there are two variants of the proposed scheme, namely B-4:2:0 and B-4:2:2. These two variants will be discussed one by one, after the conventional scheme is reviewed.

Then, the Bayer-pattern video compression using H.264/AVC will be presented in Section 4.3 and Wyner-Ziv Bayer-pattern video compression will be presented in Section 4.4. Afterwards, the experimental results will be presented in Section 4.5. Results for Bayer-pattern video coding using H.264/AVC as well as WZVC will be shown and analyzed. Finally, this chapter will be summarized in Section 4.6.

4.1 State of the art

Most consumer digital color cameras are equipped with a single chip. In such cameras, a color filter array is used to capture only one color component per pixel instead of a RGB triple. Bayer pattern [Bay76] is currently the most popular color filter array used in single chip cameras. Conventionally, missing color components at each pixel are interpolated from its neighboring pixels, so that full color images are constructed. This process is typically referred to as demosaicing and is illustrated in Fig. 4.1. Only the full color images can be correctly displayed.

The conventional compression scheme for Bayer-pattern video is shown in Fig. 4.2. The first step at the encoder is demosaicing. Next, the full color images in the RGB domain

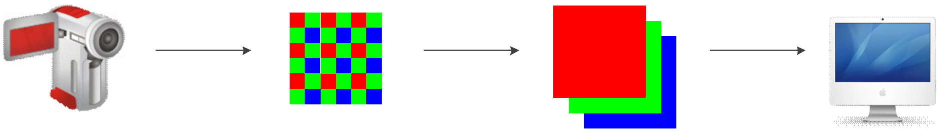


Figure 4.1: Capturing, demosaicing and displaying of Bayer-pattern images

are converted to the YUV domain via color space transform. Typically, U and V pixels are subsampled by a factor of two both horizontally and vertically, which results in a format of YUV 4:2:0. Then, the resulting YUV video data are forwarded to an H.264/AVC video codec for compression. After the YUV data have been decoded, the missing U and V pixels are interpolated. Finally, the YUV data are converted back to full color images in the RGB domain. This conventional Bayer-patter video coding system is referred to as H264-C-4:2:0 in the following.

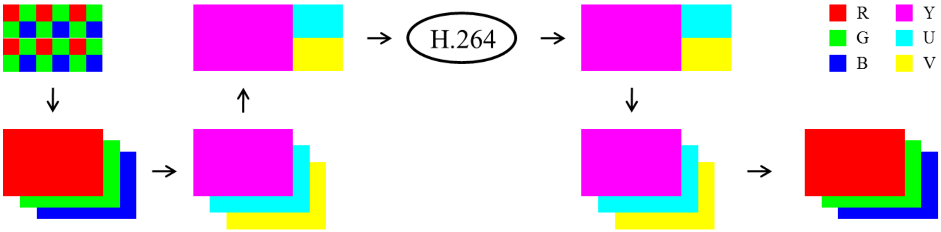


Figure 4.2: Conventional system H264-C-4:2:0 for Bayer-pattern video compression

Simplicity is the most significant advantage of the conventional approach. There are existing or standard methods for all the main steps, including demosaicing, color space transform, chroma subsampling and H.264/AVC video coding. However, such a simple combination of different techniques limits the coding efficiency. Therefore, it is necessary to look at the complete system and optimize its components. And the major contribution of this chapter lies in a novel color space transform (CST) scheme, which will be discussed in detail in Section 4.2.

In the literature, there has been abundant work in demosaicing of Bayer-pattern images and the motivation is to improve the objective as well as subjective quality of the reconstructed images. A few typical ones will be described in Section 4.1.1. In comparison, Bayer-pattern image and video compression is not so well-studied, and just a few methods have been proposed. These methods will be presented in Section 4.1.2.

4.1.1 Bayer-pattern demosaicing

At the beginning of this section, it is worth pointing out a good overview paper for demosaicing. Gunturk et al. list quite a few typical and popular schemes and make a comprehensive performance comparison in [GGA$^+$05b]. These schemes and related work will be discussed first, and then a few more recent and advanced techniques will be discussed.

The classic and most simple demosaicing schemes include nearest-neighbor, bilinear and cubic interpolation methods [LYDB02]. Particularly, bilinear interpolation proves quite efficient and can be performed pretty fast. Moreover, Pei and Tam propose to interpolate the color difference instead of applying bilinear interpolation directly to the images [PT00b].

Edge-directed interpolation or gradient based methods are discussed in [LP94] and [CCP99]. The second derivative of the red and blue channels are used to estimate the edge direction in the green channel, when the interpolation is performed along the edge instead of across the edge. The green channel is used to compute the missing values in the red and blue channels. A variant of this method is proposed in [AH97, HA97], where the second derivative of the green channel and the first derivative of the red or the blue channel are used to estimate the edge direction in the green channel. This is also well-known as Laplace interpolation. Based on a similar principle, several other gradient based demosaicing schemes have been proposed [KOY03, KR03].

Smooth hue transition methods [Cok87, CE87, Ada95] are also a set of typical demosaicing schemes, which exploit inter-channel correlation. It is assumed that the independent interpolation of the green channel is more reliable than that of the red and the blue channel, because twice as many green samples are available in the Bayer pattern. A second assumption lies in the hue similarity of neighboring pixels. Therefore, missing pixels are interpolated such that the ratio of green, red and blue intensity values are similar for neighboring pixels. Interestingly, the smooth hue transition methods is combined with gradient based methods in [Kim99b].

Different from the above mentioned schemes which can be carried out via one or two stages, there have been some more sophisticated iterative methods. In [CT02], spatial and spectral correlations among neighboring pixels are exploited to define the interpolation step, while adaptive median filtering is used as the enhancement step. Additionally, iterative maximum a posteriori (MAP) methods are proposed in [KO99, HOK02].

Interestingly, a joint spatial and temporal method is proposed in [WZ05]. A pixel of the current frame is matched to another in a reference frame via motion analysis, such that the CCD sensor samples different color components of the same object position in the two frames. The resulting inter-frame estimates of missing color components are fused with suitable intra-frame estimates to achieve a more robust color restoration. Apart from the above mentioned methods, there have been quite a few other schemes [WCB97, ZW05, WZ04, HP03, LP04, LMP03, LPHA04, KB02, RS03].

4.1.2 Bayer-pattern image and video compression

Koh et al. look into the possibility of compressing Bayer-pattern images using JPEG in [KMM03]. They maintain that compressing Bayer-pattern images directly in the RGB domain is inefficient. As a remedy, the authors propose to apply a color space transform and compress images in the YCbCr domain using JPEG, as shown in Fig. 4.3. Finally, the proposed scheme is tested with three demosaicing methods, including bilinear, cubic and Laplace interpolation and proves more efficient than compressing Bayer-pattern image directly in the RGB domain.

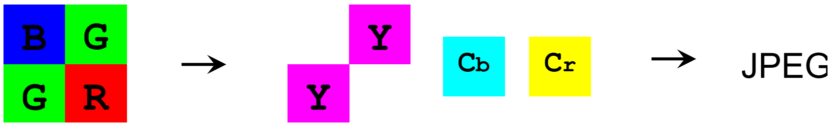


Figure 4.3: Color space transform for Bayer-pattern image compression

Additionally, there have been some publications on Bayer-pattern video compression. In [GKC+05], the red, green and blue pixels in the Bayer pattern are separated into three arrays and then an MPEG-2 like video coder is used for compression. This method is supposed to have a limited coding efficiency for P-frames because severe aliasing is generally contained in Bayer pattern. To alleviate aliasing, it is proposed in [DN06, DNP08] that Bayer-pattern videos are compressed using an H.264/AVC video coder and the motion compensation is adapted to the Bayer pattern. However, these two schemes are both confined to the RGB domain and, due partly to this, outperform the conventional method only at relatively high bit rates. Instead, transforming RGB data to YUV data and performing compression in the YUV domain generally leads to a higher coding efficiency. When it comes to Bayer-pattern compression, finding a good color space transform is a non-trivial and interesting issue and will be discussed thoroughly in Section 4.2.

4.2 Color space transform

The major contribution of this chapter lies in the proposed color space transform (CST), and more specifically the chroma subsampling. Conventional Bayer-pattern video compression begins with demosaicing or color interpolation, as illustrated in Figure 4.2. Then full color RGB images are converted to the YUV domain. After this, chroma subsampling is applied to the U and V components by a factor of two both horizontally and vertically, as shown in Figure 4.4, so that the YUV data are available in the standard format 4:2:0. This conventional CST method is referred to as C-4:2:0 in the following. Now the YUV data are compressed using an H.264/AVC video coder. The decoder reconstructs the YUV data in the format 4:2:0 and then interpolates the U and V components to their full size. Finally, the YUV images are converted back to RGB images. The color space transform and its inverse are described for instance in Keith Jack's book [Jac07]:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix}, \tag{4.1}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.164 & 0 & 1.596 \\ 1.164 & -0.813 & 0.391 \\ 1.164 & 2.018 & 0 \end{bmatrix} \begin{bmatrix} Y-16 \\ U-128 \\ V-128 \end{bmatrix}. \tag{4.2}$$

Now that the U and V components are subsampled, a central issue is which U or V samples should be taken such that the reconstructed video ends up with the highest possible quality. The conventional Bayer-pattern compression takes the nominal chroma sample positions standardized in ITU-T recommendation H.264/AVC, as illustrated in Figure 4.4. U and V samples lie halfway between Y samples. The standard supports alternative chroma sample positions too. But all standardized chroma sample positions have in common that U and V are always at the same location. Although this practice has been taken for granted, it actually results in a performance loss for Bayer-pattern video compression. More reasonable chroma sample positions are chosen in the proposed CST scheme B-4:2:0, which will be presented in Section 4.2.1.

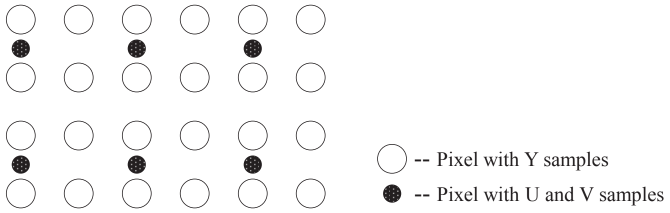


Figure 4.4: Chroma subsampling for YUV 4:2:0

Another issue lies in the number of Y samples taken and forwarded to the video coder. Conventionally, all the Y samples are kept. This option is inherited in the proposed method B-4:2:0. In this way, each two by two pixel array in the original RGB Bayer-pattern images results in four Y samples, one U pixel and one V pixel. More specifically, the number of samples increases by two after the color space transform. In comparison, the number of samples remains the same before and after the color space transform when only two Y samples are taken. This option is taken in the proposed method B-4:2:2. The proposed method B-4:2:2 leads to a solution of low complexity, because the number of Y samples needed to be compressed by the video coder is reduced by half compared to that in the conventional method C-4:2:0 and the proposed method B-4:2:0.

One thing worth mentioning lies in the demosaicing scheme. In Section 4.2.1 and Section 4.2.2, the discussion is confined to the bilinear interpolation. The proposed color space conversion methods only take different positions for U and V in the chroma subsampling and do not require any change in the demosaicing method, and therefore they are also applicable to other demosaicing methods, for instance the Laplace interpolation.

4.2.1 CST method B-4:2:0

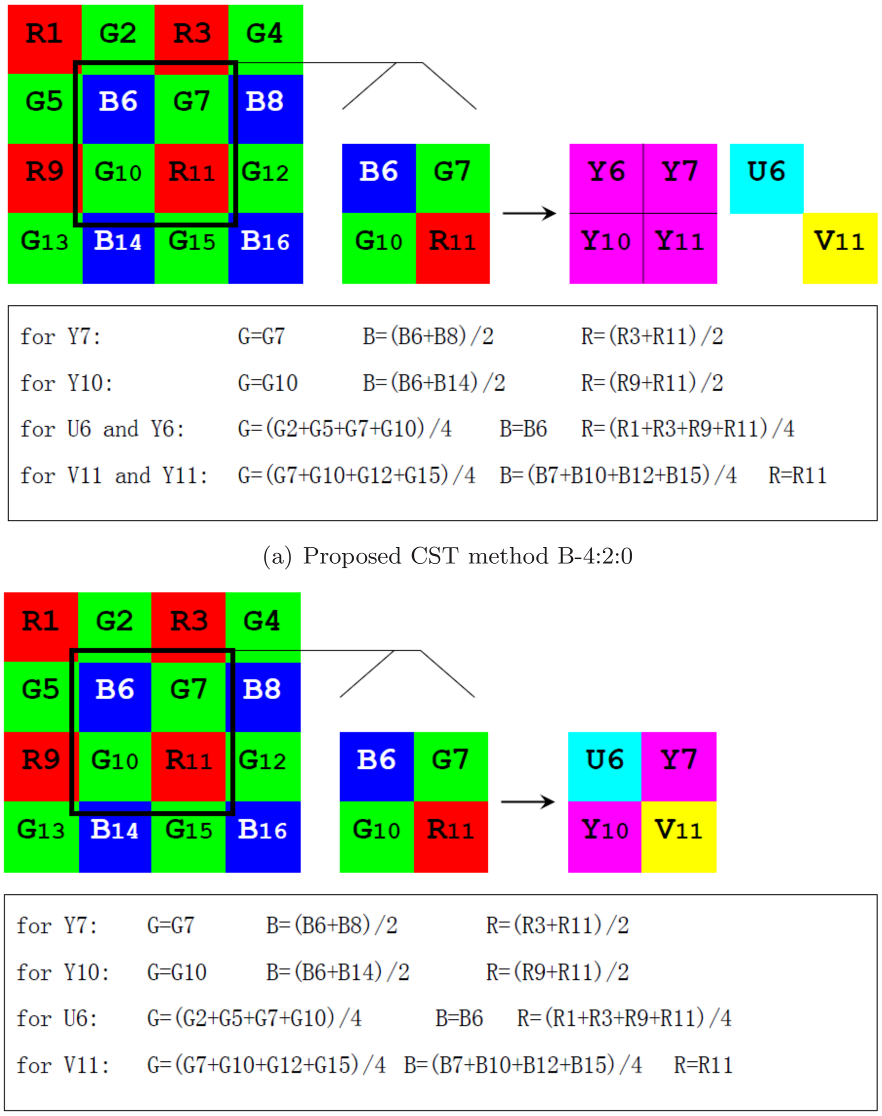
The proposed method B-4:2:0 for color space transform is illustrated in Figure 4.5(a). Y samples are calculated at all the positions in the Bayer-pattern image. When it comes to chroma samples U and V, the positions to calculate them are carefully chosen. V values are calculated at the positions of R pixels, and U values at B pixels. This way of selecting the chroma sample positions can be justified by looking into the equations in (4.2). In the inverse color space transform, only Y and V are needed to reconstruct R pixels. This means, V is more important than U at positions of R pixels and this is why V values are calculated. For the same reason, U values are calculated where B pixels are located. The different positions selected for U and V samples in the proposed scheme B-4:2:0 are more reasonable for the reconstruction of R and B pixels than the standardized chroma sample positions in the conventional method. The standard chroma subsampling takes U and V always at the same location without taking into account the different positions of R and B pixels in the Bayer pattern, thus it is non-optimal for Bayer-pattern image and video compression. In comparison, the chroma sample positions in the proposed scheme B-4:2:0 are chosen by considering the color space transform at the encoder and the inverse color space transform at the decoder jointly, and thus can lead to a better rate-distortion performance of Bayer-pattern video compression than the conventional method C-4:2:0.

4.2.2 CST method B-4:2:2

The proposed method B-4:2:2 is designed for a low complexity solution for Bayer-pattern video compression. The chroma subsampling in the proposed method B-4:2:2 is identical to that in the proposed method B-4:2:0, but the number of Y samples is halved. Precisely speaking, only the Y samples at the positions of G pixels are calculated and those at R and B pixels are discarded. Now that the number of Y samples to compress are halved, the encoding time for video compression can be reduced by almost 50%. The color space transform of this method based on demosaicing using bilinear interpolation is illustrated in Figure 4.5(b). And how the resulting YUV data can be converted to a rectangular grid appropriate for H.264 video coding will be discussed in Section 4.3.2.

4.3 Bayer-pattern video compression using H.264/AVC

In Section 4.2.1 and Section 4.2.2, the two proposed CST methods B-4:2:0 and B-4:2:2 have been presented. In this section, the two variants of the Bayer-pattern video compression sys-



for Y7: G=G7 B=(B6+B8)/2 R=(R3+R11)/2

for Y10: G=G10 B=(B6+B14)/2 R=(R9+R11)/2

for U6 and Y6: G=(G2+G5+G7+G10)/4 B=B6 R=(R1+R3+R9+R11)/4

for V11 and Y11: G=(G7+G10+G12+G15)/4 B=(B7+B10+B12+B15)/4 R=R11

(a) Proposed CST method B-4:2:0

for Y7: G=G7 B=(B6+B8)/2 R=(R3+R11)/2

for Y10: G=G10 B=(B6+B14)/2 R=(R9+R11)/2

for U6: G=(G2+G5+G7+G10)/4 B=B6 R=(R1+R3+R9+R11)/4

for V11: G=(G7+G10+G12+G15)/4 B=(B7+B10+B12+B15)/4 R=R11

(b) Proposed CST method B-4:2:2

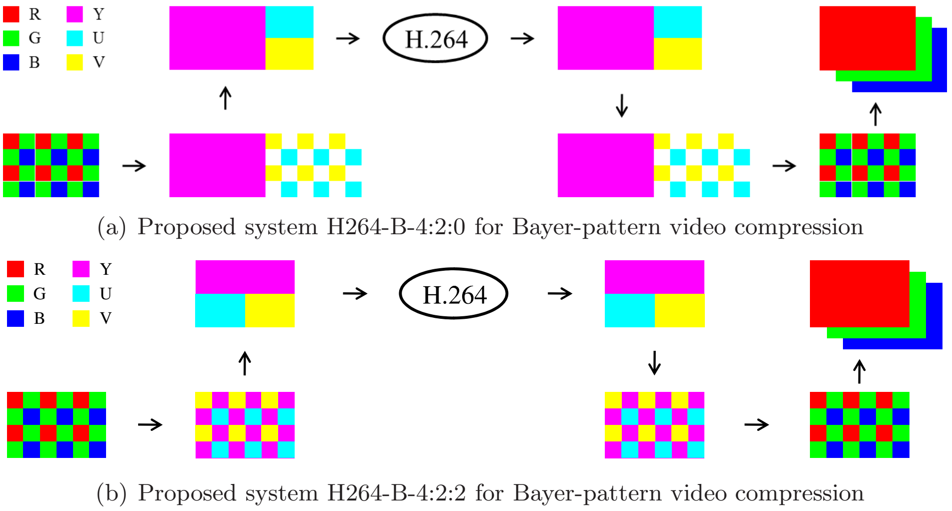
Figure 4.5: Proposed CST methods for Bayer-pattern video compression. The method B-4:2:0 leads to the best video quality while the method B-4:2:2 can be used for a low complexity solution.

tem using H.264/AVC will be presented. Because the fundamental difference of the proposed systems and the conventional one lies in the color space transform, the naming convention for the proposed CST methods is reused for the proposed Bayer-pattern video compression systems. More specifically, the first one is referred to as the proposed system H264-B-4:2:0, because the CST method B-4:2:0 is deployed in this system. Similarly, the second one is referred to as the proposed system H264-B-4:2:2.

The fundamental difference of the two proposed systems lies in the number of Y samples. On one hand, the best video quality can be achieved by keeping all the Y samples in the method B-4:2:0. In this instance, the YUV data appear in a format similar to 4:2:0. On the other hand, almost half of the computation can be saved by keeping only half of the luma samples in the method B-4:2:2, if a small loss in video quality is tolerable. This method results in a format similar to YUV 4:2:2.

4.3.1 Bayer-pattern video coding scheme H264-B-4:2:0

The proposed system H264-B-4:2:0 is illustrated in Figure 4.6(a). After the CST method B-4:2:0 is performed, the resulting YUV data end up in a format of 4:2:0 and are written into a YUV file according to this format. Then, the H.264/AVC video coder is configured for this format and used to compress the YUV data. At the decoder, the reconstructed U and V components are interpolated using bilinear interpolation back to the original size. Next, the YUV data are used to reconstructed the Bayer-pattern images in the RGB domain. Finally, demosaicing is applied to the Bayer-pattern images to reconstruct the full color RGB images.



(a) Proposed system H264-B-4:2:0 for Bayer-pattern video compression

(b) Proposed system H264-B-4:2:2 for Bayer-pattern video compression

Figure 4.6: Proposed systems H264-B-4:2:0 and H264-B-4:2:2 for Bayer-pattern video compression using H.264/AVC. The former one ensures the best video quality while the latter one reduces the computational complexity almost by half.

One thing worth mentioning is that the U and V samples in this system are at different locations. U samples are taken at the locations of blue pixels and V samples at the locations of red pixels. This information needs to be signalled to the decoder, so that the interpolation of the U and V components can be performed correctly. Nevertheless, the H.264/AVC video coder is kept standard, because it does not require the exact locations of U and V samples. The process of video coding remains the same, even if U and V samples come from different locations.

4.3.2 Bayer-pattern video coding scheme H264-B-4:2:2

The proposed system H264-B-4:2:2 is illustrated in Fig 4.6(b). Different from the system H264-B-4:2:0, the Y samples in the system H264-B-4:2:2 form a quincunx pattern as G pixels do in the Bayer-pattern RGB images. Therefore, it is necessary to add in a step of structure conversion [KMM03], which converts this quincunx pattern to a rectangular one, in order that the H.264/AVC video coder works correctly. Such structure conversion is shown in Fig 4.7. Y samples in the even rows are moved one unit upward, before the resulting complete rows of Y samples are pushed together to a rectangular array. The chroma samples are also pressed together. After this structure conversion, the YUV data match the standard format 4:2:2 and are written into a YUV file according to this format. The H.264/AVC video coder, after being configured for the YUV format 4:2:2, can be used to compress the YUV data.

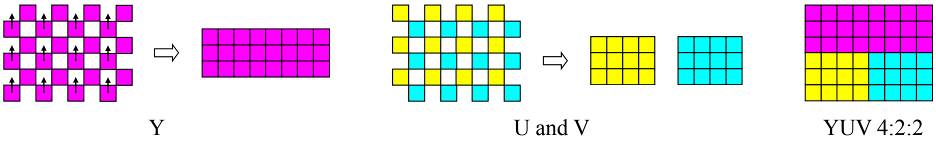


Figure 4.7: Structure conversion for YUV data in the proposed system H264-B-4:2:2

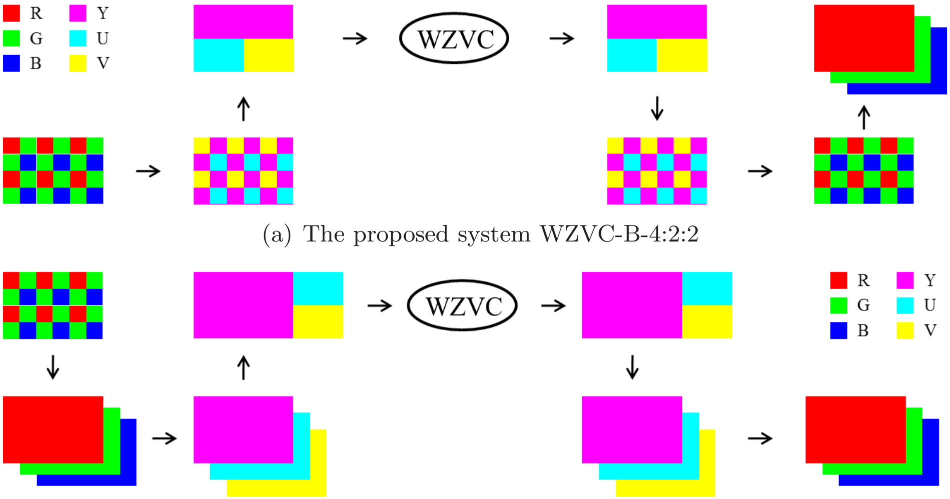
At the decoder, the reconstructed Y component in a rectangular pattern needs to be converted back to the quincunx pattern. Then the missing Y samples are interpolated using linear interpolation. At the same time, the U and V components are also interpolated to the original size using bilinear interpolation. Afterwards, the YUV data are converted to Bayer-pattern images in the RGB domain. Finally, full color RGB images are reconstructed by demosaicing the Bayer-pattern images.

One issue worth mentioning lies in the influence of the structure conversion on the coding efficiency, particularly on the luma samples. The structure conversion breaks the original sample grid and thus introduces unwanted frequency. Doutre and Nasiopoulos maintain that the resulting aliasing effect reduces the efficiency of motion estimation and compensation (MEC) and thus modify the MEC algorithm in the H.264/AVC to cancel out the aliasing effect [DN06, DNP08]. Their work is confined to the compression directly in the RGB domain. When this method is reproduced in the proposed method H264-B-4:2:2 in the YUV domain, the coding efficiency can be slightly improved and the composite PSNR (CPSNR) of the reconstructed full color RGB images can increase by up to 0.2 dB approximately. This improvement is limited, but breaks the H.264/AVC standard in the aspect of MEC. Therefore, the method of modifying MEC for Bayer-pattern video compression is not considered in this thesis, and the proposed scheme H264-B-4:2:2 is compatible with the H.264/AVC standard.

4.4 Wyner-Ziv Bayer-pattern video coding

The motivation of compressing Bayer-pattern video using WZVC is to achieve low encoding complexity. The proposed Wyner-Ziv Bayer-pattern video coding system, referred to as WZVC-B-4:2:2, is illustrated in Fig. 4.8(a). Compared with the proposed system H264-B-4:2:2 in Fig. 4.6(b), the only difference is that WZVC instead of H.264/AVC is used for video compression.

The proposed system WZVC-B-4:2:2 has a very low encoding complexity. For one thing, the proposed CST method B-4:2:2 reduces the number of Y samples by half compared to the method C-4:2:0, and thus the computational complexity for video compression, including both encoding and decoding, is reduced almost by half. For another, the Wyner-Ziv video encoder is also of low encoding complexity, because motion estimation can be shifted partly or completely to the decoder. Above all, the combination of the proposed CST method B-4:2:2 and the Wyner-Ziv video codec leads to a Wyner-Ziv Bayer-pattern video coding system of very low encoding complexity.



(a) The proposed system WZVC-B-4:2:2

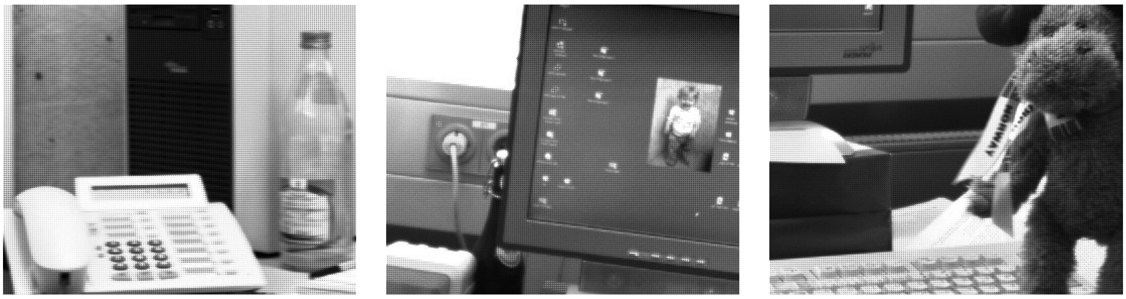
(b) The comparison system WZVC-C-4:2:0

Figure 4.8: Wyner-Ziv Bayer-pattern video compression systems

Apart from the complexity issue, the coding efficiency needs to be investigated, too. For this purpose, a comparison system is introduced and shown in Fig. 4.8(b). This comparison system is referred to as WZVC-C-4:2:0. Herein, the conventional CST method C-4:2:0 is used. The motivation of comparing WZVC-B-4:2:2 to WZVC-C-4:2:0 is to verify that the proposed CST method B-4:2:2 can lead to a higher video coding efficiency than the conventional method C-4:2:0, if WZVC is used for Bayer-pattern video compression.

4.5 Experimental results

Three Bayer-pattern video sequences have been captured and used for simulating Bayer-pattern video compression. These three videos represent three different motion modes, panning, zooming and an object moving in front of a static background. The video frames have been captured in the CIF format (352×288) and at a frame rate of 25 fps. The first 39 frames of each sequence are used for simulation. Screenshots of selected frames of the three test videos are shown in Figure 4.9.



(a) Panning (b) Zooming (c) Moving Object

Figure 4.9: Screenshots for test sequences

The mean composite peak-signal-to-noise ratio (CPSNR) is used as the measure for the reconstructed video quality throughout this chapter. Every original Bayer-pattern image and its reconstructed counterpart are interpolated to full color RGB images and the CPSNR is calculated between them. The CPSNR values of all the images in a sequence are averaged. The equations (4.3) and (4.4) are used to calculate CPSNR of an image, as defined in [MLM02a]. Herein, $I(i,j,k)$ represents the pixel intensity at position (i,j) of the k-th color component for the reference image and $I'(i,j,k)$ for the reconstructed image. M and N are height and width of the frame.

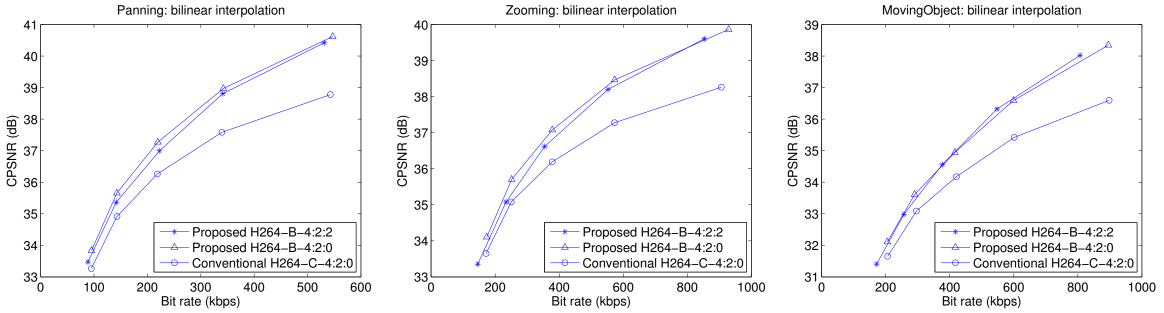
$$CPSNR = 10log_{10}\frac{255^2}{MSE} \tag{4.3}$$

$$MSE = \frac{1}{3MN}\sum_{k=1}^{3}\sum_{i=1}^{N}\sum_{j=1}^{M}\left[I(i,j,k) - I'(i,j,k)\right]^2 \tag{4.4}$$

4.5.1 Bayer-pattern video compression using H.264/AVC

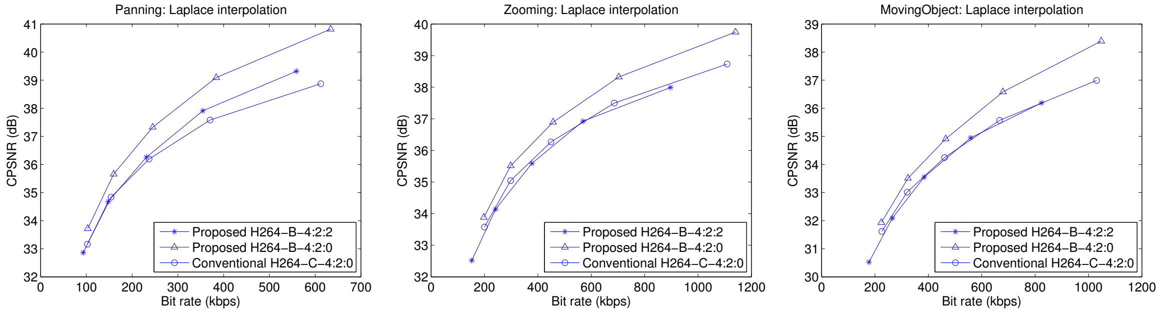
The H.264/AVC reference software JM 17.2 is used for video compression in our simulation. Now that 39 frames of each sequence are simulated, the GOP length is set to be 39 for simplicity. The first frame in the GOP is an I-frame and all the 38 following frames are P-frames. The YUV format is set to 4:2:0 or 4:2:2 according to different simulations. As for other parameters in the configuration file of the JM coder, their default values are taken.

In a first experiment, bilinear interpolation is used for demosaicing. Linear interpolation is used for the R, B, U and V components. To interpolate a missing G pixel, bilinear interpolation is applied, as shown in Figure 4.5. The interpolation method for the Y component for the proposed method B-4:2:2 is the same as that for the G component. As to the chroma subsampling in the conventional case, U and V samples are prefiltered before being sampled both horizontally and vertically by a factor of two. Accordingly, postfiltering is applied for the interpolation of the U and V components at the decoder. Rate-distortion curves for different methods using bilinear interpolation and for different test sequences are plotted in Figure 4.10.



(a) Panning (b) Zooming (c) Moving Object

Figure 4.10: Rate-distortion curves for Bayer-pattern video compression using H.264/AVC. Herein, bilinear interpolation is used for demosaicing.



(a) Panning (b) Zooming (c) Moving Object

Figure 4.11: Rate-distortion curves for Bayer-pattern video compression using H.264/AVC. Herein, Laplace interpolation is used for demosaicing.

Apart from bilinear interpolation, Laplace interpolation is also used to carry out experiments. Laplace interpolation is used for the demosaicing of the original RGB images as well as the reconstructed RGB images. Moreover, it is also applied to the interpolation of the Y component for the proposed method B-4:2:2. Other parts of the simulation remain the same. Rate-distortion curves are shown in Figure 4.11.

Experimental results in Figure 4.10 show that the proposed schemes B-4:2:0 and B-4:2:2 both outperform the conventional method over the entire bit rate range in the case of bilinear interpolation. At high bit rates, there is a significant gain of more than 1.5 dB. At low bit rates, the improvement is less, but still more than 0.5 dB. When the two proposed methods are compared, we find that the method B-4:2:0 outperforms the method B-4:2:2 in most cases and the gain in rate-distortion performance is 0.2 dB on average.

As for the results for Laplace interpolation, as shown in Figure 4.11, the proposed scheme B-4:2:0 is still significantly more efficient than the conventional scheme C-4:2:0, although the gain is not as large as in the case of bilinear interpolation. In comparison, the proposed method B-4:2:2 is less efficient than B-4:2:0. The reason lies in the fact that Laplace interpolation keeps more information in the high frequency bands, typically edge information, and thus reducing the Y samples by a factor of two in the proposed scheme B-4:2:2 results in a relatively large loss of information. Nevertheless, the proposed B-4:2:2 method is favorable, because its performance is still better than or comparable to that of the conventional scheme, while the complexity is reduced by almost 50%.

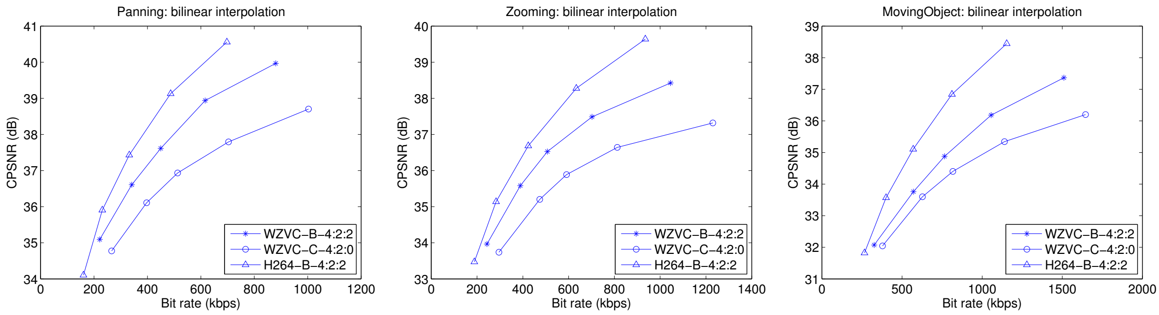
4.5.2 Bayer-pattern video compression using WZVC

In the simulation for Wyner-Ziv video compression, the GOP structure of the Wyner-Ziv video codec is set to I-WZ-I-WZ...I, which means that every second frame is an I-frame and the frame between two consecutive I-frames is a Wyner-Ziv frame. For simplicity, it is assumed that the side information at the decoder can be quite accurately generated. Specifically, a bidirectional MCP scheme, similar to the one in the JM coder, is used to generate side information for Wyner-Ziv frames at the Wyner-Ziv video encoder and this side information is used to perform Wyner-Ziv decoding. In practice a MCI scheme needs to be implemented at the decoder to generate side information and its efficiency is lower than that of the MCP at a conventional video encoder.

The results of Wyner-Ziv Bayer-pattern video compression are compared with those of Bayer-pattern video compression using H.264/AVC. The JM 17.2 reference software is used to simulate H.264/AVC video coding. The GOP structure is set to I-B-I-B...I, which means that every second frame is an I-frame and there is a B-frame between every two consecutive I-frames. The YUV format is set to be 4:2:0 for the conventional scheme and 4:2:2 for the proposed scheme B-4:2:2. Default values are taken for other coding parameters.

Rate-distortion curves for different methods using bilinear interpolation and for different test sequences are presented in Figure 4.12. At high bit rates, the scheme WZVC-B-4:2:2 outperforms the comparison scheme WZVC-C-4:2:0 by up to 1.5 dB in PSNR. This result means that the proposed CST method B-4:2:2 can contribute to a higher video quality when it comes to Wyner-Ziv Bayer-pattern video compression. Nevertheless, WZVC still has a relatively large performance gap from H.264/AVC.

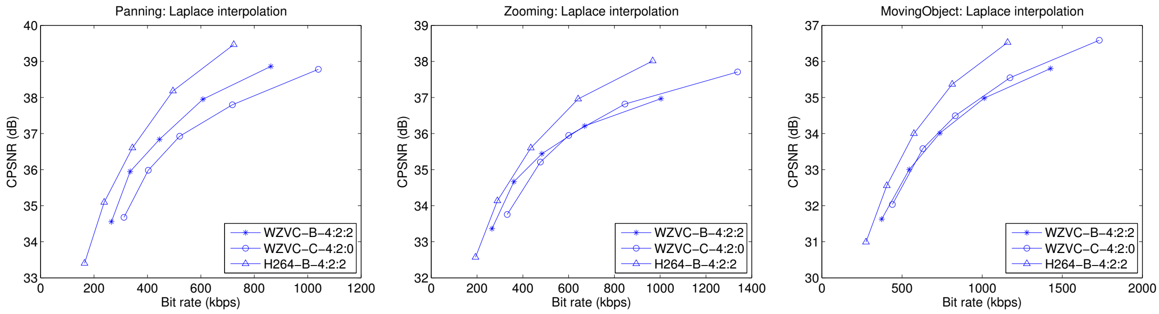
The results for Laplace interpolation are shown in Figure 4.12. For the Panning se-



(a) Panning (b) Zooming (c) Moving Object

Figure 4.12: Rate-distortion performance evaluation for Wyner-Ziv Bayer-pattern compression. In this instance, bilinear interpolation is used for demosaicing.

quence, WZVC-B-4:2:2 still outperforms WZVC-C-4:2:0. For the other two sequences, however, WZVC-B-4:2:2 and WZVC-C-4:2:0 have a similar performance. Nevertheless, the former one is supposed to be preferable, because its coding complexity is almost halved compared to that of the latter one. Compared to H.264/AVC, however, WZVC remains to be improved for a higher coding efficiency.



(a) Panning (b) Zooming (c) Moving Object

Figure 4.13: Rate-distortion performance evaluation for Wyner-Ziv Bayer-pattern compression. In this instance, Laplace interpolation is used for demosaicing.

4.5.3 Encoding time reduction

Besides the rate-distortion curves, the average encoding time of different schemes and different test sequences are listed in Table 4.1. The major specifications of the computer used for this simulation is listed below:

- Processor: Intel(R) Core(TM)2 Duo CPU P8600 @ 2.4 GHz 2.4 GHz;
- RAM: 4.00 GB;
- Operational system: 64-bit Windows 7 Professional.

The reference software JM 17.2 with the configuration file "encoder_baseline.cfg" is used for H.264/AVC video encoding and the DISCOVER codec [AAD+07] is used for Wyner-Ziv video encoding. The same JM coder with the same configuration file is used in the DISCOVER codec for encoding the I-frames. The GOP structure for H.264/AVC is I-B-I-B...I, which means that every second frame is an I-frame and there is a B-frame between two every two consecutive I-frames. The GOP structure for WZVC is I-WZ-I-WZ...I, which means that every second frame is an I-frame and there is a Wyner-Ziv frame between any two consecutive I-frames. Comparatively speaking, the time used for the preprocessing of Bayer-pattern video, including demosaicing and color space transform, is negligible. Therefore, only the time for video encoding is taken into account.

One thing worth mentioning is that the original Wyner-Ziv video codec used in this thesis has been developed using the programming language Matlab, whereas the JM reference software using the programming language C. Generally speaking, the execution time of an implementation using Matlab is significantly longer than that of the same implementation using C. Therefore, it is not fair to compare the encoding time of the original Wyner-Ziv video codec with that of the JM coder. Instead, the state-of-the-art DISCOVER codec implemented using the C language, with a similar functionality to that of the original Wyner-Ziv video codec in this thesis, is used to obtain the encoding time of the WZVC-C-4:2:0 and the WZVC-B-4:2:2 schemes. The DISCOVER codec is used only for this purpose. All the other results in the thesis have been obtained using the original Wyner-Ziv video codec developed in Matlab.

Table 4.1: Video encoding time of different schemes and sequences. In this simulation, the quantization parameter (QP) for the JM coder is 25 and the quantization index (Qindex) for the DISCOVER codec is 8. The same JM coder with the same configuration and QP is used in the DISCOVER codec for encoding the I-frames.

Method	Encoding time (ms/frame)		
	Panning	Zooming	Moving Object
H264-C-4:2:0	7575	7382	6841
H264-B-4:2:0	7353	7519	6991
H264-B-4:2:2	3544	3512	3501
WZVC-C-4:2:0	390	387	426
WZVC-B-4:2:2	200	202	223

In this simulatoin, the quantization parameter (QP) for the JM coder is 25 and the quantization index (Qindex) for the DISCOVER codec is 8 and this Qindex represents the finest quantizer in the DISCOVER codec. The same JM coder with the same configuration and QP is used in the DISCOVER codec for encoding the I-frames.

As shown in the table, the encoding time of the conventional method H264-C-4:2:0 and the proposed method H264-B-4:2:0 are very close to one another, while the method H264-B-4:2:2 takes only half of the time to encode a frame. Such a large reduction in computational complexity makes the method B-4:2:2 quite an attractive solution for Bayer-pattern video compression using H.264/AVC, provided that a small loss in coding efficiency can be tolerated.

When WZVC-B-4:2:2 is compared with WZVC-B-4:2:0, the encoding time is also reduced approximately by half. Moreover, the encoding time of the scheme WZVC-B-4:2:2 is approximately seventeen times lower than that of the scheme H264-B-4:2:2. This significant reduction of the encoding complexity is due to the fact that the motion estimation for every Wyner-Ziv frame in WZVC-B-4:2:2 is moved from the encoder to the decoder, whereas a complex bidirectional motion estimation needs to be carried out for every B-frame in H264-B-4:2:2.

4.6 Chapter summary

This chapter refers to Wyner-Ziv Bayer-pattern video compression. The motivation of the work presented in this chapter lies in a Bayer-pattern video compression system of low encoding complexity. This system is developed in three stages.

Firstly, a novel CST scheme B-4:2:0 is proposed. The positions for chroma samples in the process of chroma subsampling are properly chosen. The U samples at the position of B pixels and the V samples at the position of R pixels are taken. And these novel sample positions lead to significant improvements of video quality compared to the nominal sample positions standardized in H.264/AVC.

Secondly, the CST scheme B-4:2:2 is proposed. Compared to B-4:2:2, only half of the Y samples, those at the position of G pixels, are taken, and the other half are discarded. This scheme reduces the computational complexity by almost 50%.

Finally, a Wyner-Ziv Bayer-pattern video compression system is developed by incorporating the CST scheme B-4:2:2 and WZVC. The B-4:2:2 scheme reduces the complexity almost by half compared to the C-4:2:0, while WZVC can further reduce the encoding complexity by shifting part or all of the computation for motion estimation to the decoder. Above all, the proposed Wyner-Ziv Bayer-pattern video compression system has a minimal encoding complexity.

Chapter 5

Conclusions and future work

In this thesis, Wyner-Ziv Bayer-pattern video coding has been studied. The research work involved preprocessing Bayer-pattern video, optimizing the Wyner-Ziv video codec and transmitting video over a wireless error-prone channel. And this work ended up with a Wyner-Ziv Bayer-pattern video communications system with low encoding complexity and high error resiliency.

5.1 Conclusions

First and foremost, a novel color space transform method B-4:2:0 has been proposed for Bayer-pattern video. This method has been presented in Chapter 4. Sampling positions of U and V in chroma subsampling are more reasonably chosen than nominal sampling positions standardized in H.264/AVC. Therefore, the proposed method B-4:2:0 improves the reconstructed video quality by up to 1.5 dB in PSNR. A second proposed method B-4:2:2 inherits the benefits of the novel chroma sampling positions in the method B-4:2:0 and reduces the luma samples by half. The method B-4:2:2 leads to a complexity reduction almost by 50% compared to the method B-4:2:0 and the resulting video quality is still higher than or comparable to that of the conventional method.

Moreover, the following two conclusions on the coding efficiency of WZVC can be drawn if the discussion is confined to source coding and the transmission channel is assumed to be error-free.

- The coding efficiency of WZVC using turbo codes can be improved considerably by adjusting the log-MAP BCJR algorithm used for turbo decoding. Because the transmission is assumed to be error-free, the received parity bits are definitely correct. Theoretically speaking, the log-likelihood-ratio (LLR) of an error-free parity bit is supposed to be infinite in the log-MAP BCJR algorithm. Infinite values, however, are invalid in practical implementations. It has been shown in Section 2.2 that the BCJR decoding using a sufficiently large LLR value for error-free parity bits can approach its optimal performance and a lower bound has been derived for this LLR value.

- A method for distributing motion estimation between encoder and decoder flexibly has been proposed in Section 2.4. This method makes the encoding complexity scalable and enables a trade-off between coding efficiency and encoding complexity. In conclusion, performing motion estimation at the Wyner-Ziv video encoder for part of the macroblocks, typically 50% of them, can improve the Wyner-Ziv video coding efficiency significantly.

Furthermore, error-resilient video transmission using WZVC via an AWGN channel has been studied based on the assumption that no retransmission is possible. This study has resulted in the two conclusions below.

- WZVC proves significantly more error-resilient than WZVC+FEC in the context of joint source-channel coding and the reconstructed video quality can be up to 1.2 dB higher. In WZVC, a single turbo coder is used for the dual purpose of source and channel coding, and thus only the total rate needs to be estimated correctly. In WZVC+FEC, however, source coding and channel coding are carried out using two turbo coders and at two separate stages, and therefore the source coding rate and the channel coding rate both need to be estimated accurately. In comparison, joint source-channel rate control of WZVC+FEC is more difficult. Therefore, WZVC+FEC results in a higher decoder failure probability and larger mean end-to-end distortion.

- Symbol based WZVC using trellis merging in turbo decoding outperforms bit-plane based WZVC using binary turbo codes. The two schemes have a comparable performance at a good channel state, whereas the former one outperforms the latter one when the channel becomes worse. This conclusion is made only for the turbo codes used in this thesis work and might need to be modified if different codes, for example LDPC codes, are used.

To sum up, the proposed Wyner-Ziv Bayer-pattern video coding system ends up with two attractive features. On one hand, a very low encoding complexity can be achieved, if motion estimation is completely moved from the encoder to the decoder, though this prototype results in a considerable loss in coding efficiency compared to conventional video coding. On the other hand, if the Wyner-Ziv video encoder is capable of motion estimation and sophisticated joint source-channel rate control, WZVC can lead to a high error-resiliency, and therefore proves to be a promising candidate for joint source-channel coding of video.

Despite the contributions in Wyner-Ziv Bayer-pattern video coding, this thesis also has some limitations. Firstly, the discussion about the preprocessing, including the demosaicing and the color space transform, has been confined to the Bayer pattern. Nevertheless, a similar methodology can be used, when it comes to an alternative color filter array. Secondly, the discussion on Wyner-Ziv video coding in this thesis, particularly for the part of joint source-channel coding, has been confined to pixel-domain Wyner-Ziv video coding. When it comes to transform-domain Wyner-Ziv video coding, it remains a question how to achieve the highest possible error-resiliency.

5.2 Future work

This thesis work can be extended in three directions. Firstly, apart from the most popular Bayer pattern, there are quite a few alternative color filter arrays, for instance the RGBE filter used in a few types of Sony cameras. It will be interesting to look into different color filter arrays and try to optimize the video coding system for them.

Secondly, the efficiency of motion estimation at the decoder remains to be improved. Different from the popular hash-based methods, the method proposed in this thesis is to perform motion estimation for part of the MBs and transmit their motion vectors to help motion estimation at the decoder. Generally speaking, motion vectors and hash information can both be regarded as helper information for motion estimation at the decoder and thus can work together in principle. And it will be worth investigating how far the combination of the two can improve the accuracy of motion estimation at the decoder.

Thirdly, joint source-channel coding using WZVC, characterized by its high error resiliency, is a promising application of WZVC. The discussion in this thesis has been confined to pixel-domain Wyner-Ziv video coding. It is worth extending the discussion to transform-domain Wyner-Ziv coding and investigating how to achieve the highest possible error-resiliency. Moreover, only forward error correction (FEC) is considered in the thesis for joint source-channel coding. A potential topic is whether and how WZVC can benefit from retransmission for a higher error resiliency.

Bibliography

Publications by the author

[CS07a] H. Chen and E. Steinbach. Fast motion-estimation-based reference frame generation in Wyner-Ziv residual video coding. In *Proc. IEEE International Conference on Multimedia & Expo*, Beijing, China, July 2007. [cited at p. 7, 31]

[CS07b] H. Chen and E. Steinbach. Wyner-Ziv video coding based on turbo codes exploiting perfect knowledge of parity bits. In *Proc. IEEE International Conference on Multimedia & Expo*, Beijing, China, July 2007. [cited at p. 7]

[CS08] H. Chen and E. Steinbach. Flexible distribution of computational complexity between the encoder and the decoder in distributed video coding. In *Proc. IEEE International Conference on Multimedia & Expo*, Hannover, Germany, June 2008. [cited at p. 7]

[CS09] H. Chen and E. Steinbach. Low-complexity bayer-pattern video compression using distributed video coding. In *Proc. SPIE Visual Communications and Image Processing*, San Jose, CA, January 2009. [cited at p. 7]

[CSC11a] H. Chen, E. Steinbach, and C. W. Chen. A comparison of the error resiliency of bit-plane based and symbol based pixel-domain distributed video coding. In *Proc. IEEE International Conference on Image Processing*, Brussels, Belgium, September 2011. [cited at p. 7]

[CSC11b] H. Chen, E. Steinbach, and C. W. Chen. Joint source-channel rate control for pixel-domain distributed video coding. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1533–1536, May 2011. [cited at p. 7, 45]

[CSS09] H. Chen, M. Sun, and E. Steinbach. Compression of Bayer-pattern video sequences using adjusted chroma subsampling. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(12):1891–1896, December 2009. [cited at p. 7]

[RCS10] A. Rehman, H. Chen, and E. Steinbach. Addressing the uncertainty in critical rate estimation for pixel-domain Wyner-Ziv video coding. In *Proc. SPIE Visual Communications and Image Processing*, Huang Shan, An Hui, China, July 2010. [cited at p. 7, 10, 28, 45]

General publications

[AAD$^+$07] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, and M. Ouaret. The discover codec: Architecture, techniques and evaluation. In *Proc. Picture Coding Symposium*, Lisboa, Portugal, November 2007. [cited at p. 9, 22, 30, 40, 41, 73]

[ABP05] J. Ascenso, C. Brites, and F. Pereira. Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding. In *Proc. 5th EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services*, Smolenice, Slovak Republic, June 2005. [cited at p. 10, 33]

[Ada95] J. E. Adams. Intersections between color plane interpolation and other image processing functions in electronic photography. *Proc. SPIE*, 2416:144–151, March 1995. [cited at p. 61]

[AFWA06] A.B.B. Adikari, W.A.C. Fernando, W.A.R.J. Weerakkody, and H.K. Arachchi. A sequential motion compensation refinement technique for distributed video coding of Wyner-Ziv frames. In *Proc. IEEE International Conference on Image Processing*, Atlanta, USA, October 2006. [cited at p. 10]

[AG02] A. Aaron and B. Girod. Compression with side information using turbo codes. In *Proc. IEEE Data Compression Conference*, pages 252–261, Snowbird, Utah, April 2002. [cited at p. 15]

[AH97] J. E. Adams and J. F. Hamilton. Design of practical color filter array interpolation algorithms for digital cameras. *Proc. SPIE*, 3028:117–125, 1997. [cited at p. 61]

[ASW08] A. Avudainayagam, J. M. Shea, and D. Wu. Hyper-trellis decoding of pixel-domain Wyner-Ziv video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(5):557–568, May 2008. [cited at p. 12, 44, 45]

[AT05a] X. Artigas and L. Torres. Improved signal reconstruction and return channel suppression in distributed video coding systems. In *Proc. International Symposium ELMAR*, pages 53–56, June 2005. [cited at p. 10]

[AT05b] X. Artigas and L. Torres. Iterative generation of motion-compensated side information for distributed video coding. In *Proc. IEEE International Conference on Image Processing*, volume 1, pages 833–836, Genova, Italy, September 2005. [cited at p. 10]

[BAP06] C. Brites, J. Ascenso, and F. Pereira. Studying temporal correlation noise modeling for pixel based Wyner-Ziv video coding. In *Proc. IEEE International Conference on Image Processing*, Atlanta, USA, October 2006. [cited at p. 10, 13]

[Bay76] B. E. Bayer. Color imaging array. U.S. Patent 3,971,065, 1976. [cited at p. 1, 59]

[BBF02] B. A. Banister, B. Belzer, and T. R. Fischer. Robust image transmission using jpeg2000 and turbo-codes. *IEEE Signal Processing Letters*, 9(4):117–119, April 2002. [cited at p. 40]

[Ber98] J. Berkmann. On turbo decoding of nonbinary codes. *IEEE communication letters*, 2(4):94–96, April 1998. [cited at p. 12, 44, 45]

[BM01] J. Bajcsy and P. Mitran. Coding for the Slepian-Wolf problem with turbo codes. In *Proc. IEEE Global Telecommunications Conference*, volume 2, pages 1400–1404, November 2001. [cited at p. 4, 15]

[BP07] C. Brites and F. Pereira. Encoder rate control for transform domain Wyner-Ziv video coding. In *Proc. IEEE International Conference on Image Processing*, volume 2, pages 5–8, San Antonio, TX, USA, September 2007. [cited at p. 10]

[BP10] C. Brites and F. Pereira. Probability updating for decoder and encoder rate control turbo based Wyner-Ziv video coding. In *Proc. IEEE International Conference on Image Processing*, pages 3737–3740, September 2010. [cited at p. 10, 13]

[CCP99] E. Chang, S. Cheung, and D. Y. Pan. Color filter array recovery using a threshold-based variable number of gradients. *Proc. SPIE*, 3650:36–43, March 1999. [cited at p. 61]

[CE87] D. R. Cok and Eastman Kodak Company. Signal processing method and apparatus for producing interpolated chrominance values in a sampled color image signal. U.S. Patent 4 642 678, 1987. [cited at p. 61]

[CHKK07] B.-D. Choi, J.-W. Han, C.-S. Kim, and S.-J. Ko. Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(4):407–416, April 2007. [cited at p. 33]

[CMCS07] T. Clerckx, A. Munteanu, J. Cornells, and P. Schelkens. Distributed video coding with shared encoder/decoder complexity. In *Proc. IEEE International Conference on Image Processing*, San Antonio, Texas, USA, September 2007. [cited at p. 31]

[Cok87] D. R. Cok. Signal processing method and apparatus for sampled image signals. U.S. Patent 4 630 307, 1987. [cited at p. 61]

[CT02] L. Chang and Y. P. Tan. Color filter array demosaicking: new method and performance measures. *IEEE Transactions on Image Processing*, 12(10):1194–1210, October 2002. [cited at p. 61]

[CWO08] N.-M. Cheung, H. Wang, and A. Ortega. Sampling-based correlation estimation for distributed source coding under rate and complexity constraints. *IEEE Transactions on Image Processing*, 17(11):2122–2137, November 2008. [cited at p. 10]

[DN06] C. Doutre and P. Nasiopoulos. An efficient compression scheme for colour filter array video sequences. In *Proc. IEEE 8th Workshop on Multimedia Signal Processing*, pages 166–169, October 2006. [cited at p. 62, 67]

[DNP08] C. Doutre, P. Nasiopoulos, and K. N. Plataniotis. H.264-based compression of bayer pattern video sequences. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(6):725–734, June 2008. [cited at p. 62, 67]

[GARRM05] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero. Distributed video coding. *Proceedings of the IEEE, Special Issue on Video Coding and Delivery*, 93(1):71–83, January 2005. [cited at p. 9, 10, 22, 30, 42]

[GE98] A. J. Goldsmith and M. Effros. Joint design of fixed-rate source codes and multiresolution channel codes. *IEEE Transactions on Communications*, 46(10):1301–1312, October 1998. [cited at p. 40]

[GGA$^+$05b] B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. W. Schafer, and R. M. Mersereau. Demosaicking: color filter array interpolation. *IEEE Signal Processing Magazine*, 22(1):44–54, January 2005. [cited at p. 60]

[GKC$^+$05] F. Gastaldi, C. C. Koh, M. Carli, A. Neri, and S. K. Mitra. Compression of videos captured via bayer patterned color filter arrays. In *Proc. 13th European Signal Processing Conference*, Antalya, Turkey, 2005. [cited at p. 62]

[h2693] Video codec for audiovisual services at p x 64 kbit/s. ITU-T Recommendation H.261, 1993. [cited at p. 2]

[h2605] Video coding for low bit rate communication. ITU-T Recommendation H.263, 2005. [cited at p. 2]

[h2610] Advanced video coding for generic audiovisual services. ITU-T Recommendation H.264 and ISO/IEC 14496-10 AVC, 2010. [cited at p. 2, 9]

[HA97] J. Hamilton and J. Adams. Adaptive color plane interpolation in single sensor color electronic camera. U.S. Patent 5 629 734, 1997. [cited at p. 61]

[HC05] G. Hua and C. W. Chen. Distributed source coding in wireless sensor networks. In *Proc. International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks*, August 2005. [cited at p. 15]

[HC08] G. Hua and C. W. Chen. Distributed video coding with zero motion skip and efficient DCT coefficient encoding. In *Proc. IEEE International Conference on Multimedia & Expo*, Hannover, Germany, June 2008. [cited at p. 11]

[HCC02] Z. He, J. Cai, and C. W. Chen. Joint source channel rate-distortion analysis for adaptive mode selection and rate control in wireless video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(6):511–523, June 2002. [cited at p. 40]

[HCC07] G. Hua, L. Cao, and C. W. Chen. Distributed source coding under noisy transmission environments. In *Proc. IEEE International Conference on Multimedia & Expo*, Bejing, China, July 2007. [cited at p. 41]

[HM08] R. Hänsel and E. Müller. Simplifying the rate control scheme for distributed video coding by flexible Slepian-Wolf decoding. In *Proc. 3rd Pacific Rim Symposium on Advances in Image and Video Technology*, PSIVT '09, pages 1022–1033, Berlin, Heidelberg, 2008. Springer-Verlag. [cited at p. 17]

[HOK02] Y. Hel-Or and D. Keren. Demosaicing of color images using steerable wavelets. Technical Report HPL-2002-206R1 20 020 830, HP Labs, Israel, 2002. [cited at p. 61]

[HP03] K. Hirakawa and T. Parks. Adaptive homogeneity-directed demosaicing algorithm. In *Proc. IEEE International Conference on Image Processing*, volume 3, pages 669–672, September 2003. [cited at p. 61]

[Jac07] K. Jack. *Video Demystified*. 978-0-7506-8395-1. Elsevier, 5 edition, April 2007. [cited at p. 62]

[JJ81a] J. R. Jain and A. K. Jain. Displacement measurement and its application in interframe image coding. *IEEE Transactions on Communications*, 29(12):1799–1808, December 1981. [cited at p. 32]

[KB02] R. Kakarala and Z. Baharav. Adaptive demosaicing with the principal vector method. *IEEE Transactions on Consumer Electronics*, 48(4):932–937, November 2002. [cited at p. 61]

[Kim99b] R. Kimmel. Demosaicing: Image reconstruction from color ccd samples. *IEEE Transactions on Image Processing*, 8(9):1221–1228, September 1999. [cited at p. 61]

[KMM03] C. C. Koh, J. Mukherjee, and S. K. Mitra. New efficient methods of image compression in digital cameras with color filter array. *IEEE Transactions on Consumer Electronics*, 49(4):1448–1456, 2003. [cited at p. 61, 67]

[KO99] D. Keren and M. Osadchy. Restoring subsampled color images. *Machine Vision and Applications*, 11(4):197–202, December 1999. [cited at p. 61]

[KOY03] N. Kehtarnavaz, H.-J. Oh, and Y. Yoo. Color filter array interpolation using color correlation and directional derivatives. *Journal of Electronic Imaging*, 12:621–632, October 2003. [cited at p. 61]

[KR03] O. Kalevo and H. Rantanen. Sharpening methods for images captured through bayer matrix. In *Proc. IS&T-SPIE, Sensors, Cameras and Applications for Digital Photography V*, volume 5017, pages 286–297, January 2003. [cited at p. 61]

[LC01] H. Li and C. W. Chen. Robust image transmission with bidirectional synchronization and hierarchical error correction. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(11):1183–1187, November 2001. [cited at p. 40]

[LC04] S. Lin and D. J. Costello. *Error Control Coding*. 0-13-042672-5 , 0-13-017973-6. Pearson-Prentice Hall, Upper Saddle River, N.J., 2 edition, 2004. [cited at p. 15, 16, 18]

[LHJ$^+$08] L. Liu, D.-K. He, A. Jagmohan, L. Lu, and E. J. Delp. A low-complexity iterative mode selection algorithm for Wyner-Ziv video compression. In *Proc. IEEE International Conference on Image Processing*, San Diego, CA, USA, October 2008. [cited at p. 11]

[LMP03] R. Lukac, K. Martin, and K. N. Plataniotis. Demosaicked image post-processing using local color ratios. *IEEE Transactions on Image Processing*, 12(10):1194–1210, October 2003. [cited at p. 61]

[LP94] C. Laroche and M. Prescott. Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients. U.S. Patent 5 373 322, 1994. [cited at p. 61]

[LP04] R. Lukac and K. N. Plataniotis. Normalized color-ratio modeling for cfa interpolation. *IEEE Transactions on Consumer Electronics*, 50(2):737–745, May 2004. [cited at p. 61]

[LPHA04] R. Lukac, K. N. Plataniotis, D. Hatzinakos, and M. Aleksic. A novel cost effective demosaicing approach. *IEEE Transactions on Consumer Electronics*, 50(1):256–261, February 2004. [cited at p. 61]

[LYDB02] P. Longère, X. Yhang, P. B. Delahunt, and D. H. Brainard. Perceptual assessment of demosaicing algorithm performance. *Proceedings of the IEEE*, 90(1):123–132, January 2002. [cited at p. 61]

[MFEK$^+$08] J. L. Martinez, G. Fernandez-Escribano, H. Kalva, W. A. R. J. Weerakkody, W. A. C. Fernando, and A. Garrido. Feedback free DVC architecture using machine learning. In *Proc. IEEE International Conference on Image Processing*, pages 1140–1143, San Diego, CA, USA, October 2008. [cited at p. 10]

[MLM02a] J. Mukherjee, M. K. Lang, and S. K. Mitra. Color demosaicing in yuv color space. In *Proc. IASTED Conference*, pages 96–101, Malaga, Spain, September 2002. [cited at p. 69]

[MRPN$^+$08] M. Morbée, A. Roca, J. Prades-Nebot, A. Piurica, and W. Philips. Reduced decoder complexity and latency in pixel-domain Wyner-Ziv video coders. *Signal, Image and Video Processing*, 2(2):129–140, June 2008. [cited at p. 10]

[PLC$^+$09] S.-U. Park, Y.-Y. Lee, J.-W. Choi, J.-W. Kang, C.-S. Kim, and S.-U. Lee. Multiple channel division for efficient distributed video coding. In *Proc. IEEE International Conference on Image Processing*, pages 1401–1404, November 2009. [cited at p. 10, 11, 13]

[PMR07] R. Puri, A. Majumdar, and K. Ramchandran. Prism: A video coding paradigm with motion estimation at the decoder. *IEEE Transaction on Image Processing*, 16(10):2436–2448, October 2007. [cited at p. 4, 9, 22, 30, 40, 41]

[PT00b] S.C. Pei and I. K. Tam. Effective color interpolation in ccd color filter array using signal correlation. In *Proc. IEEE International Conference Image Processing*, volume 3, pages 10–13, September 2000. [cited at p. 61]

[PTG$^+$08] F. Pereira, L. Torres, C. Guillemot, T. Ebrahimi, R. Leonardi, and S. Klomp. Distributed video coding: Selecting the most promising application scenarios. *Signal Processing: Image Communications*, 23:339–352, April 2008. [cited at p. 4]

[PZS10] Y. Peng, F. Zhang, and E. Steinbach. Error-resilient video transmission for short-range point-to-point wireless communication. In *The 2nd IEEE International Workshop on Multimedia Computing and Communications*, Zurich, Switzerland, August 2010. [cited at p. 40]

[RBG08] S. Rane, P. Baccichet, and B. Girod. Systematic lossy error protection of video signals. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(10):1347–1360, October 2008. [cited at p. 40, 41]

[RM00] D. Rowitch and L. Milstein. On the performance of hybrid FEC/ARQ systems using rate compatible punctured turbo codes. *IEEE Transactions on Communications*, 48(6):948–959, June 2000. [cited at p. 12]

[RS03] R. Ramanath and W. E. Snyder. Adaptive demosaicking. *Journal of Electronic Imaging*, 12(4):633–642, 2003. [cited at p. 61]

[SHG$^+$08] T. Sheng, G. Hua, H. Guo, J. Zhou, and C. W. Chen. Rate allocation for transform domain Wyner-Ziv video coding without feedback. In *Proc. ACM International Conference on Multimedia*, pages 701–704, New York, NY, USA, 2008. [cited at p. 10]

[SK10] J. Seiler and A. Kaup. Complex-valued frequency selective extrapolation for fast image and video signal extrapolation. *IEEE Signal Processing Letters*, 17(11):949–952, November 2010. [cited at p. 10]

[SK11a] J. Seiler and A. Kaup. A fast algorithm for selective signal extrapolation with arbitrary basis functions. *EURASIP Journal on Advances in Signal Processing*, 2011, January 2011. [cited at p. 10]

[SK11b] J. Seiler and A. Kaup. Motion compensated three-dimensional frequency selective extrapolation for improved error concealment in video communication. *Journal of Visual Communication and Image Representation*, 22:213–225, March 2011. [cited at p. 10]

[SKR85] N. Z. Shor, K. C. Kiwiel, and A. Ruszcayǹski. *Minimization methods for non-differentiable functions.* Springer-Verlag New York, Inc., New York, NY, USA, 1985. [cited at p. 29]

[SW73] D. Slepian and J.K. Wolf. Noiseless coding of correlated information sources. *IEEE Transactions on Information Theory*, 19(4):471–480, July 1973. [cited at p. 3, 9]

[SZ97] P. G. Sherwood and K. Zeger. Progressive image coding for noisy channels. *IEEE Signal Processing Letters*, 4(7):189–191, July 1997. [cited at p. 40]

[TA97] C.-W. Tang and O. C. Au. Unidirectional motion compensated temporal interpolation. In *Proc. IEEE International Symposium on Circuits and Systems*, September 1997. [cited at p. 10]

[TKA$^+$90] T. Tanaka, S. Katoh, I. Akiyama, N. Teranishi, K. Orihara, and E. Oda. Hdtv single-chip ccd color camera. *IEEE Transactions onConsumer Electronics*, 36(3):479–485, August 1990. [cited at p. 1]

[VAG05] D. Varodayan, A. Aaron, and B. Girod. Rate-adaptive distributed source coding using low-density parity-check codes. In *Proc. Asilomar Conference on Signals and Systems*, Pacific Grove, CA, November 2005. [cited at p. 15]

[WBGL06] 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 *Proc. IEEE International Conference on Image Processing*, pages 277–280, October 2006. [cited at p. 10, 13, 17]

[WBGL07] R.P. Westerlaken, S. Borchert, R.K. Gunnewiek, and R.L. Lagendijk. Analyzing symbol and bit plane-based LDPC in distributed video coding. In *Proc. IEEE International Conference on Image Processing*, volume 2, pages 17–20, September 2007. [cited at p. 22, 27, 29]

[WCB97] X. Wu, W. K. Choi, and P. Bao. Color restoration from digital camera data by pattern matching. *Proc. SPIE*, 3018:12–17, April 1997. [cited at p. 61]

[WGL06] R. P. Westerlaken, R. K. Gunnewiek, and R. L. Lagendijk. Turbo-code based Wyner-Ziv video compression. In *Proc. 26th Symposium on Information Theory in the Benelux*, pages 113–120, May 2006. [cited at p. 10, 13]

[WZ76] A. D. Wyner and J. Ziv. The rate-distortion function for source coding with side information at the decoder. *IEEE Transactions on Information Theory*, 22(1):1–10, January 1976. [cited at p. 3, 9]

[WZ04] X. Wu and N. Zhang. Primary-consistent soft-decision color demosaicking for digital cameras. *IEEE Transactions on Image Processing*, 13(9):1263–1274, September 2004. [cited at p. 61]

[WZ05] X. Wu and N. Zhang. Joint temporal and spatial color demosaicking. *Proc. SPIE*, 5017(2):307–313, May 2005. [cited at p. 61]

[XX06] Q. Xu and Z. Xiong. Layered Wyner-Ziv video coding. *IEEE Transactions on Image Processing*, 15(12):3791–3803, December 2006. [cited at p. 9, 40, 41]

[YFPP08] C. Yaacoub, J. Farah, and B. Pesquet-Popescu. Feedback channel suppression in distributed video coding with adaptive rate allocation and quantization for multiuser applications. *EURASIP Journal on Wireless Communications and Networking*, 2008, December 2008. [cited at p. 17]

[ZL08] M. Zamani and F. Lahouti. Distributed source coding using symbol-based and non-binary turbo codes - applications to wireless sensor networks. *Communications, IET*, 2(8):1089–1097, September 2008. [cited at p. 22]

[ZW05] L. Zhang and X. Wu. Color demosaicking via directional linear minimum mean square-error estimation. *IEEE Transactions on Image Processing*, 14(12):2167–2178, December 2005. [cited at p. 61]

[ZXSY09] Y. Zhang, H. Xiong, L. Song, and S. Yu. Spatial non-stationary correlation noise modeling for Wyner-Ziv error resilience video coding. In *Proc. 16th IEEE International Conference on Image Processing*, pages 2929–2932, November 2009. [cited at p. 10, 13]