Distributed Video Coding: Iterative Improvements

Luong, Huynh Van; Forchhammer, Søren; Larsen, Knud J.

Publication date:
2013

Document Version
Publisher's PDF, also known as Version of record

Distributed Video Coding: Iterative Improvements

Huynh Van Luong

Kongens Lyngby, 3-June 2013
DTU Fotonik, Technical University of Denmark

Technical University of Denmark
Department of Photonics Engineering
Building 343, DK-2800 Kongens Lyngby, Denmark
Phone +45 4525 6352, Fax +45 4593 6581
info@fotonik.dtu.dk
www.fotonik.dtu.dk DTU Fotonik-PhD-2013

Summary

Nowadays, emerging applications such as wireless visual sensor networks and wireless video surveillance are requiring lightweight video encoding with high coding efficiency and error-resilience. Distributed Video Coding (DVC) is a new coding paradigm which exploits the source statistics at the decoder side offering such benefits for these applications. Although there have been some advanced improvement techniques, improving the DVC coding efficiency is still challenging.

The thesis addresses this challenge by proposing several iterative algorithms at different working levels, e.g. bitplane, band, and frame levels. In order to show the information theoretic basis, theoretical foundations of DVC are introduced. The first proposed algorithm applies parallel iterative decoding using multiple LDPC decoders to utilize cross bitplane correlation. To improve Side Information (SI) generation and noise modeling and also learn from the previous decoded Wyner-Ziv (WZ) frames, side information and noise learning (SING) is proposed. The SING scheme introduces an optical flow technique to compensate the weaknesses of the block based SI generation and also utilizes clustering of DCT blocks to capture cross band correlation and increase local adaptivity in noise modeling. During decoding, the updated information is used to iteratively reestimate the motion and reconstruction in the proposed motion and reconstruction reestimation (MORE) scheme. The MORE scheme not only reestimates the motion vectors for improving SI and noise modeling but also compensates the residual motion based on the previously decoded WZ frames. Furthermore, the MORE codec enhances the reconstruction by proposing a generalized reconstruction algorithm to optimize reconstructing with multiple competitive SIs. Finally, an adaptive mode decision is investigated to take advantage of skip and intra mode in DVC by deciding the coding modes based on the quality of key frames and rate of WZ frames. Overall, the proposed algorithms significantly improve the coding efficiency of DVC contributing valuable solutions for the emerging applications.

Resumé

I dag kræver nye applikationer, såsom trådløse visuelle sensornetværk og trådløs videoovervågning, simpel videoindkodning med høj kodningseffektivitet og robusthed overfor fejl. Distributed Video Coding (DVC) er et nyt kodningsparadigme, som udnytter kildestatistikken ved dekoderen og kan dermed være en fordel ved førnævnte applikationer. Selvom der er blevet udviklet avancerede teknikker til forbedring af DVC, er forbedring af DVC kodning stadig udfordrende.

Denne afhandling tager denne udfordring op ved at foreslå flere iterative algoritmer på forskellige niveauer, f.eks. på bitplane-, band-, og frame-niveau. For at vise det teoretiske grundlag bliver teorien for DVC introduceret. Den første foreslåede algoritme benytter parallel iterativ afkodning ved hjælp af flere LDPC dekodere for at udnytte korrelationen bitplaner imellem. For at forbedre estimeringen af sideinformationen (SI), støjmodelleringen og samtidig lære af de tidligere dekodede Wyner-Ziv (WZ) frames, foreslås "Side Information and Noise learninG"(SING). SING introducerer optisk flow for at kompensere for de svagheder i blokken, der skyldes SI-estimeringen og udnytter også grupperingen af DCT-blokkene for at fange korrelationen forskellige bands imellem, samtidig med at øge den lokale tilpasning i støjmodellering. Under afkodning bliver de opdaterede oplysninger anvendt til iterativt at opdatere den beregnede bevægelse og gendannelse i den foreslåede "MOtion and REconstruction reestimation"(MORE) metode. MORE opdaterer ikke kun bevægelsesvektorerne for at forbedre SI og støj-modelleringen, men kompenserer også residualet af bevægelsen baseret på de tidligere dekodede WZ frames. Endvidere øger MORE gendannelsen ved at foreslå en generaliseret rekonstruktion ved at optimere rekonstruktionen med flere konkurrencedygtige SI'er. Endelig foreslås en adaptiv algoritme til tilstandsbeslutning som drager fordel af skip og intratilstandene i DVC. Dette sker på basis af kvaliteten i nøgle frames og raten for WZ frames. Samlet set forbedrer de foreslåede algoritmer i en væsentlig grad kodningseffektiviteten af DVC og bidrager med værdifulde løsninger for nye videoapplikationer.

Acknowledgements

I would like to thank everyone who, by some means or other, helped me to complete this work. First of all, I would like to express my sincere gratitude to my supervisor Prof. Søren Forchhammer for his support, guidance, and encouragement throughout this research. My supervisor has directed this research with competence, instilling his enthusiasm, and providing support in uncountable occasions. This work would not have been completed without his help, support, and practically infinite supply of comments and ideas. It has been a great honor to work with him during my stay at the Technical University of Denmark. In addition, I would like to express my thanks to my co-supervisors, Prof. Knud J. Larsen in our group and Prof. Mads Nielsen from University of Copenhagen, for their advice and support during my study. I would also like to thank my thesis examiners - Prof. Lars Dittmann from DTU Fotonik, Prof. Fernando Pereira from Instituto Superior Técnico, Portugal, and Prof. Peter Schelkens from Vrije Universiteit Brussel, Belgium - for their comments and suggestions.

I would also like to thank the co-workers of the collaborations during my PhD. Firstly, I would like to thank my co-workers, Xin Huang when he was at our group and Lars Lau Rakêt from University of Copenhagen, for their nice contributions to the success of our joint project. Secondly, I would like to thank Jürgen Slowack from University of Ghent for his precious advice in our collaboration during my research stay at Multimedia Lab, University of Ghent, Belgium. I would like to thank all the members of our Coding and Visual Communication Group in a creative and friendly research environment.

Finally, I would specially like to share a great deal of my achievement with my parents, my lovely family (my wife ThanhThuy, our daughter VietDan, and our little son NguyenKhoi), and my friends, who always support and encourage me in my life.

Publications

The Ph.D. project has resulted in the following contributions directly related to the proposals discussed in this thesis, which are fully reported in Appendix C:

Journals

1. **TIP12:** Huynh Van Luong, Lars Lau Rakêt, Xin Huang, and Søren Forchhammer, "Side Information and Noise Learning for Distributed Video Coding using Optical Flow and Clustering," *IEEE Transactions on Image Processing*, vol. 21, no. 12, pp. 4782-4796, Dec. 2012.
2. **TIP13:** Huynh Van Luong, Lars Lau Rakêt, and Søren Forchhammer, "Reestimation of Motion and Reconstruction for Distributed Video Coding," submitted to *IEEE Transactions on Image Processing*, Apr. 2013. (not in Appendix C but this contribution is reported in the thesis)

Conferences

1. **ICASSP11:** Huynh Van Luong, Xin Huang, and Søren Forchhammer, "Multiple LDPC Decoding using Bitplane Correlation for Trainsform Domain Wyner-Ziv Video Coding," in *IEEE International Conference on Acoustics, Speech and Signal Processing 2011 (ICASSP 2011)*, Prague, Czech Republic, May 2011.
2. **ICIP11:** Huynh Van Luong, Xin Huang, and Søren Forchhammer, "Parallel Iterative Decoding of Transform Domain Wyner-Ziv Video using Cross Bitplane Correlation," in *IEEE International Conference on Image Processing 2011 (ICIP 2011)*, Brussels, Belgium, Sep. 2011.
3. **MMSP11-1:** Huynh Van Luong, Xin Huang, and Søren Forchhammer, "Adaptive Noise Model for Transform Domain Wyner-Ziv Video using

Clustering of DCT Blocks," in *IEEE International Workshop on Multimedia Signal Processing 2011 (MMSP 2011)*, Hangzhou, China, Oct. 2011.

4. **MMSP11-2:** Xin Huang, Lars Lau Rakêt, Huynh Van Luong, Mads Nielsen, Francois Lauze, and Søren Forchhammer, "Multi-hypothesis Transform Domain Wyner-Ziv Video Coding including Optical Flow," in *IEEE International Workshop on Multimedia Signal Processing 2011 (MMSP 2011)*, Hangzhou, China, Oct. 2011. (Top 10% paper award)

5. **PCS12:** Huynh Van Luong and Søren Forchhammer, "Noise Residual Learning for Noise Modeling in Distributed Video Coding," in *Picture Coding Symposium 2012 (PCS 2012)*, Krakow, Poland, May 2012.

6. **SPIE12:** Lars Lau Rakêt, Jacob Søgaard, Matteo Salmistraro, Huynh Van Luong, and Søren Forchhammer, "Exploiting the Error-Correcting Capabilities of Low Density Parity Check Codes in Distributed Video Coding using Optical Flow," in *SPIE Optics+ Photonics, Optical Engineering+Applications*, San Diego, California, Aug. 2012.

7. **PCS13:** Huynh Van Luong, Jürgen Slowack, Søren Forchhammer, Jan De Cock, and Rik Van de Walle, "Adaptive Mode Decision with Residual Motion Compensation for Distributed Video Coding," submitted to *Picture Coding Symposium 2013 (PCS 2013).* (not in Appendix C but this contribution is reported in the thesis)

The Ph.D. project has also resulted in the contributions (not in Appendix C):

1. **WITMSE11:** Søren Forchhammer, Huynh Van Luong, and Xin Huang, "Multiple LDPC Decoding for Distributed Source Coding and Video Coding" in *Workshop on Information Theoretic Methods in Science and Engineering 2011 (WITMSE 2011)*, Helsinki, Finland, Aug. 2011.

2. **DCC12:** Søren Forchhammer, Matteo Salmistraro, Knuds J. Larsen, Xin Huang, and Huynh Van Luong, "Rate-adaptive BCH coding for Slepian-Wolf coding of highly correlated sources," in *Data Compression Conference 2012 (DCC 2012)*, Snowbird, Utah, Apr. 2012.

Contents

List of Figures

List of Tables

Acronyms

AMD Adaptive Mode Decision

AVC Advanced Video Coding

BER Bit Error Rate

BP Belief-Propagation

BSC Binary Symmetric Channel

CRC Cyclic Redundancy Check

DCT Discrete Cosine Transform

DSC Distributed Source Coding

DVC Distributed Video Coding

GOP Group Of Pictures

ICL Ideal Code Length

LDPC Low Density Parity Check

LDPCA LDPC Accumulate

LLR Log-Likelihood Ratio

LSB Least Significant Bit

ML Maximum Likelihood

MMSE Minimum Mean-Square Error

MORE MOtion and REconstruction reestimaton

MSB Most Significant Bit

NRR Noise Residue Refinement

OBMC Overlapped Block Motion Compensation

OF Optical Flow

PID Parallel Iterative Decoding

PSNR Peak Signal-to-Noise Ratio

QCIF Quarter Common Intermediate Format

QP Quantization Parameter

RD Rate Distortion

RMC Residual Motion Compensation

SI Side Information

SING Side Information and Noise learninG

SW Slepian-Wolf

TDWZ Transform Domain Wyner-Ziv

WVSN Wireless Visual Sensor Network

WZ Wyner-Ziv

Chapter 1

Introduction

1.1 Motivation

Digital video is continuously growing in a wide range of emerging applications. Increasing practical applications in video communications such as Wireless Visual Sensor Networks (WVSNs) and mobile phone cameras require low complexity encoding, where conventional video standards as H.264/AVC are disadvantageous. DVC [1] is a new coding paradigm which entails low-complexity encoders as well as separate encoding of correlated video sources. DVC is suitable for the applications where the computational burden is moved from encoder to decoder. This is particularly attractive for upstream transmissions such as camera systems in visual sensor networks, where camera sensors require a simple encoder while base stations can decode with high computational burden. Therefore, the challenge in such systems makes data compression and resource constraints key issues which are needed to be solved.

DVC is promising for the WVSNs that have constrained resources in terms of battery, memory, processing capability, and data rate in error-prone environments. The WVSNs are challenged by requiring advanced video coding and processing techniques in the energy-constrained wireless communications. One of the main design objectives of the WVSNs is a local (on-board) coding and processing technique with high compression efficiency, low-complexity, and error-resilience. The WVSNs also require real-time performance for the process extracting visual information from physical environments (by cameras) to transmit it to control centers (by users). Thus, most camera sensors have embedded processors that only support lightweight processing algorithms. Figure

1.1 illustrates a WVSN including sensor node such as a MICAz sensor mote [2] and a wireless endoscopy Pill-Cam ESO2 [3]. These wireless visual sensors capture information to send to the base station (sink) which is connected to users through Internet or satellite. At control center, users can issue monitoring queries and display results obtained from the WVSN.

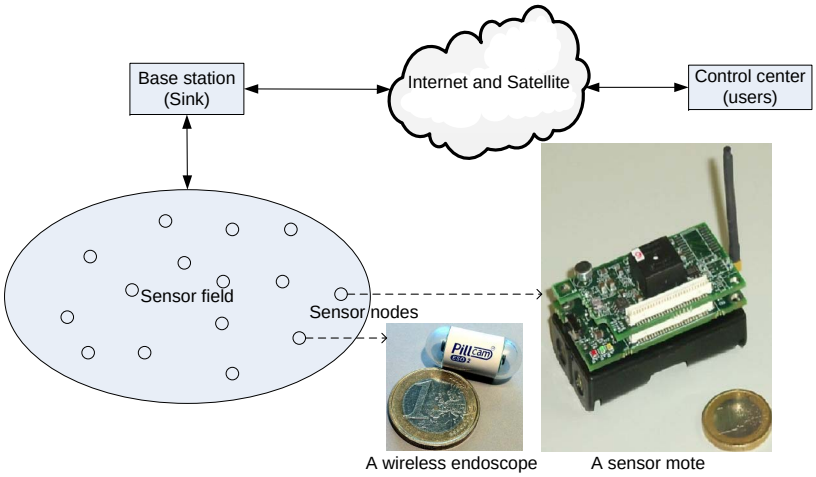


Figure 1.1: A wireless visual sensor network with a Pill-Cam ESO2 endoscope and a MICAz sensor mote.

The WVSNs have a range of applications including surveillance networks, health care systems, and monitoring systems. The surveillance visual sensors combined with signal processing and computer vision techniques can be used to locate criminals, terrorists, or accidents. The sensor networks can be integrated with other multimedia networks to provide health care services. Remote medical centers are able to perform advanced remote monitoring of their patients via multimedia sensors with remote assistance services. The wireless capsule endoscopy provides visual recordings inside the human body for diagnosis and monitoring. The WVSN is possibly a part of advance health informatics challenge, which is one of the grand challenges [4] enabling a new system of distributed tools to collect medical data. In addition, the monitoring systems using visual sensors are used to monitor natural environment, health of human-made structures, e.g bridges, building, ships, etc., and disasters. Multimedia sensors can be used to monitor and control the industrial processes and systems in critical conditions.

The emerging applications require the visual sensors to be exploited using a high efficiency lower power data compression technique, i.e. lightweight encoders, still retaining high compression efficiency, and error-resilience. Most practical visual sensor platforms use intra-frame coding [5], such as JPEG or modified JPEG compression, to offer low-complex high efficiency compression. Despite the potential benefits of DVC with high coding efficiency, low power, and error resilience, none of the existing WVSN platforms and prototypes have implemented or tested DVC [5]. In principle, the Slepian-Wolf [6] and Wyner-Ziv [7] theorems show that the DVC can achieve the same performance as conventional (non-distributed) codings. There are growing endeavors in research societies to improve the Rate Distortion (RD) performance and deal with major obstacles of DVC in practical applications under the constraints, where recent results [8] show that the DVC codec gives a better RD performance than low-complex H.264/AVC intra-coding except for very complex motion sequences. However, there is still a gap between the coding efficiency performances of DVC and H.264/AVC. The DVC coding efficiency is critically dependant on generating high accuracy SI at the decoder and estimating correlation between the corresponding source and the SI. Moreover, exploiting different coding modes with adaptive techniques and controlling rate under delay constraints can also be challenges towards an efficient and practical DVC for the emerging applications.

1.2 Objectives

The goal of this thesis is to develop novel algorithms for an advanced and efficient DVC architecture to improve the DVC coding efficiency. The proposed improvement algorithms are implemented and evaluated on a popular and efficient approach to DVC, which is TDWZ video coding with a feedback channel [1]. The significant improvements of the proposed DVC codecs are also compared with those of the existing DVC codecs and conventional video H.264/AVC codecs. The main objectives of this thesis include:

- Developing advanced and novel algorithms to efficiently estimate noise correlation between source and side information at different levels, e.g. bitplane, band, and frame levels, taking place at the decoder to improve compression performance without changing the complexity video encoding.
- Reestimating, learning, and optimizing information from previously de-

coded information and from multiple sources including spatial correlation, block-based and optical flow-based side information to improve decoding and reconstructing processes.

- Exploiting and integrating the proposed techniques into a TDWZ DVC to evaluate the coding performance of the proposed DVC codecs compared with the existing DVC codecs [9, 10] as well as conventional hybrid predictive video coding such as H.246/AVC.

1.3 Contributions

The thesis contributes a number of solutions at different levels, e.g. bitplane, band, and frame levels at the decoder side to improve the DVC coding efficiency. The main contributions from this thesis are:

- **Parallel iterative decoding using multiple LDPC Accumulate (LDPCA) decoders**: is proposed to utilize cross bitplane correlation [11, 12] by iteratively refining the soft-input, updating a modeled noise distribution and thereafter enhancing the bitplane decoding performance. This parallel iterative decoding exploits a Belief-Propagation (BP) algorithm to propagate soft information back and forth at both bitplane (bit) and coefficient (symbol) levels. The cross bitplane correlation model is able to recalculate the soft-input based on the outputs of LDPCA decoders and update the estimated noise distribution from the noise model. Consequently, the DVC scheme employing this technique reduces the bit rate of WZ frames and improves the rate-distortion (RD) performance of TDWZ.

  This contribution is presented in Chapter 3, which has resulted in the following publications (Papers **ICIP11-ICASSP11**). In Chapter 3, Wyner-Ziv codec with parallel iterative decoding is described in Sec. 3.2 using Sec. 3 in Paper **ICIP11**, where the working flow of the parallel iterative decoding algorithm is added. In addition, performance evaluation in Sec. 3.3 (Chapter 3) shows the experimental results in Sec. 4 (Paper **ICIP11**) and additionally compares Bjøntegaard relative bit-rate savings (%) and PSNR improvements of TDWZ codecs (in [13] and Paper **ICASSP11**) with those of the proposed TDWZ (Paper **ICIP11**) over DISCOVER [9].

  ICIP11: Huynh Van Luong, Xin Huang, and Søren Forchhammer, "Parallel Iterative Decoding of Transform Domain Wyner-Ziv Video us-

ing Cross Bitplane Correlation," in *IEEE International Conference on Image Processing 2011 (ICIP 2011)*, Brussels, Belgium, Sep. 2011.

ICASSP11: Huynh Van Luong, Xin Huang, and Søren Forchhammer, "Multiple LDPC Decoding using Bitplane Correlation for Transform Domain Wyner-Ziv Video Coding," in *IEEE International Conference on Acoustics, Speech and Signal Processing 2011 (ICASSP 2011)*, Prague, Czech Republic, May 2011.

- **Side Information and Noise learninG (SING)**: is proposed using Optical Flow (OF) and clustering of DCT blocks [8] to improve side information and noise modeling and learn information from the previously decoded WZ frames. The optical flow technique is exploited at the decoder side to compensate for weaknesses of block-based methods, when using motion-compensation to generate side information frames. Clustering [14, 15] is introduced to capture cross band correlation and increase local adaptivity in the noise modeling. Furthermore, learning techniques from previously decoded (WZ) frames are also proposed to influence the noise distribution of the current frame. Different techniques are combined by calculating a number of candidate soft side information for (LDPCA) decoding using a multiple soft input decoding approach. Finally, a new SING TDWZ video scheme is proposed based on enhancing the basic TDWZ with optical flow in a multi-hypothesis set-up and the novel clustering for noise modeling.

  This contribution is presented in Chapter 4, which has resulted in the following publications (Papers **TIP12**-**PCS12**-**MMSP11-1**). Adaptive noise model and noise residual learning in Secs. 4.1-4.2 (Chapter 4) use Sec. IV in Paper **TIP12**. The proposed SING scheme in Sec. 4.3 (Chapter 4) is presented using Sec. V in Paper **TIP12**. Furthermore, performance evaluation in Sec. 4.4 (Chapter 4) shows more results, e.g. GOP4, than in Sec. VI in Paper **TIP12**.

  TIP12: Huynh Van Luong, Lars Lau Rakêt, Xin Huang, and Søren Forchhammer, "Side Information and Noise Learning for Distributed Video Coding using Optical Flow and Clustering," *IEEE Transactions on Image Processing*, vol. 21, no. 12, pp. 4782-4796, Dec. 2012.

  PCS12: Huynh Van Luong and Søren Forchhammer, "Noise Residual Learning for Noise Modeling in Distributed Video Coding," in *Picture Coding Symposium 2012 (PCS 2012)*, Krakow, Poland, May 2012.

MMSP11-1: Huynh Van Luong, Xin Huang, and Søren Forchhammer, "Adaptive Noise Model for Transform Domain Wyner-Ziv Video using Clustering of DCT Blocks," in *IEEE International Workshop on Multimedia Signal Processing 2011 (MMSP 2011)*, Hangzhou, China, Oct. 2011.

- **MOtion and REconstruction reestimaton (MORE)**: is proposed using optical flow reestimation to reestimate SI and noise residue, a residual motion compensation to improve the noise residue based on the reconstructed WZ frames, and a generalized reconstruction to optimize the multiple hypothesis reconstruction. A motion reestimation technique is based on optical flow to improve side information and noise residue frames by taking partially decoded information into account. To improve noise modeling, a noise residual motion reestimation technique is proposed by using residual motion compensation with motion updating to estimate a current residue based on previously decoded frames and correlation between estimated side information frames. In addition, a generalized reconstruction algorithm is proposed to optimize a multi-hypothesis reconstruction. The proposed techniques using the motion and reconstruction reestimation are integrated in the SING TDWZ to create a new MORE TDWZ scheme, which significantly improves the RD performance.

  This contribution is presented in Chapter 5, which has resulted in the following submitted paper (Paper **TIP13**). Sec. III presenting the OF works in Paper **TIP13** based on the OF in [16] is not included in Chapter 5.

  TIP13: Huynh Van Luong, Lars Lau Rakêt, and Søren Forchhammer, "Reestimation of Motion and Reconstruction for Distributed Video Coding," submitted to *IEEE Transactions on Image Processing*, Apr. 2013.

- **An adaptive mode decision**: is proposed to take advantage of skip and intra mode based on the quality of key frames and the rate of WZ frames. The adaptive mode decision is also combined with the residual motion compensation to improve noise distribution estimation for a more accurate mode decision. To take advantage of both the refinement technique in [8, 10] and the decoder-side mode decision in [17], the mode decision uses estimated rate to form the adaptive mode decision and combined with a residual motion compensation to generate a more accurate correlation noise. The proposed technique is integrated in the DVC codecs [8, 10] to enhance the RD performance of the TDWZ scheme.

This contribution is presented in Chapter 6, which has resulted in the following submitted paper (Paper **PCS13**). Besides the description in Paper **PCS13**, the final DVC scheme based on the MORE codec with the adaptive mode decision and its experimental results are additionally presented in Chapter 6.

PCS13: Huynh Van Luong, Jürgen Slowack, Søren Forchhammer, Jan De Cock, and Rik Van de Walle, "Adaptive Mode Decision with Residual Motion Compensation for Distributed Video Coding," submitted to *Picture Coding Symposium 2013 (PCS 2013).*

1.4 Outline of the Thesis

The thesis is organized as follows. The foundations and practical codecs of DVC are introduced in Chapter 2. In Chapter 3, the parallel iterative decoding using multiple Low Density Parity Check (LDPC) decoders is proposed. Chapter 4 describes the proposed SING scheme where the DVC performance is compared to several previous DVC codecs. In Chapter 5, the MORE scheme is proposed based on the SING scheme to exploit motion and reconstruction reestimation where the MORE's performance improvements are depicted. Furthermore, the proposed adaptive mode decision is shown in Chapter 6 to consider the advantage of promising coding modes in DVC . Finally, Chapter 7 summarizes this thesis and discusses some possible future directions.

Chapter 2

Distributed Video Coding

Distributed video coding is an interesting instance of Distributed Source Coding (DSC) that was attracted the interest of many researchers. DSC is an instance of source coding dealing with multiple correlated information sources in a distributed context. This chapter introduces two information theory theorems as basis of DSC, the Slepian-Wolf theorem [6] and the Wyner-Ziv theorem [7]. The Slepian-Wolf theorem states for lossless coding of correlated sources that the optimal rate by joint encoding and joint decoding can be achieved by independent encoding and joint decoding. The Wyner-Ziv theorem extends this result for lossy coding of correlated data sets when independent encoding and joint decoding are performed utilizing the correlation between the sources only at the decoder side. In addition, this chapter reviews practical solutions for the Slepian-Wolf coding [18] including practical codec design and rate adaptation for asymmetric SW coding. This chapter also reviews one efficient approach to DVC, which is Transform Domain Wyner-Ziv (TDWZ) video coding. The first practical DVC architectures are from Stanford University [1] and the University of California, Berkeley [19]. Later, the well known DISCOVER DVC codec [9] was introduced and now used as a good DVC benchmark in literature. There have been some recent advances [20–24] to improve the coding efficiency of DVC. More recently, an enhanced DVC scheme [10] has been proposed by utilizing the cross-band correlation, which is used as a starting framework for the contributions presented in this dissertation.

2.1 Distributed Source Coding

This section provides an overview of source coding, where lossless and lossy source codings are characterized, and the fundamental theorems in DSC. Two

information theorems, namely the Slepian-Wolf theorem [6] and the Wyner-Ziv theorem [7], are introduced. These theorems' results are promising efficient lossless and lossy coding of correlated source data sets when independent encoding and joint decoding are performed by utilizing the correlation between the sources only at the decoder side.

2.1.1 Source Coding

The lossless source coding [18] describes a source sequence with bit strings that the original source can be recovered without loss. The lossless source coding maps L samples of the source sequence to the set of bit strings of a fixed length N. The performance of a lossless source code can be measured by the ratio N/L of the number of bits N of this bit string to the number of source samples L. An achievable rate $R = N/L$ is a ratio that allows the reconstruction error to go to zero as the source sequence length goes to infinity. In many source coding issues, the available bit rate is not sufficient to code the information source lossless. We want to use the available rate to describe the source to within the smallest possible average distortion D, which is determined by a distortion function $d(.,.)$, a mapping from the source s and reconstruction $\hat{s}$ alphabets to non-negative reals, $\Re^+$. The mean-squared error is widely used, that is, $d(s,\hat{s}) = |s-\hat{s}|^2$. Thus lossy source coding is to deal with the achievable trade-offs between rate and distortion which can be characterized by the rate-distortion function.

2.1.2 Slepian-Wolf Theorem

We consider the coding scenario illustrated in Figure 2.1 [18], where two source streams S_1 and S_2 are dependent on each other. The coding question now involves two separate source codes that appear at rates R_1 and R_2, respectively, and a receiver where the source codes are jointly decoded. This setting (Figure 2.1) shows a case as a nonasymmetric Slepian-Wolf (SW) coding. Specifically, if we assume $R_2 > \log|S_2|$, the decoder would know S_2 without error, and thus this problem includes the special case of side information at the decoder. The sources S_1 and S_2 play different roles, thus the scheme is usually referred to as asymmetric SW coding (Figure 2.3). The Slepian-Wolf theorem [6,18] is stated in Theorem 2.1.

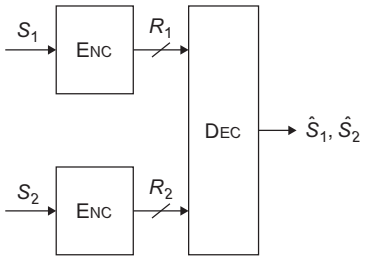


Figure 2.1: Distributed source coding problem.

THEOREM 2.1 *Given discrete sources S_1 and S_2, define $\mathcal{R}$ as*

$$\mathcal{R} = \left\{ (R_1, R_2) : R_1 + R_2 \geqslant H(S_1, S_2), R_1 \geqslant H(S_1|S_2), R_2 \geqslant H(S_2|S_1) \right\}. \tag{2.1}$$

Let $\mathcal{R}^0$ be the interior of $\mathcal{R}$. Then the theorem [6] proves that $(R_1, R_2) \in \mathcal{R}^0$ are achievable for the two terminal lossless source coding problem, and $(R_1, R_2) \notin \mathcal{R}$ are not.

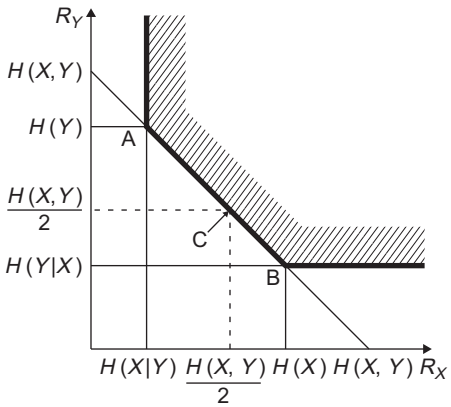


Figure 2.2: Achievable rate region for the coding of correlated sources X and Y.

We consider Theorem 2.1 applying on two sources X and Y with rates R_X and R_Y, where X, Y, R_X, and R_Y are here considered as S_1, S_2, R_1, and R_2, respectively, in Theorem 2.1. If the two coders communicate, it is well known from Shannon's theory that the minimum lossless rate for X and Y is

given by the joint entropy $H(X, Y)$. Slepian and Wolf theorem [6] established that this lossless compression rate bound can be approached with a vanishing error probability for infinitely long sequences, even if the two sources are coded separately, provided that they are coded jointly and their correlation is known to both the encoder and the decoder.

The SW region [18] for two discrete sources is an unbounded region with two corner points (see points A and B in Fig. 2.2). At the point A, source Y is compressed at its entropy rate and can therefore be reconstructed at the decoder independently of the information received from other source X. The source Y is called the SI (available at the decoder only). X is compressed at a smaller rate at the conditional entropy $H(X|Y)$ and can therefore be reconstructed only if Y is available at the decoder.

2.1.3 Wyner-Ziv Theorem

The Wyner-Ziv theorem [7] extends the Slepian-Wolf result for lossy coding of correlated sources. Figure 2.3 [18] considers the setting in which the second encoder has an unconstrained rate link to the decoder. This configuration is often referred as the Wyner-Ziv source coding problem as stated in Theorem 2.2 [7, 18].

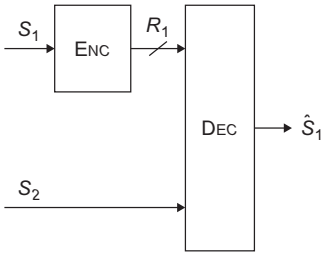


Figure 2.3: The Wyner-Ziv source coding problem.

THEOREM 2.2 *Given a discrete memoryless source S_1, discrete memoryless side information source S_2 with the property that $(S_1(k), S_2(k))$ are i.i.d over k, and bounded distortion function $d : \mathcal{S} \times \mathcal{U} \rightarrow \mathfrak{R}^+$. A rate R is achievable with lossy source coding with side information at the decoder and with distortion D*

if $R > R^{WZ}_{S_1|S_2}(D)$. *Here*

$$R^{WZ}_{S_1|S_2}(D) = \min_{p(u|s_1):\ E[d(S_1,U)]\leqslant D} I(S_1;U|S_2) \tag{2.2}$$

is the rate distortion function for side information at the decoder. Conversely, for $R < R^{WZ}_{S_1|S_2}(D)$, *the rate* R *is not achievable with distortion* D *[7], where* $p(u|s_1)$ *is a conditional distribution of* $u \in U$ *given* $s_1 \in S_1$, $E[.]$ *denotes an expectation operator, and* $I(\ ;\ )$ *denotes the mutual information.*

The Theorem 2.2 provides bounds for DSC systems, where DVC is an interesting specific case. However, building practical DVC systems to achieve those bounds is challenging.

2.2 Distributed Video Coding Architectures

2.2.1 The Slepian-Wolf Coding

This section deals with practical solutions for the Slepian-Wolf coding [18], which refers to the problem of lossless compression of correlated sources with codes that do not communicate. The challenge is here to construct a set of encoders that do not communicate and a joint decoder that can achieve the theoretical limit. This section gives practical design and rate adaptation for asymmetric SW coding.

2.2.1.1 The syndrome approach

If a codeword $\mathbf{x}$ is sent over a Binary Symmetric Channel (BSC) with a crossover probability p and error sequence $\mathbf{z}$, the received sequence is $\mathbf{y} = \mathbf{x}+\mathbf{z}$. Maximum Likelihood (ML) decoding over the BSC searches for the closest codeword to $\mathbf{y}$ with respect to the Hamming distance $d_H(.,.)$.

Let $\mathbf{x}$ and $\mathbf{y}$ be two correlated binary sequences of length n. These sequences are the realizations of the sources X and Y. Figure 2.4 [18] shows a syndrome approach to asymmetric SW coding. The encoder computes and transmits the syndrome of $\mathbf{x} \in \mathcal{C}_\mathbf{s}=\{\mathbf{x}:\ \mathbf{s} = \mathbf{x}H^T\}$. The sequence $\mathbf{x}$ of n input bits is thus

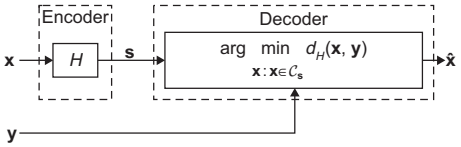


Figure 2.4: The syndrome approach of asymmetric SW coding

mapped into its corresponding $(n-k)$ syndrome bits, leading to a compression ratio of $n:(n-k)$. The decoder, given the correlation between the sources X and Y and the received coset index $\mathbf{s}$, searches for the sequence in the coset that is closest to $\mathbf{y}$ in order to retrieve the original sequence $\mathbf{x}$:

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}:\mathbf{x}\in\mathcal{C}_\mathbf{s}} d_H(\mathbf{x}, \mathbf{y}). \tag{2.3}$$

2.2.1.2 The parity approach

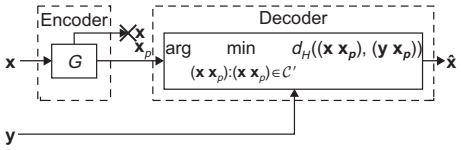


Figure 2.5: The parity approach of asymmetric SW coding

Figure 2.5 [18] shows a parity approach to asymmetric SW coding. Let $\mathcal{C}'$ be an $(n, 2n-k)$ systematic binary linear code, defined by its $(2n-k)\times n$ generator matrix $G = (IP)$: $\mathcal{C}' = \{\mathbf{x}G = (\mathbf{x}\ \mathbf{x}_p) : \mathbf{x} \in \{0,\ 1\}^n\}$. The compression ratio $n:(n-k)$ of the source X is achieved by transmitting only the parity bits $\mathbf{x}_p$ of the source X. The correlation between the source X and the SI Y is modeled as a virtual noise channel, where the pair $(\mathbf{y}\ \mathbf{x}_p)$ is regarded as a noise version of $(\mathbf{x}\ \mathbf{x}_p)$. The decoder corrects the virtual channel noise and thus estimates $\mathbf{x}$ given the parity bits $\mathbf{x}_p$ and the SI $\mathbf{y}$ regarded as a noisy version of the original sequence $\mathbf{x}$ by:

$$\hat{\mathbf{x}} = \arg \min_{(\mathbf{x}\ \mathbf{x}_p):(\mathbf{x}\ \mathbf{x}_p)\in\mathcal{C}'} d_H((\mathbf{x}\ \mathbf{x}_p), (\mathbf{y}\ \mathbf{x}_p)). \tag{2.4}$$

2.2.1.3 Rate adaptation

Using LDPC codes [25], the Belief-Propagation (BP) decoder can be adapted to take into account the syndrome. The syndrome bits are added to the graph such that each syndrome bit is connected to the parity check equation to which it is related. The update rule at a check node is modified in order to take into account the value of the syndrome bit known at the decoder.

In order to select an adequate code and code rate, the correlation between the sources needs to be known or estimated at the transmitter before the compression process. In practical scenarios, this correlation may vary, e.g. the correlation decreases, the rate bound moves away from the estimate. The rate can be controlled by a feedback channel. The decoder could estimate the Bit Error Rate (BER) at the output with the help of the log-likelihood ratios computed by the channel decoder. If the BER at the output of the decoder exceeds a given value, more bits are requested from the encoder. The code should be incremental. In the parity approach, the parity bits are punctured and the decoder compensates for this puncturing. The source sequence $\mathbf{x}$ is compressed through some punctured parity bits $\tilde{\mathbf{x}}_p$. The decoder retrieves the original sequence aided by the SI $\mathbf{y}$. The sequence $(\mathbf{y}\tilde{\mathbf{x}}_p)$ can be considered as a combination of a perfect channel (the unpunctured parity bits), an erasure channel (the punctured parity bits), and a BSC channel (the correlation between $\mathbf{x}$ and $\mathbf{y}$).

One of the rate-adaptive codes for DSC is LDPC Accumulator (LDPCA) [26] which has been widely used in DVC, where the puncturing of LDPC encoded syndromes is investigated in [26]. To avoid degrading the performance of the LDPC code, the syndrome bits are first protected by an accumulator code before being punctured. The combined effect of puncturing and of the accumulator code is equivalent to merging some rows of the parity check matrix. For each rate, a set of parity-check matrices is defined, thereafter, decoding is performed according to the modified sum-product algorithm.

2.2.2 Transform Domain Wyner-Ziv Video

Transform domain Wyner-Ziv video coding [1] is an efficient approach to DVC, where a feedback channel is employed at the decoder to control the rate by requests. DISCOVER codec [9] is an improved DVC based on the initial TDWZ architecture [1]. The architecture of a TDWZ video codec [9] is depicted in Fig.

2.6.

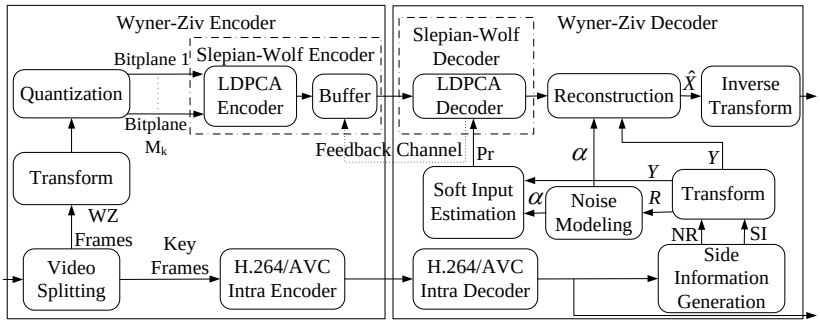


Figure 2.6: Transform domain Wyner-Ziv video codec architecture.

2.2.2.1 The Wyner-Ziv Encoder

- **Video Splitting**: The sequence of frames is split into key frames and so-called WZ frames. Key frames are intra coded using conventional video coding techniques such as H.264/AVC intra coding. Key frames are periodically inserted with a certain GOP size, e.g. a GOP size of 2 means that odd frames are coded as key frames and even frames are WZ frames.

- **Transform**: The Wyner-Ziv frames are transformed by an integer 4×4 Discrete Cosine Transform (DCT). The DCT coefficients are then grouped based on the position of each coefficient within the 4×4 blocks to form the DCT bands.

- **Quantization**: After the transform operation, each DCT band is uniformly quantized by a uniform quantizer with given 2^{M_k} levels (where the number of bitplanes M_k depends on the DCT coefficients). Different qualities can be achieved by different number of quantization levels, 2^{M_k}, used for each DCT band. For instance, eight RD points corresponding to the various 4×4 quantization matrices depicted in Fig. 2.7, where the value at position k indicates the number of quantization levels 2^{M_k} associated with the corresponding DCT coefficients.

- **LDPCA Encoder**: Thereafter, the quantized DCT band is decomposed into bitplanes, which bits of the same significance, e.g most significant bitplane to least significant bitplane, are grouped together. Each bitplane

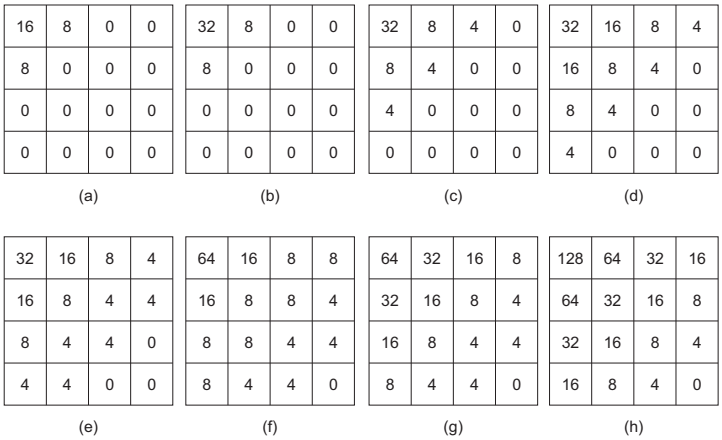


Figure 2.7: Eight quantization matrices associated with 8 RD points

is fed to a rate-compatible LDPC Accumulate (LDPCA) encoder [26] from the Most Significant Bit (MSB) to Least Significant Bit (LSB). The corresponding error correcting information is stored in a buffer and requested by the decoder through a feedback channel.

2.2.2.2 The Wyner-Ziv Decoder

- **Side Information Generation**: At the decoder side, the WZ frame is predicted by using already decoded frames as references. The predicted frame, called the SI frame, is an estimate of the original WZ frame. The SI is created by using frame interpolation by an Overlapped Block Motion Compensation (OBMC)-based or OF methods between key frames that can be coded with constant QPs as defined in Table 2.1. The selection of these QP values for the key frames was chosen so that the average decoded video quality of frames (both key frames and WZ frames) are almost constant. The better the predicted frame SI is estimated, the smaller the bit rate is required for successful decoding. In addition, the residual frame, NR, the estimated difference between the original WZ frame and the SI frame, is also generated for noise modeling.

- **Noise Modeling**: The 4×4 DCT of the transform is then applied on the SI and NR frames to obtain the Y and R frames in transform domain, respectively. The residual frame R, which is the statistics between cor-

Table 2.1: Key frame QPs for 8 RD points

Sequence	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Foreman	40	39	38	34	34	32	29	25
Hall	37	36	36	33	33	31	29	24
Soccer	44	43	41	36	36	34	31	25
Coast	38	37	37	34	33	31	30	26

responding WZ frame X and the SI frame Y is modeled by a Laplacian distribution. The Laplacian parameter α can be estimated at different granularity levels, e.g. frame, band, or coefficient levels.

- **Soft Input Estimation**: Given the available SI Y and the Laplacian distribution α, soft-input information (conditional probabilities Pr for each bit) within each bitplane is estimated. The soft-input Pr is defined as the conditional probability, given the information from the previously decoded bitplanes, of each bit being equal to 0 or 1, which is to be fed into the LDPCA decoder.

- **LDPCA Decoder**: Thereafter the LDPCA decoder starts to decode the bitplanes selected by the quantizer, ordered from most to least significant bitplane, to correct the bit errors. The decoder requests bits from the buffer via the feedback channel until the bitplane is decoded. Thereafter Cyclic Redundancy Check (CRC) check bits are sent for confirmation. Once all the bitplanes of the DCT coefficient band are successfully decoded, the LDPCA decoder starts decoding the next band. This process is carried out until all the DCT coefficient bands are successfully decoded.

- **Reconstruction**: When all the bitplanes are successfully decoded, the WZ frame are decoded through combined de-quantization and reconstruction to get the reconstructed frame $\widehat{X}$.

- **Inverse Transform**: After the 4×4 inverse DCT is performed, the reconstructed pixel domain WZ frame is generated. Finally, the decoded video sequence is obtained by combining the decoded key frames with the WZ frames.

2.2.3 Recent Advances on TDWZ Video

There have been some recent advances on DVC. Noticeably, the work in [20] proposed a SI refinement technique to improve SI during decoding. More recently, a learning based decoding approach was proposed in [21] using overlapped motion vectors for updating the motion field to achieve a better SI quality and a more accurate correlation. To estimate correlation among source and SI, an adaptive correlation estimation integrated in joint bitplane decoding was proposed in [22]. Moreover, an approach in [23] was to develop side information dependant correlation channel estimation in hash-based DVC to express the correlation noise as statistically dependent on SI. Regarding to feedback channel issues, DVC with feedback channel constraints was developed in [24] to constrain number of feedback requests for delay-aware DVC. The RD performance of TDWZ was improved in [10] using a cross-band noise refinement technique. Despite the advances in practical TDWZ video coding, there is still a gap between the RD performance of TDWZ video coding and that of conventional video coding approaches such as H.264/AVC. The cross-band noise refinement for DVC is introduced in next section as a starting framework for the following contributions to further improve the DVC coding efficiency.

2.2.4 The Cross-band TDWZ Video Codec

A cross-band noise model [10] was introduced utilizing cross-band correlation based on the previously decoded neighboring bands. This decoder side cross-band noise model [10] is shown in Fig. 2.8 to improve RD performance of TDWZ video coding. The decoder noise model includes a classification estimation module, which is used by the adaptive noise model. The classification utilizes successfully decoded neighboring lower frequency bands to evaluate the higher frequency bands and classifies coefficients into different categories reflecting their reliability. The adaptive noise model uses a modified maximum likelihood estimator, which is applied to the different reliability classes in order to calculate a higher level noise parameter first. Thereafter, a lower level noise parameter is adaptively determined for each coefficient. Furthermore, a bitplane level Noise Residue Refinement (NRR) scheme was applied in the cross-band decoder to adaptively refine the quality of side information frame during decoding. OBMC was used for side information generation [10].

- **Classification Estimation**: To utilize the cross-band correlation, the

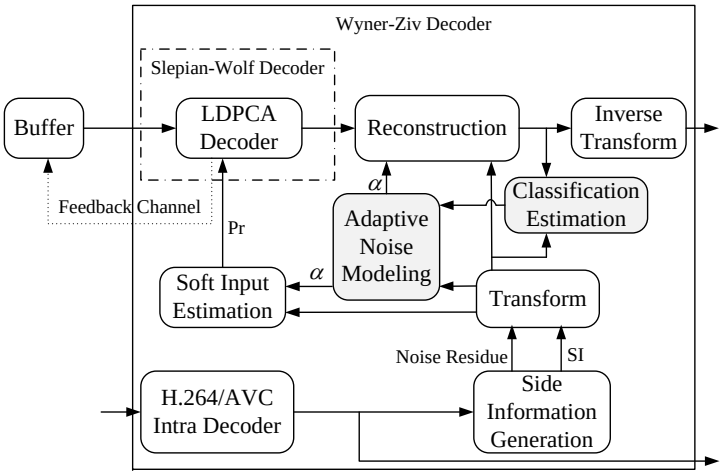


Figure 2.8: Transform domain Wyner-Ziv video codec architecture with a cross-band based adaptive noise model.

classification estimation module classifies the coefficients of each band in to two categories, called classification maps, based on the value of their Laplacian parameters. The cross-band correlation is utilized by using the classification maps of successfully decoded bands to influence the classification map of the current band.

- **Adaptive Noise Modeling**: In order to adapt the Laplacian parameters for each of the classification maps, the modified maximum likelihood estimator [10] is calculated within each of the estimated classification maps. This means that the Laplacian parameters for each map and each band are treated differently in the noise model. This cross-band modeling is combined with a coefficient level noise model using a weighted scaling. Consequently, the Laplacian parameters in the current band are assigned based on both their own coefficient noise parameters and the adaptivity of neighboring decoded bands.

- **Noise Residue Refinement**: The NRR scheme is used to adaptively refine the accuracy of the noise residue during decoding. The refinement is carried out after each successful decoding of a bitplane. Given the decoded bitplane, the error map for the corresponding bitplane is estimated. Thereafter, the NRR refinement for the next bitplane is only applied in positions in the error map. The NRR scheme is trying to scale the partially decoded residue or adaptively balance the weights between

the previous residue and the partially decoded residue to reduce the noise estimation errors for the next bitplane.

2.3 Summary

This chapter provided backgrounds of theoretical and practical results in DVC. The theoretical foundations of DVC were introduced in the context of DSC systems, where correlated sequences were coded using independent encoding and joint decoding. Bounds on the compression performance of such DSC systems were derived by the Slepian-Wolf theorem for lossless source coding and the Wyner-Ziv theorem for extending to lossy source coding. These theoretical contributions were also applied to DVC, where practical solutions for DVC systems were described. First of all, practical issues for the Slepian-Wolf coding were characterized. Thereafter, a TDWZ video codec was described, which was based on information theory results. Some recent advances on DVC to improve the DVC coding efficiency were also reviewed. Finally, the cross-band DVC recently proposed was introduced to improve the coding efficiency of the practical Wyner-Ziv video codec. These DVC architectures are used as a starting point for the contributions presented in the following chapters of this dissertation.

Chapter 3

Parallel Iterative Decoding using Multiple LDPCA Decoders

In this chapter, a parallel iterative LDPC decoding scheme is proposed to improve the coding efficiency of TDWZ video codecs. The proposed parallel iterative LDPC decoding scheme is able to utilize cross bitplane correlation during decoding, by iteratively refining the soft-input, updating a modeled noise distribution and thereafter enhancing the bitplane decoding performance.

TDWZ was first proposed in [27], and thereafter improved by many other techniques, e.g. advanced side information generation schemes [9,28–30], finer noise models [13,28] and refinement schemes [20,31]. To further improve the coding efficiency of TDWZ video coding, a Wyner-Ziv codec with parallel iterative LDPC decoding is proposed in this chapter. The proposed scheme is based on the initial work in [11], inspired by the work in [32] using joint bitplane LDPC decoding and the work in [31] with refinement of the modeled noise distribution. The main advantage of joint bitplane LDPCA decoding is to exploit correlation across bitplanes by exchanging soft information between bitplanes during the decoding. Different from [31, 32], the proposed scheme utilizes multiple LDPCA decoders in parallel, taking inter bitplane correlation into account to iteratively refine the soft-input of bitplanes and update a modeled noise distribution during decoding, thereby improving the overall RD performance of the TDWZ codec. Compared with [11], the novelty is that the modeled noise distribution keeps updating based on the iteratively refined soft-input during parallel decoding.

The rest of the chapter is organized as follows. Section 3.1 presents the TDWZ video codec with rate-adaptive LDPCA codec adopted in this chapter. Section 3.2 describes the proposed parallel iterative LDPC decoding scheme. The performance of the proposed approach is analyzed and compared with other existing methods in Section 3.3.

3.1 TDWZ Video Coding with Rate-adaptive LDPCA Codec

In TDWZ video coding (Sec. 2.2.2), the coding efficiency of the LDPCA codec plays a key role in terms of overall RD performance. At the encoder side, the LDPCA encoder encodes each bitplane that is fed to a rate-compatible LDPC Accumulate (LDPCA) encoder [26] from MSB to LSB. The corresponding error correcting information generated by the LDPCA encoder for each bitplane is stored in the buffer. The amount of information to be transmitted depends on the requests made by the decoder through a feedback channel. At the decoder side, the LDPCA decoder starts to decode the bitplanes selected by the quantizer, ordered from most to least significant bitplane, to correct the bit errors. The decoder requests bits from the buffer until the bitplane is decoded.

For LDPCA decoding, a BP algorithm is used to retrieve each transmitted bitplane. The BP algorithm is a soft-decoding approach, which is passing a Log-Likelihood Ratio (LLR) of Pr back and forth between source nodes and the syndrome nodes. Let $X = (b_{m-1}, ..., b_1, b_0)$ denote a quantized DCT coefficient of a Wyner-Ziv frame, where b_{m-1} is an MSB bit and b_0 is an LSB bit and let Y denote a quantized DCT coefficient of the side information. The LDPCA corrects errors one bitplane after another e.g., from MSB to LSB. The LLR of a bit b_i $(0 \leqslant i \leqslant m-1)$ of the i^{th} significant bitplane is described as:

$$\mathrm{L}(b_i) = \log\left(\frac{\Pr(b_i = 0|Y, b_{m-1}, ..., b_{i+1})}{\Pr(b_i = 1|Y, b_{m-1}, ..., b_{i+1})}\right), \tag{3.1}$$

where $b_{m-1}, ..., b_{i+1}$ represent bits from previous successfully decoded bits of the transformed coefficient. The LDPCA decoder utilizes information from previous successfully decoded bitplanes for decoding future bitplanes.

3.2 Wyner-Ziv Codec with Parallel Iterative Decoding

3.2.1 Multiple LDPCA Decoders Using Cross Bitplane Correlation

In the TDWZ codec described in Section 3.1, the LDPCA decoder utilizes side information, modeled noise correlation and the information from previous decoded bitplanes to decode future bitplanes. One limitation is that the inter bitplane correlation is not fully explored during decoding. Although a refinement scheme is employed in [31] to utilize the bitplane correlation to update the noise distribution, thereby refining soft-input for decoding further bitplanes, the soft-input of the LDPCA decoder is fixed until successful decoding. To overcome the above limitations and improve the performance of the LDPCA codec, a novel Wyner-Ziv codec is proposed in this section to iteratively refine soft-input for each bitplane during the decoding process. The soft estimate of Wyner-Ziv coefficients is used to iteratively update the noise distribution and thereby refine the reliability of soft-input.

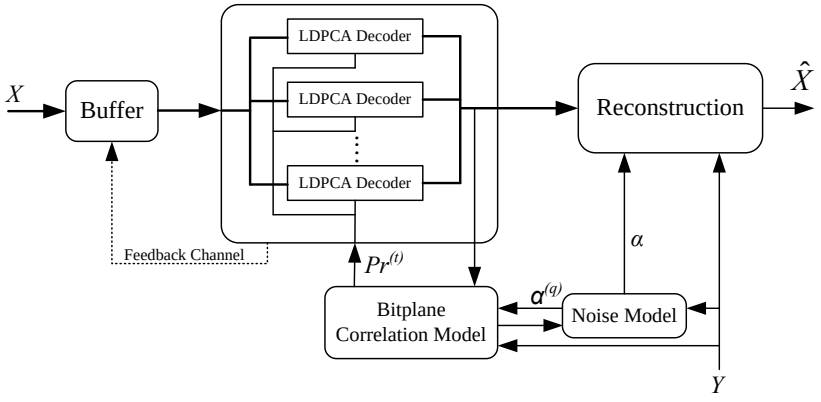


Figure 3.1: Multiple LDPCA Decoders.

The proposed Wyner-Ziv codec is depicted in Fig. 3.1. It mainly includes multiple LDPCA decoders and a bitplane correlation model. The bitplane correlation model is able to recalculate the soft-input based on the outputs of LDPCA decoders and update the estimated noise distribution from the noise

model. The new soft-input information of the source X is estimated by conditioning on Y and using an iteratively refined Laplacian parameter from the noise model. The multiple LDPCA decoders are running in parallel to keep refining the soft-input. Each LDPCA decoder is responsible for one bitplane. Different from single bitplane LDPCA decoding, where the decoder corrects errors one bitplane after another e.g., from MSB to LSB [13] or from LSB to MSB [33], the multiple LDPCA decoders operates on all available bitplanes at once and exploits the correlation between bitplanes and passes information from one bitplane to another. In addition, the soft estimate of each Wyner-Ziv coefficient is iteratively generated to update the noise distribution within the bitplane correlation model. Therefore, the soft-input for decoding is regenerated in a way that exploits the noise correlation between Wyner-Ziv coefficients and the side information coefficients.

3.2.2 Parallel Iterative Decoding Algorithm

The proposed codec employs iterative refinement at both bitplane (bit) and coefficient (symbol) levels. The overall decoding procedure using multiple LDPCA decoders executes the BP algorithm to propagate LLRs back and forth between the syndrome nodes, bit nodes, and symbol nodes [32]. Let $\beta_k = \Pr(b_k = 0)$ define the probability distribution for bit b_k. At bit level, the main difference between our proposed approach and [9] is that the LLR for a bit b_i $(0 \leqslant i \leqslant m-1)$ of the i^{th} significant bitplane is computed conditioned on the binary distributions $(\beta_k, 1-\beta_k)$ of the remaining bits, $b_k (k \neq i)$. This means that the LLR is calculated by using soft information of the other bits. Moreover, the order of full decoding in our approach is not restricted to follow the order of significance of bitplanes. The LLR described in (3.1) only uses the bits from previous successfully decoded bitplanes and decodes from MSB to LSB. Here the LLR expression is generalized for a bit b_i of bitplane i as: The LLR of a bit b_i $(0 \leqslant i \leqslant m-1)$ of the i^{th} significant bitplane is described as:

$$\mathrm{L}(b_i) = \log\left(\frac{\Pr(b_i = 0|Y, \beta_{m-1}, ..., \beta_{i+1}, \beta_{i-1}, ..., \beta_1, \beta_0)}{\Pr(b_i = 1|Y, \beta_{m-1}, ..., \beta_{i+1}, \beta_{i-1}, ..., \beta_1, \beta_0)}\right), \tag{3.2}$$

where $\beta_k (k \neq i)$ are soft-input values for the same coefficient as b_i. In order to approximate the LLR expression (3.2), let $\Pr^{(t-1)}(b_i)$ denote the a priori probability of b_i at iteration $t-1$ at bit level. Note that, at bit level, $\Pr^{(t-1)}(X|Y) = \Pr^{(q-1)}(X|Y)$, where $q-1$ indicates iteration $q-1$ at coefficient level. The denominator and numerator of (3.2) are substituted by applying the sum-product expressions [32, 34] for specific values of $b_i = \{0, 1\}$

and consequently, LLR can be computed via the sum-product algorithm [32,34] as:

$$\mathrm{L}^{(t)}(b_i) = \log\left(\frac{\sum\limits_{X\in S_0} \Pr^{(q-1)}(X|Y) \prod\limits_{k\neq i} \Pr^{(t-1)}(b_k)}{\sum\limits_{X\in S_1} \Pr^{(q-1)}(X|Y) \prod\limits_{k\neq i} \Pr^{(t-1)}(b_k)}\right), \tag{3.3}$$

where $X = (b_{m-1}, b_i, ..., b_1, b_0)$, S indicates the set of values $\{0, 1, 2, ..., 2^{m-1}\}$ for the coefficient X which is coded by m bitplanes (for DC and the magnitude of AC coefficients) and $S_0 = \{X \in S : b_i = 0\}$, $S_1 = \{X \in S : b_i = 1\}$. $\Pr^{(q-1)}(X|Y)$ is calculated at iteration $q-1$ at coefficient level by using the updated noise distribution between the side information coefficient and the original Wyner-Ziv coefficient via the noise model [13] as shown in Fig. 3.1.

Similar to bit level, we can rewrite the expression at coefficient level. Let us have an a priori belief of X conditioning on Y given by the probability distribution $\Pr^{(q-1)}(X|Y)$ and variables $(\beta_{m-1}, ..., \beta_1, \beta_0)$, with likelihood $\Pr^{(q-1)}(\beta_{m-1}, ..., \beta_1, \beta_0|\psi)$, where $\psi = \Pr(X|Y)$, then the posterior probability $\Pr^{*(q)}(X|Y)$ is approximated by:

$$\Pr^{*(q)}(X|Y) \propto \Pr^{*(q-1)}(X|Y)\Pr^{(q-1)}(\beta_{m-1}, ..., \beta_1, \beta_0|\psi). \tag{3.4}$$

Suppose that prior beliefs of $(\psi, \beta_{m-1}, ..., \beta_1, \beta_0)$ are independent, we get an approximation of (3.4):

$$\Pr^{*(q)}(X|Y) \propto \Pr^{*(q-1)}(X|Y) \sum_k \Pr^{(q-1)}(b_k). \tag{3.5}$$

Thereafter $\Pr^{*(q)}(X|Y)$ is normalized and used to update the noise residual coefficient at iteration q by:

$$R^{(q)} = |\sum_{X\in S} X\Pr^{*(q)}(X|Y) - Y|. \tag{3.6}$$

A Laplacian distribution with parameter α is used to model the noise between X and Y. With the updated residue $R^{(q)}$ in (3.6), the Laplacian parameter $\alpha^{(q)}$ is refined according to the noise model in [13]. The resulting soft estimate of Wyner-Ziv coefficient X is denoted:

$$\Pr^{(q)}(X|Y) = \Pr(X|Y, \alpha^{(q)}). \tag{3.7}$$

Since all LDPCA decoders are running in parallel, once a bitplane is successfully decoded, instantaneously, the re-initialization procedure is performed. The new

soft-inputs for the rest of the bitplanes are assigned conditional on the successfully decoded bitplane. The LDPCA decoders with the successfully decoded bitplane will no longer request syndromes from the buffer. Assume b_i is successfully decoded with value 0, then $Pr^{(t)}(bi = 0) = 1$ and the iteration count is reset as $t = 0$. In addition, the remaining unfinished bitplanes $(b_j, j \neq i)$ are re-initialized by $Pr^{(0)}(b_j = 0) = 1/2$. The LDPCA decoders are iteratively operated up to a maximum numbers of iterations $(T_{\max})$ with the given syndrome bits. If they are not successful after $(T_{\max})$ iterations at bitlevel, the soft estimate of source X is iteratively updated as in (3.7). Furthermore, if they are not successful after a maximum number of iterations $(Q_{\max})$ at coefficient level either, the LDPCA decoders request more syndromes (one for each of the bitplanes not fully decoded yet) from the buffer via the feedback channel. Thereafter a new process is started until all the bitplanes of the DCT coefficients of the band are successfully decoded.

In some cases, the required number of syndromes consumed for the LSB is (close to) a maximum number of syndromes denoted by $N_{\max}$, even though there is some correlation. This is due to a (relative) loss in the LDPCA decoder, which may be reduced by first coding the LSB independently and thereafter apply the proposed codec to the remaining bitplanes after decoding the LSB. Thus, an entropy prediction mechanism is proposed to automatically predict these cases. A set of predefined thresholds is utilized to evaluate (up to 3) less significant bitplanes. The evaluation starts from LSB with its marginalized probabilities. For the LSB bitplanes considered, the entropy is estimated based on the updated LLRs from the output of the multiple LDPCA decoders after trying to decode by using the first syndrome, i.e. $n = 1$. The predefined thresholds are experimentally determined to detect bitplanes for which the average estimated entropy of each bit is close to 1. If the estimated entropy of the LSB is larger than its corresponding threshold, the bitplane will be independently decoded. Then the second LSB will be evaluated based on the conditional probabilities and so on. As a result, the coding efficiency in terms of bit-rate is improved. If no LSB bitplanes are decoded first, the basic iterative multiple LDPCA decoding is handled as follows and the intuitive work flow is depicted in Fig. 3.2 for each band, one at a time:

1. Initiate parameters. Number of syndromes $n = 0$; Iteration count: $q = 0$ at coefficient level, $t = 1$ at bit level; For all bits b_i, $\Pr^{(0)}(b_i = 0) = 1/2$.
2. Increase and check conditions.
   a. Syndrome bit condition: Increase $n = n+1$. If $n \geqslant N_{\max}$ then end, else request a new syndrome for all bitplanes not decoded and continue

to Step 2.b.

b. Iteration count condition at coefficient level: Increase $q = q + 1$. If $q \geqslant Q_{\max}$ return to Step 2.a, else go to Step 3.

3. Compute the LLRs. For each bitplane, (3.3) is computed to get the LLRs, $\mathrm{L}^{(t)}(b_i)$, which are forwarded as input to the multiple LDPCA unit for parallel decoding.

4. Check for each bitplane if the LDPCA is successfully decoded.

   a. No: Compute probabilities of bitplanes. New probabilities of bitplanes, $\Pr^{(t)}(b_i)$, are obtained based on the updated LLRs output by the LDPCA.

   b. Yes: Re-initialize the process. Assume LDPCA (b_i) is successfully decoded with value $b_i = 0$, assign $\Pr^{(t)}(b_i = 0) = 1$. Reset iteration count $t = 0$ and the remaining unfinished LDPCA decoders by $\Pr^{(0)}(b_j = 0) = 1/2$.

5. Iteration counts at bit level. Increase $t = t + 1$. If $t < T_{\max}$ return to Step 3, else go to Step 6.

6. Compute the soft estimate of source X at coefficient level. The soft estimate, $\Pr^{(q)}(X|Y)$, is updated by (3.7), where the noise $\alpha^{(q)}$ is computed with the updated residue based on (3.6).

7. Check all LDPCA decoders. The process is ended if all bitplanes are successfully decoded, otherwise, return to Step 2.b. The above procedure is repeated for all bands of the DCT coefficients for which Wyner-Ziv bits are transmitted.

3.3 Performance Evaluation

In this section, the RD performance of the proposed approach is presented and compared with the TDWZ video codec described in Section 3.1 as well as relevant benchmarks. The test sequences are 149 frames of *Foreman, Hall Monitor, Soccer,* and *Coast-guard* with 15Hz frame rate and Quarter Common Intermediate Format (QCIF) format. Group Of Pictures (GOP) size is 2, where the odd frames are coded as key frames using H.246/AVC Intra and the even frames are coded using Wyner-Ziv coding. Eight RD points (Q_j) are considered corresponding to eight 4×4 quantization matrices [9]. The values within these

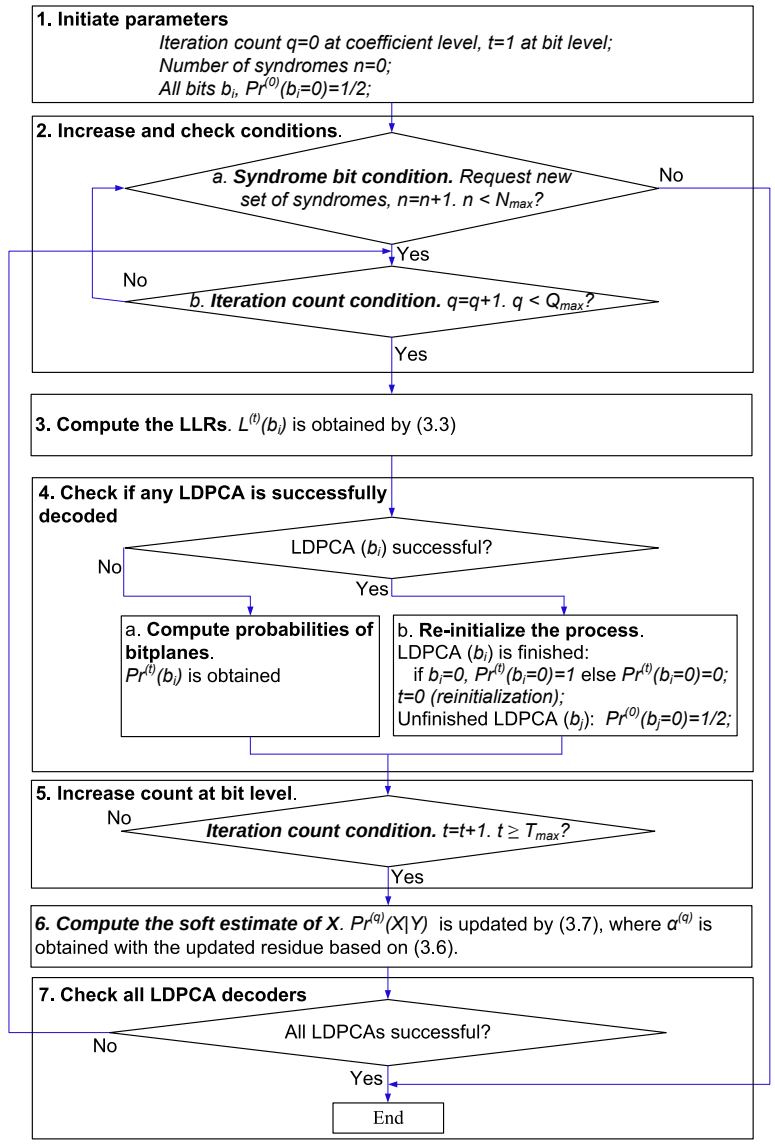


Figure 3.2: Parallel iterative decoding algorithm.

matrices determine the number of bitplanes associated to the DCT coefficient bands, therefore, the number of LDPCA decoding instances is known. The

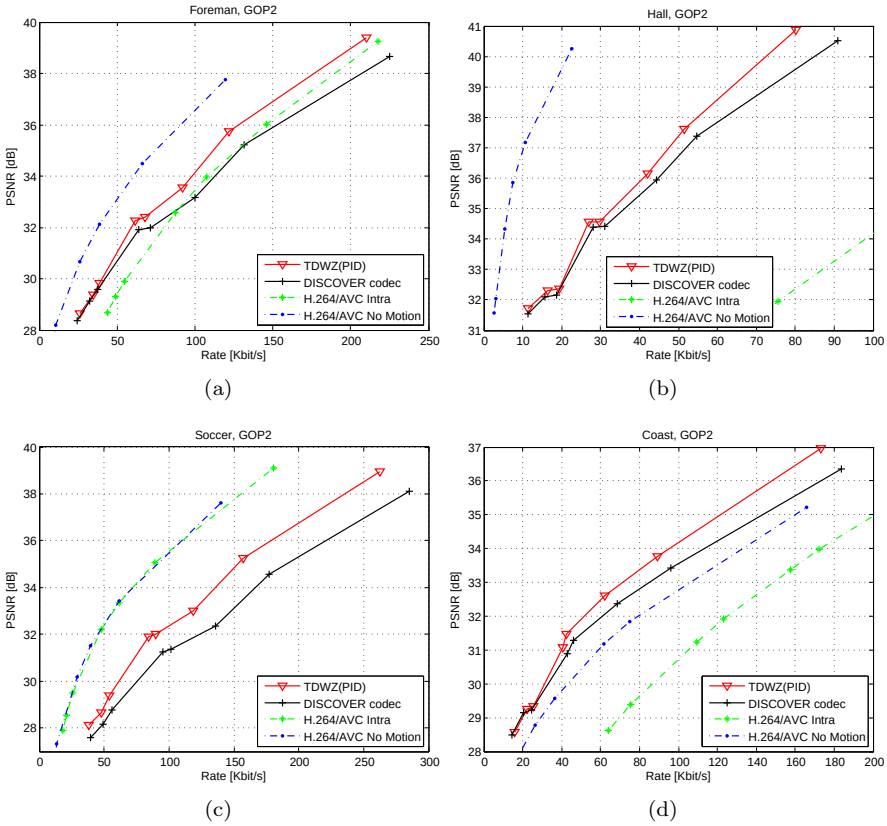


Figure 3.3: PSNR vs. rate for the proposed TDWZ(PID) codec for WZ frames (QCIF, 15Hz, GOP2).

proposed scheme uses m (number of bitplanes of a given band) regular LDPC accumulate decoders [26] with a length of 1584 bits for each. So 1584 transform coefficients per given band of a frame are decoded in parallel at a time by m LDPCA decoders each decoding one bitplane.

Table 3.1 shows the savings in total rate, ΔR (in %), and WZ rate, ΔR_{WZ} (in %) of the proposed TDWZ codec with the Parallel Iterative Decoding (Parallel Iterative Decoding (PID)), denoted by TDWZ(PID), compared with the TDWZ codec in [13], denoted by ICASSP09. The proposed scheme achieves a reduction of bit-rate for WZ frames up to 3.53% for Foreman; 5.61% for Hall

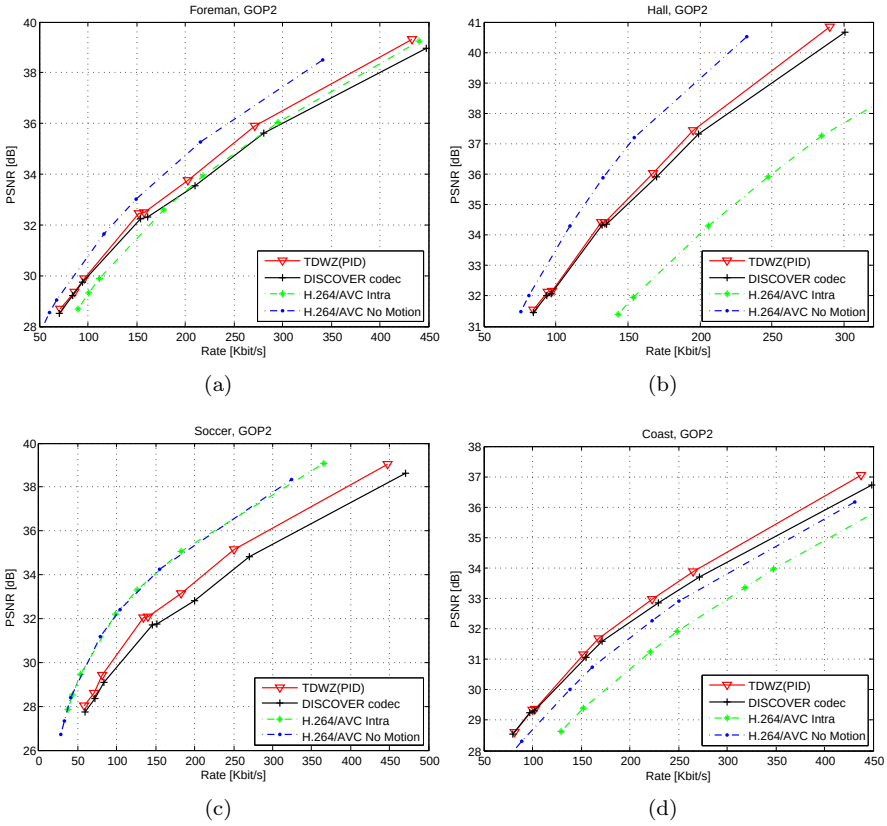


Figure 3.4: PSNR vs. rate for the proposed TDWZ(PID) codec for all frames (QCIF, 15Hz, GOP2).

Monitor; 4.13% for Soccer; 3.75% for Coast-guard. It can be noted that the Peak Signal-to-Noise Ratio (PSNR) values are the same for both the proposed scheme and TDWZ in [13]. In addition, in Tables 3.2-3.3, the relative average bitrate savings for the ICASPP09 codec [13] and the TDWZ codec in [11], denoted by ICASSP11, and the proposed TDWZ(PID) scheme over the DISCOVER codec for WZ frames are 11.97%, 13.86%, 15.53%, respectively (by average of the Bjøntegaard metric [35] for the four test sequences). Overall RD performance of the proposed scheme is depicted in Figs. 3.3-3.4. It can be seen that RD performance has been significantly improved compared with the DISCOVER codec. The performance of H.264/AVC Intra coding and No Motion

Table 3.1: Total rate and WZ rate savings (in %) for the proposed scheme compared with ICASSP09 TDWZ

Q_j	*Foreman*		*Hall*		*Soccer*		*Coast*	
	ΔR	ΔR_{WZ}	ΔR	ΔR_{WZ}	ΔR	ΔR_{WZ}	ΔR	ΔR_{WZ}
1	0.95	2.57	0.79	5.61	2.75	4.13	0.75	3.75
2	1.45	3.53	0.87	4.80	2.65	3.86	0.61	2.65
3	1.28	3.10	0.81	3.95	2.32	3.48	0.65	2.57
4	1.00	2.42	0.72	3.41	1.81	2.85	0.75	2.76
5	1.17	2.69	0.75	3.28	1.91	2.95	0.82	3.15
6	1.51	3.26	0.89	3.43	2.12	3.23	1.21	4.22
7	1.03	2.25	0.89	3.30	1.45	2.29	1.02	2.95
8	0.79	1.62	1.16	4.06	1.18	1.99	0.89	2.22

Table 3.2: Bjøntegaard relative bit-rate savings (%) over DISCOVER for WZ and all frames

Sequence	ICASPP09[13]		ICASSP11[11]		TDWZ(PID)	
	WZ	All	WZ	All	WZ	All
Foreman	11.23	4.84	13.02	5.58	14.13	6.03
Hall	5.96	1.99	7.86	2.43	10.09	2.90
Soccer	20.88	11.47	23.13	12.80	24.48	13.58
Coast	9.80	3.42	11.43	3.87	13.42	4.40
Average	11.97	5.43	13.86	6.17	15.53	6.73

Inter coding are also included. It can be noticed that the TDWZ video coding with the proposed scheme gives a better RD performance than H.264/AVC Intra coding for some sequences, e.g. *Hall Monitor* and *Foreman*, but remain worse than H.264/AVC no motion Inter coding for most of the test sequences. However, the gaps between no motion Inter coding and TDWZ are significantly reduced.

Table 3.3: Bjøntegaard PSNR improvements (dB) over DISCOVER for WZ and all frames

Sequence	ICASSP09[13]		ICASSP11[11]		TDWZ(PID)	
	WZ	All	WZ	All	WZ	All
Foreman	0.50	0.27	0.58	0.31	0.64	0.34
Hall	0.27	0.15	0.36	0.18	0.46	0.21
Soccer	1.06	0.58	1.16	0.64	1.22	0.68
Coast	0.29	0.16	0.35	0.18	0.40	0.21
Average	0.53	0.29	0.61	0.33	0.68	0.36

3.4 Summary

A Wyner-Ziv video codec with parallel iterative LDPC decoding is discussed in this chapter. The technique takes bitplane correlation into account by iteratively refining the soft-input for each bitplane and updating the noise distribution during decoding. Experimental results show that the proposed scheme can improve the coding efficiency of TDWZ in terms of WZ rate savings up to 5.6% compared with the available TDWZ video codec [13]. For a GOP size of 2, an average bitrate saving of 15.5% (or equivalent an average Bjøntegaard improvement in PSNR of 0.7dB) was achieved by the TDWZ(PID) codec for WZ frames compared with the DISCOVER codec.

Chapter 4

Side Information and Noise Learning

This chapter considers Transform Domain Wyner-Ziv (TDWZ) coding and proposes the use of optical flow to improve side information generation and clustering to improve noise modeling. The optical flow technique is exploited at the decoder side to compensate weaknesses of block based methods, when using motion-compensation to generate side information frames. Clustering is introduced to capture cross band correlation and increase local adaptivity in the noise modeling. This chapter also proposes techniques to learn from previously decoded (WZ) frames. Different techniques are combined by calculating a number of candidate soft side information for (LDPCA) decoding.

The DVC coding efficiency is highly dependent on the accuracy of side information at the decoder. A soft-input estimate is calculated at the Wyner-Ziv decoder, obtained by side information frame generation and noise modeling calculated using reference frames [9, 10, 20]. Although the quality of side information frames and the accuracy of the noise model [9] have been improved [10,20], the coding efficiency of TDWZ coding trails that of conventional video coding solutions, such as H.264/AVC, most notably for high motion sequences. We shall consider techniques which can enhance the performance of these basic TDWZ schemes and thereafter integrate the proposed techniques in the DVC codec in Sec. 2.2.4 [10] to enhance performance. As one technique for improved performance, multiple side information based TDWZ has been proposed [30,36]. In [36], two different frame interpolation methods are employed, but the Wyner-Ziv decoder only considers the average of the two estimates for decoding and reconstruction. In [30], the results of frame interpolation and frame extrapolation are combined using weighting to generate multiple soft-inputs to the decoder in a TDWZ scheme. However, the contribution brought

by frame extrapolation is limited and only used for the soft inputs, while for the reconstruction part, only the frame interpolation is used. Providing multiple soft inputs to the Slepian-Wolf (SW) decoder may be seen as a generic way to introduce adaptivity in SW coding and thereby in TDWZ.

In order to enhance performance and reduce the rate-distortion gap between TDWZ and conventional video coding, which is especially pronounced in high motion sequences, a multiple-input TDWZ decoder is used in this chapter. Multiple versions of soft side information are generated by applying both block based and optical flow based side information generation techniques using frame interpolation. The intuition is that optical flow based frame interpolation can generate side information which is different and to some extent may compensate the weaknesses in block based methods, if the scheme allows the techniques to efficiently compensate each other. Optical flow has previously been used in a DVC scheme [37], where the optical flow was calculated using the classical method of Lucas and Kanade [38], which is a local method that can be considered as a limit of block matching. In this chapter a global method for optical flow based on an TV-L^1 energy is used, which should complement block-based approaches better. Furthermore, in contrast to previous multiple soft-input DVC methods [30], the decoding and reconstruction are based on a weighted joint distribution. In this way, the proposed multi-hypothesis based TDWZ decoder will not only reduce the required bitrate for decoding but also improve the quality of reconstructed frames.

The noise estimation is also an important aspect influencing the coding performance. The decoder needs to estimate the correlation between the corresponding source and the side information, which can be obtained through frame interpolation at the decoder side. The accuracy of the correlation has a significant impact on the compression performance of DVC. Our goal is to improve coding efficiency by improving the adaptive noise modeling and by better learning of the correlation between source and side information using both spatial and temporal correlation. Several noise models [9, 10, 39] have been proposed using the Laplacian distribution for the DCT coefficients. The advanced noise models operate with different granularity levels, e.g. frame level, band level, and coefficient level. Estimating the correlation noise has been enhanced by utilizing the correlation of coefficients in each residual frame [9,39,40] and noise residual refinement [10] in the transform domain.

The technique in [40] estimates the correlation noise by first classifying blocks within a frame. A residual energy between source and side information of a given block is used to classify blocks, and for each class a predefined value of the

Laplacian parameter is assigned. In [10], the reconstructed bands were used to influence the noise model for subsequent bands by classifying the reconstructed band into two categories. The cross-band correlation was only based on 1-2 already decoded neighboring bands. Furthermore, two categories may not be enough to fully utilize the correlation. The noise residue refinement [10] updates the estimated noise residue for noise modeling and side information quality during decoding. More recently, an initial work on an adaptive noise model using clustering of DCT blocks was presented [14] to explore cross-band correlation. This technique not only utilizes the correlation over all bands but takes the decoded bands into account to influence the decoding of subsequent bands. In a recent work [22], adaptive correlation is performed integrated in joint bitplane decoding.

In order to further improve the noise estimation, this chapter proposes a refinement technique that utilizes clustering of DCT blocks for cross-band correlation and enhances performance by using the correlation of neighbor coefficients to refine the Laplacian parameter of the coefficient considered, and thereafter, updates the noise parameters. To utilize the temporal redundancy, we shall use residuals of already decoded (WZ) frames to influence the noise distribution of the current frame. As a last enhancement of the noise model, adaptive optimization of the number of clusters in the noise model is addressed to adaptively get the best soft side information during decoding. These improvements of noise modeling are finally combined with the side information generation using optical flow. The techniques are combined using a multiple soft input decoding approach.

The rest of this chapter is organized as follows. The adaptive noise model using clustering of DCT blocks is presented in Section 4.1 along with the new learning techniques proposed in Section 4.2. A new TDWZ video scheme is presented in Section 4.3 based on enhancing the basic TDWZ with optical flow in a multi-hypothesis set-up and the new clustering for enhanced noise modeling. Section 4.4 presents simulation results, analyzes the contributions of the different techniques and compares the performance with reference methods.

4.1 Adaptive Noise Model for Distributed Video Coding

We consider the difference between the original Wyner-Ziv frame X and the side information frame Y. The residual difference, Z, between the transformed coefficients of the WZ frame and the interpolated frame will be modeled by a Lapla-

cian distribution with probability density function $f(z) = (\alpha/2)\exp(-\alpha|z|)$ with variance $\sigma^2 = 2/\alpha^2$.

Rate distortion bounds for simple source models may be derived [18]. Assuming quadratic distortion D and a memoryless source with variance σ^2 and entropy power Q, the upper and lower rate distortion bounds are [18]

$$\frac{1}{2}\log\frac{Q}{D} \leq \Re(D) \leq \frac{1}{2}\log\frac{\sigma^2}{D} \tag{4.1}$$

where $\Re(D)$ denotes the rate at distortion D, the entropy power is

$$Q = (1/2\pi e)\exp(2\mathfrak{h}(Z)), \tag{4.2}$$

and $\mathfrak{h}(Z) = \mathrm{E}[-\log f(Z)]$ denotes the differential entropy of the source Z, where E[.] denotes the expectation operator. For the Laplacian distribution, the entropy power is $Q = (e/\pi)\sigma^2$ [18]. Inserting in (4.1) gives

$$\frac{1}{2}\log\frac{e}{\pi}\frac{\sigma^2}{D} \leq \Re(D) \leq \frac{1}{2}\log\frac{\sigma^2}{D}. \tag{4.3}$$

The bounds in (4.1) may be decreased if the outputs of a given source are split into a number of subsets having different variance and entropy (assuming we also know which subset each sample belongs to). This may be shown based on the concavity of the log and entropy functions, applying Jensen's inequality, $f(\mathrm{E}[Z]) \geqslant \mathrm{E}[f(Z)]$, to $\log\sigma^2$ of the upper bound and the entropy term $-f(Z)\log f(Z)$ of $\mathfrak{h}(Z)$ in the lower bound (4.1). As a result, for a given distortion level, the $\Re(D)$ bounds (4.1) over all clusters are reduced. Below we will describe the process of using clustering for DVC noise modeling.

4.1.1 Adaptive Noise Model Using Clustering of DCT Blocks

The decoder must estimate the statistics of the residual without access to the original frame X. Consistent with the remarks above, it was noted in [14] that the variance of the residual frame based on an estimated residual is higher than the expected variance over the sub-sets (see Appendix B). This means that the estimation at cluster level should be more accurate than at frame level. This motivates reducing the codelength by clustering into sub-sets, which are processed using different parameter values. The techniques proposed in this chapter are based on an initial work on the adaptive noise model using

clustering of DCT blocks [14]. The adaptive noise model considers the (4×4 DCT) transformed residual of frequency bands in a block as components of a (feature) vector.

Let R_h be the residual frame in the transform domain using a frame interpolation scheme h. R_h is used to calculate the parameter of the Laplacian noise distribution $f_{X|Y_h}$. The value of the Laplacian parameter expresses the reliability of the corresponding estimated side information frame. R_h is initialized at the decoder based on the difference between matching blocks of the reference images [10]. Let R_{hk} denote block k out of the N 4×4 blocks in the residual frame R_h, $1 \leq k \leq N$. Each block R_{hk}, considered as a feature vector, contains 16 frequencies given by the transformed residual coefficients. Consider block k of band l and let R^l_{hk} and $\widehat{R}^l_{hk}$ $(1 \leq l \leq 16)$ denote the initial coefficient of the residual and a refined coefficient based on the partially decoded information, respectively. The feature vector of each block $R_{hk} = (\widehat{R}^1_{hk}, \widehat{R}^2_{hk}, ..., \widehat{R}^{l-1}_{hk}, R^l_{hk}, R^{l+1}_{hk}, ..., R^{16}_{hk})$ belongs to the updated residual based on the successfully decoded bands (up to band $l-1$) before decoding band l. This feature vector is classified into one of M clusters, within which an estimate of the noise parameter is calculated. Thus, using clustering of DCT blocks, an adaptive noise model creates M noise parameters, α, one for each cluster.

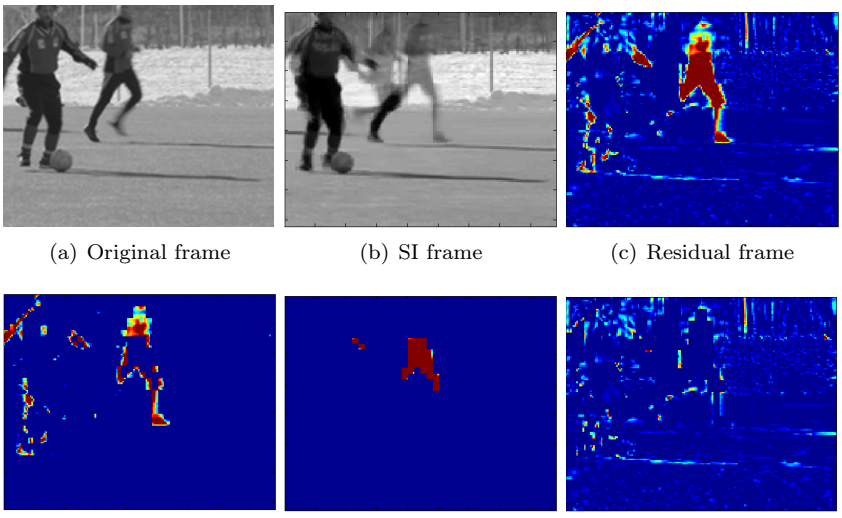
4.1.2 Noise Model B

An extended noise model, which we denote Noise Model B, is obtained by adaptively combining the cluster level noise model in Section 4.1.1 with the noise model in [10]. The clustering technique in [14] was updated at coefficient level and is here extended by updating at bitplane level. A noise residue refinement is exploited at bitplane level and integrated in the DVC scheme in [10]. The refinement is carried out once a bitplane is successfully decoded. The model consists of 4 steps as follows.

Step 1. Clustering of DCT blocks

Our block clustering algorithm is operating on a set of N feature vectors R_{hk}. This set is separated into M subsets or clusters by using Fuzzy-C means clustering, the algorithm is described in Appendix A [41], (the algorithm is configured with the fuzzification degree equal 2 and the predefined termination $\varepsilon = 0.0001$ as in [14].) For block k belonging to cluster j, let $R^l_{hkj} = R^l_{hk}$ denote the coefficients of feature vectors and α^l_{hj} denote the Laplacian noise distribution parameter of cluster j $(1 \leq j \leq M)$ containing N_j elements of

band l, where $\sum_j N_j = N$. Figure 4.1 illustrates an example of clustering of



(a) Original frame (b) SI frame (c) Residual frame

(d) Cluster 1 (e) Cluster 2 (f) Cluster 3

Figure 4.1: An example of clustering *Soccer* frame no. 88 into 3 clusters.

DCT blocks for the *Soccer* sequence where OBMC was used to generate the SI frame (Fig. 4.1(b)). The residual frame in the transform domain R_h (Fig. 4.1(c)) is estimated at the decoder side before decoding the first (DC) band, $l = 1$. Thereafter the residual is classified into 3 clusters ($M = 3$) (Figs. 4.1(d)-4.1(f)).

Step 2. Noise parameter estimation

In band l, a noise parameter, α_{hj}^l, is obtained for each cluster j of the band based on the N_j observations within the cluster. We estimate this Laplacian parameter, α_{hj}^l, based on the variance $\sigma_{hj}^{l^2}$ by

$$\alpha_{hj}^l = \sqrt{2}/\sigma_{hj}^l, \tag{4.4}$$

where $\sigma_{hj}^l = \sqrt{\mathrm{E}[|R_{hkj}^l|^2] - \mathrm{E}[|R_{hkj}^l|]^2}$. As a result, a noise parameter is estimated for each of the M clusters in a given band l.

Step 3. Updating feature vectors

The bands are decoded in a zig-zag order starting from DC and traversing the other (AC) coefficients, $l > 1$, following the order in [10]. Whenever a bitplane of band l is successfully decoded, the coefficients of the band are partially reconstructed and the set of feature vectors is now updated. Thereafter, the set of updated feature vectors is used to refine these vectors by Step 4 below. When all bitplanes are successfully decoded, band l is completely decoded. Subsequently, the set of feature vectors is updated as $R_{hk} = (\widehat{R}^1_{hk}, \widehat{R}^2_{hk}, ..., \widehat{R}^{l-1}_{hk}, \widehat{R}^l_{hk}, R^{l+1}_{hk}, ..., R^{16}_{hk})$ before decoding band $l+1$. This set of updated feature vectors is further refined by Step 4 (below) and thereafter α^{l+1}_{hj} is updated for the next band $l+1$ to be decoded. When all bands are successfully decoded, the process is completed.

Step 4. Refining feature vectors using neighbors

To take advantage of the correlation between the DCT coefficients of the residual of neighbor blocks within each band, a refinement of residuals is proposed. This technique uses neighboring residual coefficients along with the estimated noise parameters. Specifically, Noise Model B refines R^l_{hkj} based on α^l_{hj} and the 8-neighbor residual coefficients, indexed by s and denoted R^l_{hks}. Using the current coefficient R^l_{hk0} and the 8-neighbors, R^l_{hks} with $1 \leq s \leq 8$, a refined R^{*l}_{hkj} ($= R^{*l}_{hk}$ for k in cluster j) is obtained by weighing the neighborhood coefficients as

$$R^{*l}_{hkj} = \sum_{s=0}^{8} \left(\frac{\exp(-\alpha^l_{hj} |R^l_{hkj} - R^l_{hks}|)}{\sum_{t=0}^{8} \exp(-\alpha^l_{hj} |R^l_{hkj} - R^l_{hkt}|)} \right) R^l_{hks}. \tag{4.5}$$

Also assuming a Laplacian distribution for the difference of a coefficient and its neighbors, the weights (4.5) may be seen as likelihood values and the denominator normalizes these. These refined residuals are used in the set of N refined feature vectors, $R^*_{hk} = (\widehat{R}^1_{hk}, \widehat{R}^2_{hk}, ..., \widehat{R}^{l-1}_{hk}, R^{*l}_{hk}, R^{l+1}_{hk}, ..., R^{16}_{hk})$ used for decoding band l. The set is reclassified again by going back to Step 1 and thereafter updating the noise parameter following Step 2 above. Consequently, refined noise parameters α^{*l}_{hj} are obtained using (4.4) based on the observations within the current band for each refined cluster j. The set of α^{*l}_{hj} parameters is denoted by α_1 and together with the set α_0 from [10], they constitute the set of estimates provided by Noise model B IV-B. The resulting coding is referred to as Clustering TDWZ.

4.2 Noise Residual Learning for Adaptive Noise Model

4.2.1 Noise Residual Learning Using Previously Decoded Residual Frames

This subsection extends Noise Model B above by using the previously WZ decoded residual frames to influence the noise distribution of the current frame. A window of previously decoded WZ frames are used to create decoded residual frames corresponding to the WZ decoded frames. The motivation is that the noise distributions based on previously decoded frames are available at the decoder and may be similar to the noise distribution of the current frame. To take advantage of both the previously decoded noise distributions and the estimated current noise distribution, the residuals based on previously decoded frames are used together with the current residual frame to form a larger set of data. This set is classified into clusters to estimate noise parameters for each cluster of the residual frame considered.

Let W be the window size specifying the number of previously decoded WZ frames for the learning process. Let $\widehat{R}_{h(2n-2W)}, ..., \widehat{R}_{h(2n-2)}$ denote residuals based on previously decoded frames and $R_{h(2n)}$ denote the current residual coefficient frame at time $2n$. Let $\widehat{R}_{h(2n-2W)k}, ..., \widehat{R}_{h(2n-2)k}, R_{h(2n)k}$ denote block k, $1 \leq k \leq N$, of N 4×4 blocks of $\widehat{R}_{h(2n-2W)}, ..., \widehat{R}_{h(2n-2)}, R_{h(2n)}$. For each of the residuals based on previously decoded frames, consider a set of N feature vectors $\widehat{R}_{h(2n-2\omega)k}$ with $1 \leq \omega \leq W$, where $\widehat{R}_{h(2n-2\omega)k} = (\widehat{R}^1_{h(2n-2\omega)k}, \widehat{R}^2_{h(2n-2\omega)k}, ..., \widehat{R}^{16}_{h(2n-2\omega)k})$ is given by the residuals of decoded bands. For the current residual frame $R_{h(2n)}$, $R_{h(2n)k} = (\widehat{R}^1_{h(2n)k}, ..., \widehat{R}^{l-1}_{h(2n)k}, R^l_{h(2n)k}, ..., R^{16}_{h(2n)k})$ is the updated residual based on the successfully decoded bands (up to band $l-1$) before decoding band l.

Consider W sets, $S_{h\omega}$, of feature vectors where each set is created by combining N feature vectors $\widehat{R}_{h(2n-2\omega)k}$ of a previous frame with N feature vectors $R_{h(2n)k}$ of the current frame,

$$S_{h\omega} = \{R_{h(2n)}, \widehat{R}_{h(2n-2\omega)}\}. \tag{4.6}$$

Each set $S_{h\omega}$ is classified into M clusters by using Fuzzy C-means clustering as in Appendix A [41]. Thereafter noise parameters $\alpha^l_{h\omega j}$ are obtained based on the observations for each cluster j of band l of set $S_{h\omega}$. As a result, there

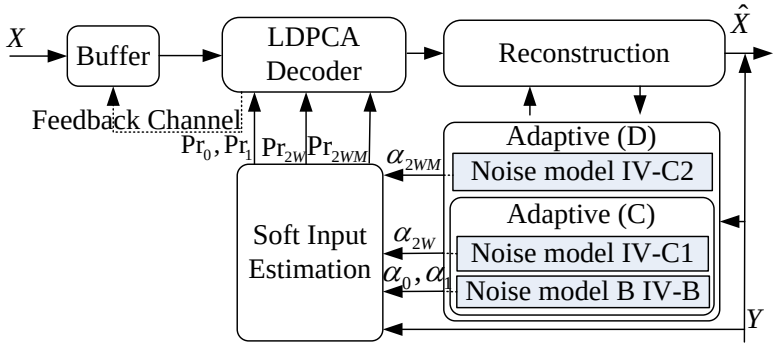


Figure 4.2: TDWZ with adaptive noise model using clustering and noise residual learning.

are W sets of noise parameters for decoding band l for each cluster j, $\alpha^l_{h\omega j}$, $1 \leq \omega \leq W$. The resulting adaptive noise model, denoted by Adaptive (C) and shown in Fig. 4.2, adaptively estimates the noise distribution by creating W different noise parameters $\alpha_{2\omega}$ by Noise model IV-C1. Together with α_0, α_1 obtained from Noise model B IV-B in Section 4.1.2, $\alpha_{2\omega}$ provide multiple inputs to the Soft Input Estimation block.

4.2.2 Adapting The Number of Clusters For Noise Modeling

This part extends the noise residual estimation by selecting the number of clusters, $m \leq M$, giving the best decoding, i.e. optimizing the model order. The statistical characteristics of the noise distribution may change from region to region, and over time when decoding. One reason being, that the noise distribution may not be estimated properly in regions containing moving objects. It may improve the noise modeling, if the noise residual R_h is adaptively modeled using a variable number of noise distributions. A dynamic mechanism is carried out to determine the optimal number of candidate distributions within each frame once a bitplane is successfully decoded.

For each cluster j, m Laplacian distributions, for $1 \leq m \leq M$, are estimated. For each set in (4.6), we apply estimation by (4.4) to obtain the noise parameters $\alpha^l_{h\omega j}$, $1 \leq \omega \leq W$, based on the observations for each cluster j, $1 \leq j \leq m$, of band l of the set $S_{h\omega}$. This results in the set of distributions,

$$D_{hm} = \{\alpha^l_{h\omega mj}\},\ 1 \leq \omega \leq W,\ 1 \leq j \leq m, \tag{4.7}$$

where $\alpha^l_{h\omega mj}$ is the noise parameter estimated for band l of set $S_{h\omega}$ of the

distribution set D_{hm}.

The noise parameters $\alpha^{l}_{h\omega mj}$ of the $W \times M$ candidates are used as candidates for decoding band l for each cluster j. The resulting noise model, called Adaptive (D) (Fig. 4.2), adaptively estimates the noise distribution by creating $W \times M$ candidate noise parameters $\alpha_{2\omega m}$ by Noise model IV-C2, along with $\alpha_{2\omega}, \alpha_0, \alpha_1$ obtained from Adaptive (C) in Section 4.2.1 as input to the Soft Input Estimation block.

4.2.3 Multiple Input LDPCA Decoding

In this subsection, multiple input LDPCA decoding is introduced using multiple soft input candidates when decoding. The multiple input LDPCA decoder tries to decode using each candidate soft side information and then selects the soft side information which converges first during decoding for each bitplane. (Convergence is confirmed using a CRC check.) This way, the decoder adaptively selects the best soft input for decoding. Thereafter, the selected noise parameter set for each bitplane is also used for the minimum mean squared error reconstruction process [36]. It can be noted that the computational cost of the LDPCA decoding is increased. In the worst case, the LDPCA decoding will try to decode using all inputs. In Fig. 4.2, the multiple input LDPCA decoder is used to decode with the candidates denoted $\mathrm{Pr}_0, \mathrm{Pr}_1, \mathrm{Pr}_{2\omega}$, and $\mathrm{Pr}_{2\omega m}$. These candidates are calculated by the Soft Input Estimation using the noise parameters from the Adaptive (D) model. In particular, $\mathrm{Pr}_0, \mathrm{Pr}_1$ are soft inputs calculated based on α_0, α_1, $\mathrm{Pr}_{2\omega}$ based on $\alpha_{2\omega}$, and $\mathrm{Pr}_{2\omega m}$ based on $\alpha_{2\omega m}$. The multiple input LDPCA decoding, as well as the learning technique, is carried out until all bitplanes (of the given quantization level) are successfully decoded. Applying multiple input LDPCA decoding based on the parameters of the Adaptive (D) noise model is referred to as Clustering(learning) TDWZ.

4.3 TDWZ Video with Side Information and Noise Learning

The quality of soft-input information plays a key role in terms of overall RD performance of TDWZ video coding. The quality of the reconstructed frame is highly dependent on the accuracy of the estimated noise distribution $f_{X|Y_h}$. The soft-input Pr is defined as the conditional probability of each bit b_i being equal to 0 or 1, and denoted $\mathrm{Pr} = \mathrm{P}(b_i|y_h, b^{-}; f_{X|Y_h})$, where y_h denotes the corresponding estimated side information value in the transform domain for

bit b_i and b^- is the information from the previously decoded bitplanes. The probability is obtained by marginalizing the estimated conditional probability density function $f_{X|Y_h}$ for the coefficient, which b_i is part of. The essential aspects to improve the coding efficiency of TDWZ video are the quality of the soft-input information fed into the LDPCA decoder and the accuracy of the noise distribution for frame reconstruction.

4.3.1 Multi-hypothesis Based Wyner-Ziv Decoding

To address these issues, multiple input LDPCA decoding (Section 4.2.3) is used. The Wyner-Ziv encoder is not changed. The basic idea is to generate H (> 1) different side information frames Y_h, $h \in [1, H]$, at the decoder for each Wyner-Ziv frame. Each side information frame is considered as an observation of the original Wyner-Ziv frame X with a different amount of noise. The

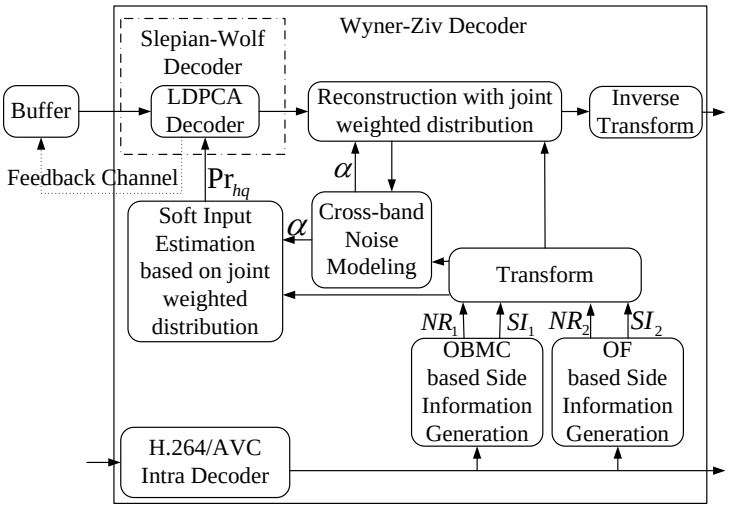


Figure 4.3: Architecture of multi-hypothesis TDWZ video codec based on two frame interpolation schemes.

architecture of the proposed Wyner-Ziv video decoder with an example of two side information generation schemes [42] ($H = 2$) is presented in Fig. 4.3. In principle, there can be any number of competitive side information generation schemes at the proposed Wyner-Ziv decoder. The two interpolation methods in Fig. 4.3 are the OBMC based frame interpolation described in [10] and

the OF based frame interpolation method [8]. As shown in Fig. 4.3, the Side Information Generations generate the side information frames SI_1, SI_2 and the noise residual frames NR_1, NR_2, using OBMC [10] and OF techniques, respectively. SI_2 and NR_2 are generated by first applying OF based Side Information Generation and thereafter 4×4 DCT to I_0 and r_0, respectively. After transformation, each side information generation scheme not only creates an estimate of the Wyner-Ziv frame, Y_h, but also an estimated noise residue frame R_h. R_h is used to estimate the noise between the Wyner-Ziv frame X and its estimated side information frame Y_h. Here based on R_h and Y_h, the coefficient level noise model [10] is used. Each transform coefficient in a given band l is assigned an estimated Laplacian distribution parameter α_h^l.

Using Laplacian parameters based on different calculations of Y_h, multiple soft-inputs are calculated based on a weighted joint distribution. All the hypotheses of soft-input are fed into the multiple input LDPCA decoder (Section 4.2.3). Based on the estimated noise distribution $f_{X|Y_h}$ for each individual side information observation Y_h, a joint weighted distribution F_q is defined as

$$F_q = \sum_{h=1}^{H} u_{hq} f_{X|Y_h}, \tag{4.8}$$

where $q,\ q \in [1, C]$, denotes the index of a candidate joint weighted distribution, C is the total number of candidate joint distributions, and u_{hq} denotes the qth predefined weight on side information $h,\ h \in [1, H]$, u_{hq} are predefined weights, $u_{hq} \geqslant 0$ and $\sum_{h=1}^{H} u_{hq} = 1$. (For the example shown in Fig. 4.3, $H = 2$, $C = 6$).

The frame interpolation schemes, using OBMC and OF [42], employed in this chapter give different results on the different test sequences as shown in Table 4.1. The OBMC and OF techniques may provide complementary results for

Table 4.1: The Average PSNR [dB] Results for Different Side Information Generation Methods (GOP2)

Sequence	Extra	OBMC	OF	OF(learning)
Foreman, QP=25	25.20	29.26	29.28	29.63
Hall, QP=24	33.24	36.46	32.28	35.71
Soccer, QP=25	19.26	21.30	22.43	22.93
Coast, QP=26	28.55	31.83	30.92	30.99

each frame and thus, compensate each other's weaknesses frame by frame and even bitplane by bitplane. We consider a multi-hypothesis TDWZ video codec with two (or more) frame interpolation schemes based on either the OBMC or the OF scheme. Without loss of generality, assume that scheme h is now considered the basic scheme. The soft input calculation is only based on the joint weighted distribution within a specific unreliable region specified by the set Λ_h. Outside of the region Λ_h, the side information is given by the basic scheme h. The values of the Laplacian parameters may express the reliability of the corresponding side information frame. Therefore a set of Λ_h values for each single side information estimation Y_h in band l is determined by evaluating the individual Laplacian parameters and their corresponding mean value by

$$\Lambda_h = \{k | \alpha_h^l(k) < \bar{\alpha}_h^l\}, \tag{4.9}$$

where $\alpha_h^l(k)$ is the Laplacian parameter of side information Y_h at the kth coefficient in band l and $\bar{\alpha}_h^l$ is the mean of all noise parameters in a given band l. Thus Λ_h (4.9) determines a map of coefficients whose noise parameters are potentially less reliable, as they are smaller than the mean value $\bar{\alpha}_h^l$. The unreliable region Ω, which will be processed differently, is defined as a union of the sets Λ_h,

$$\Omega = \bigcup_{h=1}^{H} \Lambda_h. \tag{4.10}$$

The multi-hypothesis soft-inputs using the Y_h as basic scheme are given by

$$\mathrm{Pr}_{hq} = \begin{cases} \mathrm{P}(b_i | y_h, b^-; f_{X|Y_h}) & \text{if } i \notin \Omega \\ \mathrm{P}(b_i | y_1, ..., y_H, b^-; F_q) & \text{if } i \in \Omega \end{cases} \tag{4.11}$$

where Pr_{hq} is the qth candidate soft-input fed into LDPCA decoder, b_i denotes the ith bit in the current bitplane, and $y_1, ..., y_H$ denote different side information values in the transform domain based on diverse side information generation schemes. Again the conditional probability of b_i is obtained by marginalizing the estimated noise distribution $f_{X|Y_h}$ ($i \notin \Omega$) or F_q ($i \in \Omega$). We use the cross-band noise model [10] to calculate $f_{X|Y_h}$ in (4.8) and (4.11). The resulting parameter set is denoted α_{hCB}.

In order to evaluate the quality of the side information, we calculate an Ideal Code Length (ICL) [10], which measures the number of bits required by applying ideal (arithmetic) coding to the given soft-input values if a (non-distributed) encoder would calculate the same soft-input values. $\mathrm{Pr}_{hq}(b_i)$ (4.11) is calculated by reading b_i as the bits after decoding. The code length, $\mathcal{L}$, for one

bitplane is calculated as

$$\mathcal{L} = \sum_{i=1}^{N} -\log \Pr_{hq}(b_i). \tag{4.12}$$

The ICL is obtained as the sum over all bitplanes. This is equivalent to a log-likelihood measure of the coded coefficients.

All the soft-input hypotheses, $\Pr_{hq}$, $q \in [1, C]$ which are calculated by (4.11) are fed into the multiple input LDPCA decoder as in Section 4.2.3. The first converging soft-input is chosen thus reducing the rate of LDPCA decoding. Subsequently, using the selected soft-input, the corresponding joint weighted distribution F_q, $q \in [1, C]$, in the unreliable region Ω is determined. Using the selected joint weighted distribution, F_q, the Minimum Mean-Square Error (MMSE) reconstructed value, x', in the unreliable region Ω is obtained as a generalization of the MMSE expression in [36]

$$x' = \mathrm{E}\left[x | x \in [L, U), y_1, ..., y_H\right] = \frac{\int_L^U x F_q(x) dx}{\int_L^U F_q(x) dx} = \frac{\sum_{h=1}^{H} \int_L^U x u_{hq} f_{X|Y_h}(x) dx}{\sum_{h=1}^{H} \int_L^U u_{hq} f_{X|Y_h}(x) dx} \tag{4.13}$$

where $[L, U)$ are decoded quantization intervals, F_q is the joint weighted distribution (4.8) selected by the LDPCA decoding. The reconstructed value outside the Ω region is calculated following the single side information reconstruction technique based on Y_h as in [36], i.e. for $H = 1$ in (4.13).

4.3.2 Side Information and Noise Learning Using Multiple-hypothesis and Adaptive Noise Modeling

To take advantage of both side information learning (Section 4.3.1) including optical flow [8] and noise learning using clustering (Section 4.2), a TDWZ scheme with **S**ide **I**nformation and **N**oise Learnin**G**, called **SING**(2SI), is proposed. The basic elements of the SING codec are depicted in Fig. 4.4. They consist of OBMC and OF(learning) based side information generations, a noise model using the residual learning as in Section 4.2, the soft-input estimation, the reconstruction using side information and noise learning (Section 4.3.2.2) and multiple input LDPCA decoding. First, the Side Information Generations generate the noise residual frames NR_1, NR_2 and the side information frames,

SI_1, SI_2, using OBMC [10] and OF(learning), where SI_2 and NR_2 are generated, as in Sec. 4.3.1, by the OF(learning) based Side Information Generation [8]. The OF(learning) is a global method for OF based on a total variation energy and its parameters is learnt from previously decoded frames. These are transformed and input to the noise models. For each side information scheme h, noise parameters α_{hRL} are calculated using the Adaptive (D) model (Section 4.2.2) and parameters α_{hCB} are calculated using multi-hypothesis (Section 4.3.1) combined with the cross-band estimate [10] for $f_{X|Y_h}$.

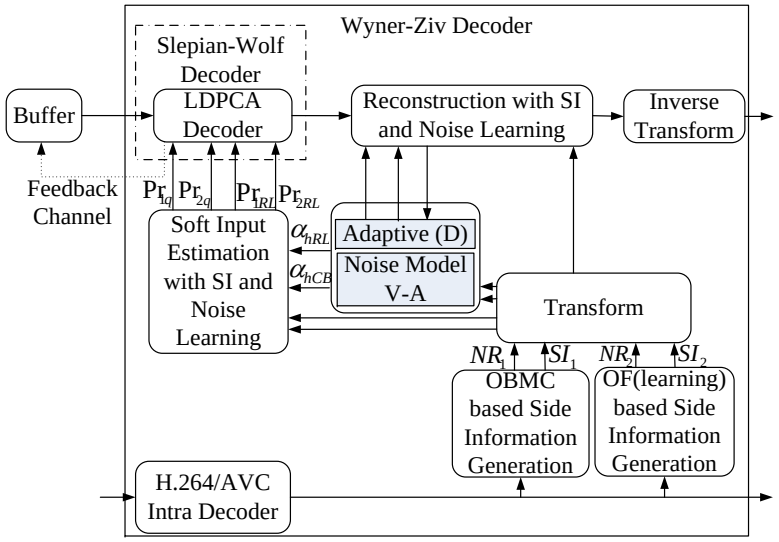


Figure 4.4: TDWZ with Side Information and Noise Learning (SING 2SI).

4.3.2.1 Soft Input Estimation with SI and Noise Learning

Based on the transformed side information frames and the noise parameters, the soft-inputs $\Pr_{1q}$, $\Pr_{2q}$, and $\Pr_{1RL}$, $\Pr_{2RL}$ are calculated. $\Pr_{1q}$ and $\Pr_{2q}$ are calculated by (4.11) based on the cross-band noise (Noise Model V-A) and multi-hypothesis techniques with two OBMC and OF(learning) side information generations as described in Section 4.3.1. $\Pr_{1q}$ are soft-inputs with the OBMC frame interpolation as basic scheme and $\Pr_{2q}$ are soft-inputs with the OF(learning) frame interpolation as basic scheme. $\Pr_{1RL}$, $\Pr_{2RL}$ are obtained by applying the Adaptive (D) model in Section 4.2.2 to each side information generation scheme, here OBMC and OF(learning). All soft-inputs are fed

into the multiple input LDPCA decoder as described in Section 4.2.3. The soft-input which converges first is selected for LDPCA decoding.

4.3.2.2 Reconstruction with SI and Noise Learning

Decided by the selected candidate, the corresponding weighted joint distribution of multi-hypothesis or the corresponding input to the Adaptive (D) noise model is chosen for reconstruction. For instance, if $\Pr_{13}$ is the best candidate, the corresponding weighted joint distribution is F_3 and the multi-hypothesis based scheme is the OBMC based method ($h = 1$ in (4.11)). Consequently, the weighted joint distribution F_3, the multi-hypothesis OBMC based scheme, and the corresponding noise parameter α_{hCB} are used for reconstruction by (4.13). As another example, if $\Pr_{2RL}$ is chosen as the best candidate, α_{2RL} from the Adaptive (D) model, NR_2 and SI_2 from the OF(learning) side information generation are used for the mean squared error reconstruction [36] (4.13) in the reliable region. In the unreliable region Ω, the reconstruction is based on the multi-hypothesis reconstruction corresponding to the basic frame interpolation scheme in (4.13). As we do not have a winner F_q for the joint weighted distribution in this case, the F_q for reconstruction is determined by the corresponding soft-input that has the smallest ICL measured on the decoded data by (4.12) among the soft-input hypotheses. By this approach, the reconstruction takes advantage of different side information generation techniques in the unliable region to achieve a better quality of the reconstructed frames.

4.4 Performance Evaluation

The RD performance of the proposed techniques are evaluated for the test sequences (149 frames of) *Foreman, Hall Monitor, Soccer,* and *Coastguard* with 15Hz frame rate and QCIF format. The GOP size is 2, where odd frames are coded as key frames using H.264/AVC Intra and even frames are coded using Wyner-Ziv coding. Eight RD points are considered corresponding to eight 4×4 quantization matrices [9]. H.264/AVC Intra is here given by the intra coding mode of the H.264/AVC reference codec JM 9.5 [43] in main profile. The parameters for H.264/AVC Intra are set as by DISCOVER [9] and QP values are set to those used for the key frames in the Wyner-Ziv video coding in the DISCOVER codec [9]. It can be noted that only the luminance component of each frame is evaluated. In this chapter, the number of candidate distributions in (4.8) is constrained to $C = 6$, which is an adequate number of candidates to improve performance. For the case $H = 2$, using side information frames generated by OBMC and OF(learning), and $C = 6$, the weighting parameters (4.8) used are $u_{1q} = \{1; 0.8; 0.6; 0.4; 0.2; 0\}$ and $u_{2q} = 1 - u_{1q},\ q \in [1, 6]$. For the case $H = 3$ and $C = 6$, the weighting parameters u_{hq} used are predefined as: $u_{1q} = \{1; 0; 1/2; 1/2; 0; 1/3\}$, $u_{2q} = \{0; 1; 1/2; 0; 1/2; 1/3\}$, and extrapola-

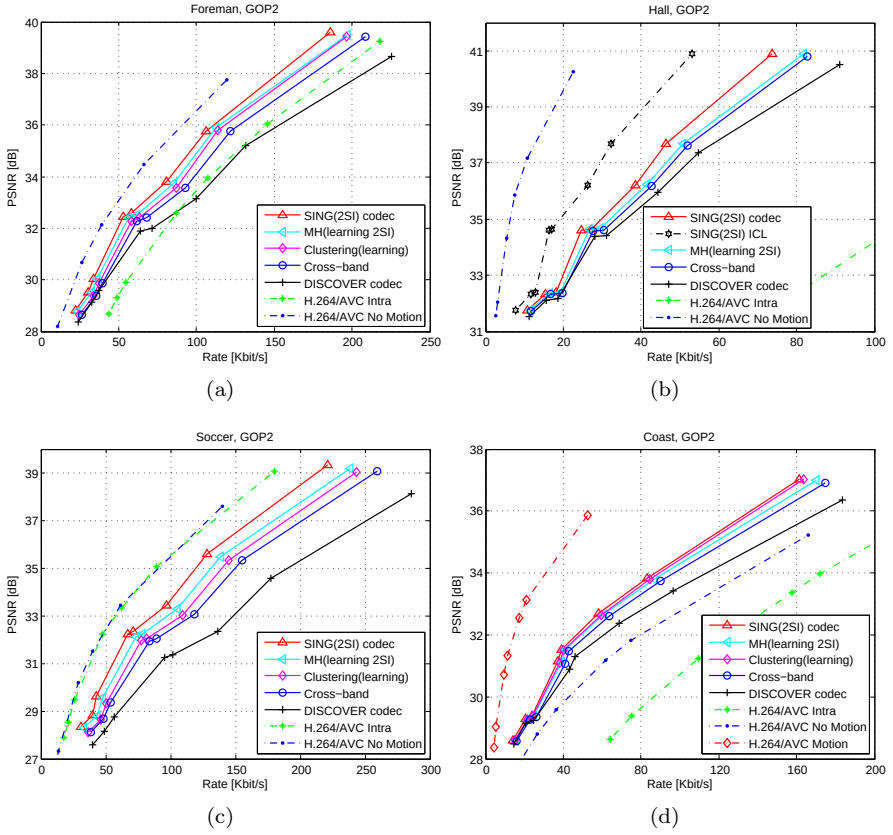


Figure 4.5: PSNR vs. rate for the proposed SING(2SI) codec for WZ frames (QCIF, 15Hz, GOP2).

tion [30] $u_{3q} = \{0; 0; 0; 1/2; 1/2; 1/3\}$. For $H = 3$, these parameters provide a uniform weighting of one, two, or three candidates. The proposed Clustering(learning) scheme (Section 4.2) uses a window size of $W = 6$ of previously decoded residual frames and a maximum number of clusters $M = 10$ (4.7), which is large enough to utilize the meaningful past information and adapt to an efficient number of noise distributions.

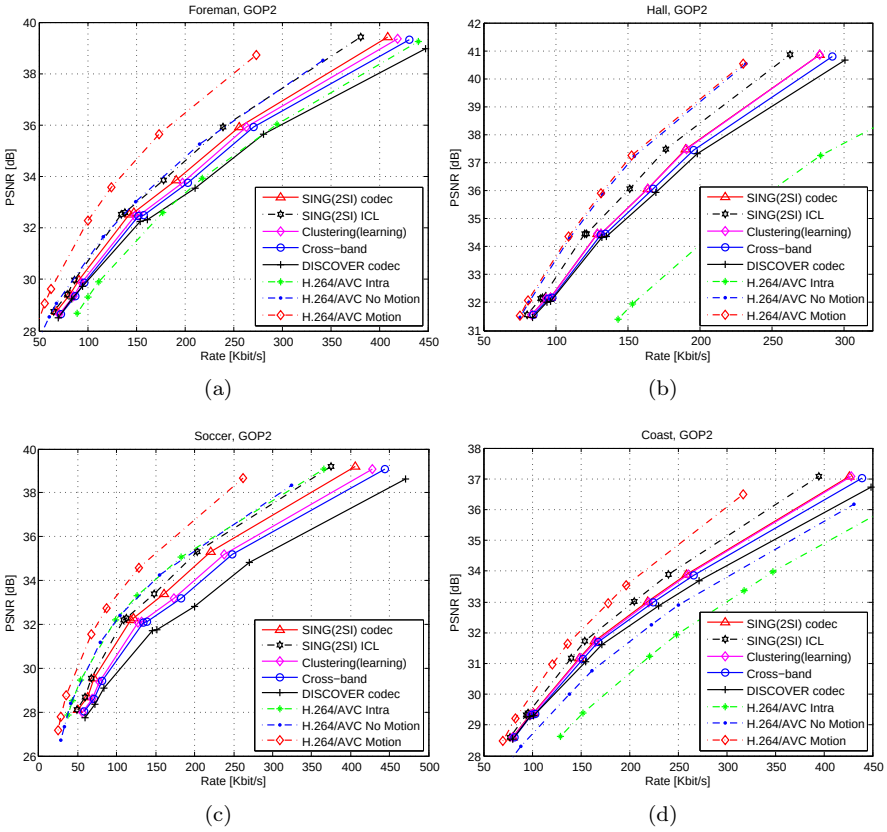


Figure 4.6: PSNR vs. rate for the proposed SING(2SI) codec for all frames (QCIF, 15Hz, GOP2).

Table 4.2: Bjøntegaard Relative Bit-rate Savings (%) over DISCOVER for WZ Frames (QCIF, 15Hz, GOP2)

Sequence	Cross-band [10]	Clustering	Clustering (learning)	MH (2SI)	MH (learning 2SI)	MH (learning 3SI)	SING (2SI)	SING (3SI)
Foreman	14.0	17.7	21.6	27.0	27.2	32.6	35.1	40.1
Hall	8.3	14.3	21.0	13.3	12.2	13.3	21.6	19.5
Soccer	26.0	30.8	34.5	41.2	46.0	49.2	61.1	62.5
Coast	11.6	17.5	21.1	17.4	17.9	19.9	24.9	25.8
Average	15.0	20.1	24.6	24.7	25.8	28.7	35.7	37.0

Table 4.3: Bjøntegaard PSNR Improvement (dB) over DISCOVER for WZ Frames (QCIF, 15Hz, GOP2)

Sequence	Cross-band [10]	Clustering	Clustering (learning)	MH (2SI)	MH (learning 2SI)	MH (learning 3SI)	SING (2SI)	SING (3SI)
Foreman	0.633	0.798	0.974	1.177	1.181	1.398	1.492	1.659
Hall	0.370	0.633	0.903	0.575	0.531	0.581	0.919	0.846
Soccer	1.305	1.521	1.677	1.921	2.088	2.216	2.649	2.690
Coast	0.352	0.530	0.637	0.526	0.540	0.600	0.741	0.762
Average	0.665	0.872	1.047	1.050	1.085	1.199	1.450	1.489

Table 4.4: Bjøntegaard Relative Bit-rate Savings (%) over DISCOVER for All Frames (QCIF, 15Hz, GOP2)

Sequence	Cross-band [10]	Clustering	Clustering (learning)	MH (2SI)	MH (learning 2SI)	MH (learning 3SI)	SING (2SI)	SING (3SI)
Foreman	6.0	7.5	9.0	11.0	11.0	13.0	13.8	15.6
Hall	2.6	3.9	5.4	3.8	3.6	3.8	5.5	4.8
Soccer	14.4	17.2	19.4	22.6	25.1	26.6	32.6	33.2
Coast	3.9	5.6	6.4	5.5	5.7	6.2	7.4	7.6
Average	6.7	8.6	10.0	10.7	11.3	12.4	14.8	15.3

Table 4.5: Bjøntegaard PSNR Improvement (dB) over DISCOVER for All Frames (QCIF, 15Hz, GOP2)

Sequence	Cross-band [10]	Clustering	Clustering (learning)	MH (2SI)	MH (learning 2SI)	MH (learning 3SI)	SING (2SI)	SING (3SI)
Foreman	0.335	0.417	0.502	0.606	0.609	0.717	0.762	0.845
Hall	0.187	0.290	0.396	0.276	0.260	0.275	0.400	0.354
Soccer	0.723	0.852	0.950	1.087	1.186	1.255	1.501	1.525
Coast	0.186	0.265	0.306	0.261	0.268	0.296	0.354	0.363
Average	0.358	0.456	0.538	0.558	0.581	0.636	0.754	0.772

Tables 4.2-4.5 report RD performance of the combined schemes (Section 4.3.2) SING(2SI) using OBMC and OF(learning) as well as SING(3SI), which additionally uses side information generation based on extrapolation [30]. Tables 4.2-4.5 present the relative average bitrate savings and equivalently the average PSNR improvements (using the Bjøntegaard metric [35] and fitting a curve through the 8 RD points measured) over the DISCOVER codec for WZ frames and overall frames. The results are also compared to the DVC scheme in Sec. 2.2.4 [10] called Cross-band. The SING codecs are based on combining the clustering and multi-hypothesis techniques, which are also evaluated individually. The noise model in Section 4.1.2 integrated in the DVC scheme in Sec. 2.2.4 [10], is named Clustering. The noise model proposed in Section 4.2 integrated in DVC scheme in Sec. 2.2.4 [10] is named Clustering(learning). Both of these are based on the OBMC side information. The proposed multi-hypothesis TDWZ codecs combining OBMC with OF and OF(learning) techniques mentioned in Section 4.3.1 are called MH(2SI) and MH(learning 2SI), respectively. MH(learning 3SI) refers to the additional use of extrapolation, respectively. Compared to DISCOVER, the average bitrate saving for the combined scheme SING(3SI) model is overall (average Bjøntegaard) 37% and 15% better on WZ frames and all frames, respectively. The performance improvement is 62.5% and 33.2% (or equivalently the average improvement in PSNR is 2.69 dB and 1.53 dB) for WZ frames and overall frames, respectively, for the difficult *Soccer* sequence. Compared to the Cross-band DVC scheme (Sec. 2.2.4), a bit-rate saving (Bjøntegaard) of 36.5% is observed for *Soccer* on the WZ frames. Looking at Table 4.2, we see that both Clustering(learning) and MH(learning) introducing OF improve the average bit-rate savings to about 25% starting from the 15% savings of the baseline Cross-band codec Sec. 2.2.4 [10]. Further, the Clustering and MH combine well in SING(2SI) for a 36% saving. Looking at the individual sequences, we see that using OF in MH improves performance most for high motion sequences *Foreman* and especially *Soccer*, whereas Clustering(learning) achieves better results on the low motion sequences as *Coast* and especially *Hall Monitor*. Our results may be compared with a few GOP2 results in [20], [40], [22], [44]. The TRACE method [44] reports 1.6% bit-rate saving for *Foreman* (at 30Hz) compared with [39]. The following comparisons are evaluated for QCIF and 15Hz frame rate at 400 Kb/s. Compared to DISCOVER, the results in [20] show an improvement of 0.4 dB for *Foreman* and 0.7 dB for *Soccer*. Improvements of 0.5dB for *Foreman* and 0.1dB degradation for *Soccer* are reported [40]. More recently, the scheme in [22] shows an improvement 0.4 dB for *Foreman* and 0.5 dB for *Soccer*. At 400Kb/s, improvements compared with DISCOVER of 1.0dB for *Foreman* (Fig. 6.5(a)) and 1.4dB for *Soccer* (Fig. 6.5(d)) are achieved by Clustering (learning). Specifically, the improvements of MH(learning 2SI) including OF(learning) are robust for the high

motion sequences as *Soccer*. The proposed SING(2SI) gains considerable improvements on the more complex motion sequences such as *Soccer* with 61.1% and *Foreman* with 35.1% bitrate savings for WZ frames. The improvements are also robust ranging from the complex sequences, e.g. *Soccer*, to the simple motion sequences, e.g. *Hall Monitor*. As a special case, the performance of SING(3SI) for *Hall Monitor* is slightly worse than SING(2SI) as shown in Tables 4.2-4.5. Looking at both rate and distortion results, the bit rate is, as expected, lower for SING(3SI) than for SING(2SI), but the problem is that the PSNR of SING(3SI) is also lower than that of SING(2SI). In general, the RD performances of all methods proposed are robustly better than using the noise model in Sec. 2.2.4 [10], as well as DISCOVER. It may be noted that the encoding and thereby also encoding complexity are the same in all cases.

The RD performance of the SING(2SI) codec and H.264/AVC coding is also depicted in Figs. 4.5-4.6 for WZ frames and all frames, respectively. The SING(2SI) codec gives a better RD performance than H.264/AVC Intra coding for *Foreman*, *Hall Monitor*, and *Coastguard*, and also better than H.264/AVC No Motion for *Coastguard*. The RD performance of the SING(2SI) codec clearly outperforms those of Sec. 2.2.4 [10] and DISCOVER. For medium to high rates the improvement for *Soccer* is up to 4dB for WZ frames. The ICL (4.12) measures the quality of the side information of the coded coefficients. The SING(2SI) ICL result (Fig. 4.6) actually matches those of H.264/AVC No Motion for *Foreman* and *Soccer*. For *Hall Monitor* SING(2SI) ICL is close to H.264/AVC Motion. This illustrates that if more efficient Slepian-Wolf coding is developed, the performance gap between practical Wyner-Ziv video coding and the conventional predictive video coding would be further reduced.

We have tested the proposed scheme SING(2SI) on four test sequences (299 frames, QCIF at 30Hz of) *Foreman*, *Soccer*, *Hall Monitor*, and *Coastguard* using a GOP size 4. The two key frames are again coded using H.264/AVC Intra. Thereafter GOP4 follows the hierarchical decoding order, where the middle frame is first decoded based on the two decoded key frames and then the two remaining frames are decoded based on the nearest decoded key frame and the decoded middle frame. RD points are calculated for the four 4×4 quantization matrices Q1, Q4, Q7, and Q8 [9]. The RD performance of the SING(2SI) codec in Figs. 4.7-4.8 is better than those obtained by the Cross-band codec [10] and DISCOVER. The SING(2SI) codec gives a better RD performance than H.264/AVC Intra and also better than H.264/AVC No Motion for *Foreman* and *Coastguard*. In particular, the SING(2SI) codec performance matches that of H.264/AVC No Motion for the high motion sequence *Soccer*. Compared to DISCOVER, the average Bjøntegaard bitrate saving is 37.5% and 23% (or

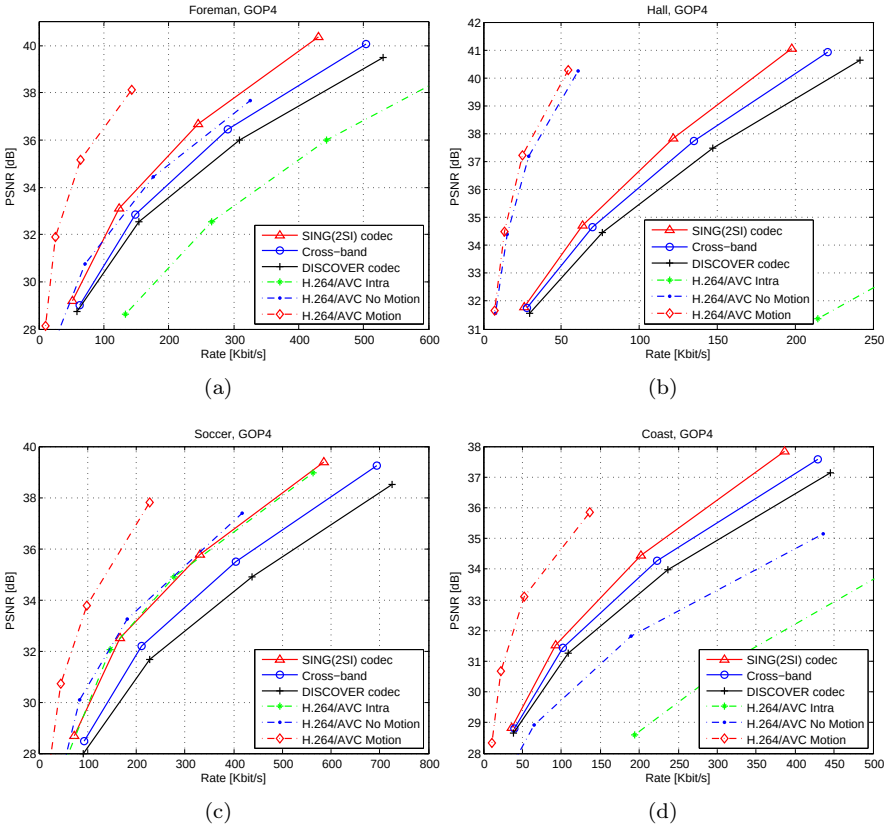


Figure 4.7: PSNR vs. rate for the proposed SING(2SI) codec for WZ frames (QCIF, 30Hz, GOP4).

equivalently the average PSNR improvement is 1.5 dB and 1.1 dB) for WZ frames and all frames, respectively. For the difficult sequence *Soccer*, the bitrate saving is 54.4% (or equivalently the improvement in PSNR is 2.2 dB) for WZ frames. The results may be compared with the GOP4 results in [20] at 400 Kb/s. Compared to DISCOVER, the results in [20] show an improvement of 1.0 dB for *Foreman* (QCIF, 15Hz) and 0.9 dB for *Soccer* (QCIF, 15Hz). In comparison, an improvement of 1.6 dB for *Foreman* and 1.9 dB for *Soccer* are achieved by the SING(2SI) codec as seen in Figs. 4.8(a) and 4.8(c).

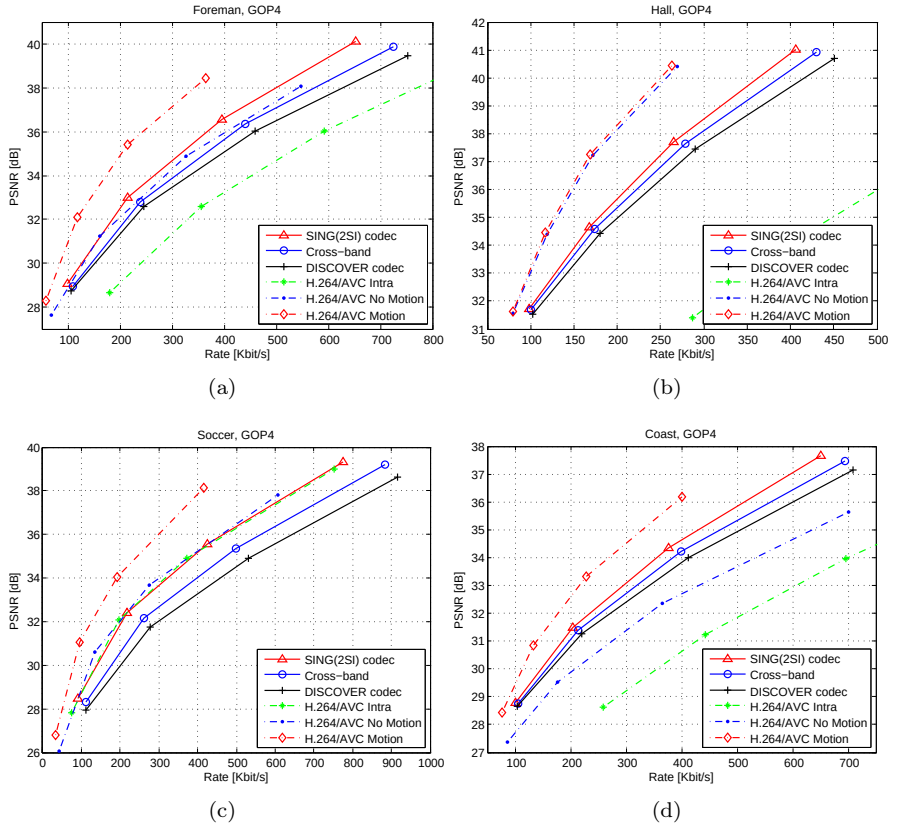


Figure 4.8: PSNR vs. rate for the proposed SING(2SI) codec for all frames (QCIF, 30Hz, GOP4).

4.5 Summary

In this chapter, TDWZ video coding was improved using optical flow and clustering of DCT blocks. Optical flow was used for frame interpolation generating side information, which was adopted in a multi-hypothesis scheme to compensate weaknesses of block based methods. Adaptive noise modeling using clustering was introduced additionally utilizing residues of previously decoded frames and generating a number of noise residual distributions within a frame for adaptive optimization of the soft side information during decoding. Moreover,

the adaptive noise model refined the residue to take advantage of correlation of DCT coefficients between neighboring blocks. Experimental results show that the coding efficiency of the proposed SING scheme which combines all the techniques can significantly improve the RD performance of TDWZ video compared to DISCOVER as well as the cross-band TDWZ scheme in Sec. 2.2.4 [10] without changing the encoder. For a GOP size of 2 the average bitrate saving of the SING(3SI) codec is 37% (or equivalent the average improvement in PSNR is 1.5 dB) on WZ frames compared with the DISCOVER codec.

Chapter 5

Motion and Reconstruction Reestimation

This chapter proposes a motion and reconstruction reestimation technique to improve side information and noise residue frames by taking partially decoded information into account. To improve noise modeling, a noise residual motion reestimation technique is proposed by using residual motion compensation with motion updating to estimate a current residue based on previously decoded frames and correlation between estimated side information frames. In addition, the chapter proposes a generalized reconstruction algorithm to optimize a multi-hypothesis reconstruction. The proposed techniques using the motion and reconstruction reestimation (MORE) are integrated in the SING TDWZ (Chapter 4) to create a MORE codec which significantly improves the TDWZ coding efficiency.

The efficiency of DVC coding mainly depends on the SI and the noise residue, which is the correlation between the source and the SI. Partially decoded information has played a valuable role in improving the SI and the residue during decoding. Partially reconstructed frames have been utilized to update motion fields to obtain better SI and residue qualities [21, 45, 46]. SI refinement was proposed in [45] to refine the SI after decoding all DCT bands in order to improve reconstruction. To extend this approach, SI and residues were refined in [46] using motion updating after decoding each DCT band. Later, a learning based decoding approach was proposed in [21] based on using overlapped motion vectors for updating the motion field to achieve a better SI quality and a more accurate correlation. It was shown in Chapter 4 that optical flow based SI generation [8, 42] compensates the weaknesses of block-based methods very well. With the newest development [16], optical flow based SI generation consistently outperforms block based SI generation in terms of quality.

Noise estimation is one important aspect influencing the coding performance.

The decoder needs to estimate the correlation between source and SI. In Chapter 4 [8], a noise learning technique was proposed to utilize the residues of previously decoded frames. Although a number of noise residual distributions were generated to optimize the soft SI during decoding, the more accurate correlation between the previously decoded residue and the current residue has not been exploited. This correlation may be expressed by motion vectors between the previous residue and the current one. Therefore, in this chapter, a residual motion compensation is proposed to generate a more accurate estimate of correlation noise by exploiting information from previously decoded frames as well as the correlation between the previous and current estimated SI frames. The proposed techniques are combined based on the SING DVC codec (Chapter 4 [8]) to improve the RD performance of the TDWZ scheme.

In order to enhance the RD performance, a multiple-input TDWZ decoder [8, 30, 36] uses multiple versions of soft SI, which can be generated by applying different SI generation methods, e.g. block based and optical flow based SI [8]. Previous decodings and reconstructions were based on the average of two hypotheses [30, 36] or the predefined weighted multiple soft-inputs for decoding and a single selected SI for reconstruction [30]. Meanwhile, the multiple soft-inputs in [8] were utilized both for decoding and reconstruction based on a predefined weighted joint distribution. Predefined weighting parameters may not be optimal for the multi-hypothesis reconstruction for all bitplanes, bands, or frames. This chapter proposes a generalized reconstruction algorithm to adaptively optimize the weighting parameters to iteratively improve the multiple-hypothesis reconstruction during the decoding.

The rest of this chapter is organized as follows. In Section 5.1, the architecture of the considered TDWZ video codec in Chapter 4 [8] is presented, which we take as starting point. In Section 5.2, a residual motion compensation and a generalized reconstruction following a block based motion reestimation are proposed. A new TDWZ video scheme is presented in Section 5.3, based on the basic TDWZ [8] with the motion and reconstruction reestimation techniques. Finally, Section 6.3 presents simulation results, analyzes the contributions of the different techniques, and compares the performance with reference methods.

5.1 Side Information and Noise Learning (SING) DVC

The SING scheme was introduced in Chapter 4 [8] by using optical flow to improve the SI generation and clustering to improve noise modeling. The optical

flow based SI generation can compensate for weaknesses of an overlapping block motion compensation (OBMC) [10]. Clustering is used to exploit cross band correlation as well as noise modeling adaptivity. In addition, the SING scheme improves the SI and noise modeling by learning from previously decoded WZ frames.

The basic elements of the SING codec are depicted in Fig. 4.4. They consist of OBMC and OF based side information generations without learning, a noise model using the residual learning [8], the soft-input estimation, the reconstruction using side information and noise learning [8] and multiple input LDPCA decoding. First, the side information generations calculate the noise residual frames NR_{01}, NR_{02} and the side information frames, SI_{01}, SI_{02}, using the OBMC [10] and OF [16], where SI_{01}, NR_{01} and SI_{02}, NR_{02} are generated by the OBMC based and OF [16] based Side Information Generations, respectively. These are transformed and input to the noise models. Noise parameters α_{hRL} and parameters α_{hCB} are calculated using the SING Noise Model. The soft-inputs Pr_{1q}, Pr_{2q}, and Pr_{1RL}, Pr_{2RL} are calculated by the Soft Input Estimation with SI and Noise Learning. All soft-inputs are fed into the multiple input LDPCA decoder [8]. After decoding, the selected candidate and the corresponding weighted joint distribution of multi-hypothesis is chosen for reconstruction.

At the SING decoder, for refining SI and noise distribution during decoding, the SING scheme reconstructs partially decoded WZ frames after each bitplane and band decoded. The partially reconstructed WZ frames are obtained by using already decoded DCT bands which are reconstructed based on decoded DCT intervals and undecoded DCT bands directly provided by the corresponding SI DCT bands. The more efficiently the partially decoded information is utilized, the more improvement we get. Therefore, the relation between the reconstructed WZ frame in pixel domain and the decoded DCT intervals that are determined by successfully decoded bitplanes at the decoder side shall be considered to drive our motion and reconstruction reestimation in the following sections.

5.2 Noise Residual Motion Reestimation and Generalized Reconstruction

In order to further improve the RD performance of the TDWZ scheme, this section takes noise residual motion reestimation and optimal reconstruction into account to enhance not only noise modeling but also the optimal reconstruction process. Firstly, the correlation information between a current residue

and previously decoded residues is exploited by a residual motion compensation technique to generate a more accurate noise distribution. Thereafter, a generalized reconstruction is used to adaptively optimize the weighting parameters for the multi-hypothesis reconstruction. Furthermore, after decoding each bitplane, the SI and residue frames are updated by a block based motion reestimation to reestimate motion vectors to regenerate a better SI and residue. These techniques are iteratively carried out during decoding until all bitplanes are successfully decoded.

5.2.1 Residual Motion Compensation

To improve the noise modeling in terms of the noise residue, this chapter proposes a technique exploiting information from previously decoded frames in terms of the correlation between the previous and current residual frames. This correlation can be expressed by the motion between the previous residue and the current residue, which may be considered the same as the motion between the previous SI and the current SI. This technique generates residual frames by compensating the motion between the previous SI frames and the current SI frame to the current residual frame to generate a more accurate noise distribution for noise modeling.

For a GOP of size two, let $\widehat{X}_{2n-2\omega}$ and $\widehat{X}_{2n}$ denote two decoded WZ frames at time $2n-2\omega$ and $2n$, where ω denotes the previously decoded ω^{th} WZ frame before the current WZ frame $2n$. Their associated SI frames are denoted by $Y_{2n-2\omega}$ and Y_{2n}, respectively. The side information generation may give similar quality estimates for the same objects that appear on the previous and current WZ frames. Here, each frame is split into N non-overlapped 8×8 blocks indexed by k, where $1 \leqslant k \leqslant N$. It makes sense to assume that the motion vector v_k of block k at position z_k between $\widehat{X}_{2n-2\omega}$ and $\widehat{X}_{2n}$ is the same as between $Y_{2n-2\omega}$ and Y_{2n}. This is represented as follows,

$$Y_{2n}(z_k) \approx Y_{2n-2\omega}(z_k + v_k). \tag{5.1}$$

A motion compensated estimate of $\widehat{X}_{2n}$ based on the motion v_k, $\widehat{X}_{2n}^{MC}$, can be obtained by

$$\widehat{X}_{2n}^{MC}(z_k) = \widehat{X}_{2n-2\omega}(z_k + v_k), \tag{5.2}$$

Based on the estimated SI frames $Y_{2n-2\omega}$ and Y_{2n}, the vectors v_k are calculated using (5.1) within a search range Φ as

$$v_k = \arg\min_{v \in \Phi} \sum\nolimits_{\text{block}} (Y_{2n}(z_k) - Y_{2n-2\omega}(z_k + v))^2, \tag{5.3}$$

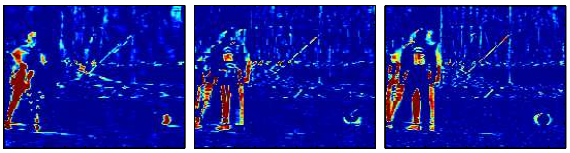
where $\sum_{\text{block}}$ is the sum over all pixel positions z_k. Thereafter, $\widehat{X}_{2n}^{MC}$ is estimated by compensating $\widehat{X}_{2n-2\omega}$ (5.2) for the selected motion v (5.3). Let R_{2n} denote the current residue at time $2n$, generated by OBMC or OF, and let $\widehat{R}_{2n}^{MC}$ denote the motion compensated residue, where $\widehat{R}_{2n}^{MC}$ can be estimated from $\widehat{X}_{2n}^{MC}$ and Y_{2n} as follows

$$\widehat{R}_{2n}^{MC}(z_k) = \widehat{X}_{2n}^{MC}(z_k) - Y_{2n}(z_k). \tag{5.4}$$

Finally, the compensated residue is obtained by inserting (5.2) in (5.4)

$$\widehat{R}_{2n}^{MC}(z_k) = \widehat{X}_{2n-2\omega}(z_k + v_k) - Y_{2n}(z_k). \tag{5.5}$$

Figure 5.1 provides an example for frame 18 of *Soccer* using the residual motion compensation (RMC) technique, where a motion compensated residue $\widehat{R}_{18}^{MC}$ (5.5) is predicted based on the decoded frame $\widehat{X}_{16}$ and the motion v between the SI frames Y_{18} and Y_{16}. The RMC residue in Fig. 5.1(b) shows a higher correlation with the ideal residue, calculated by $X_{18}-Y_{18}$ (Fig. 5.1(c)), than the OBMC residue [10] (Fig. 5.1(a)). Figure 5.2 depicts the frame by frame PNSR performance for *Soccer* for the residue using OBMC and the residue using the residual motion compensation (5.5), denoted as OBMC and RMC, compared with the ideal residue. The residue using motion compensation consistently outperforms the residue using OBMC.



(a) OBMC residue (b) RMC residue (c) Ideal residue

Figure 5.1: Residual motion compensation for *Soccer* frame 18.

Table 5.1 shows PSNR comparisons of the ideal residue with the original residue for both the OBMC technique and the RMC technique. On all these test sequences, the RMC quality outperforms the OBMC quality.

5.2.2 Generalized Reconstruction

In this section, we propose a method for optimizing the quality of reconstructed frames, by means of iteratively refining a multi-hypothesis reconstruction. The

Table 5.1: The Average PSNR [dB] Results for Quality of Residue Using OBMC and The Residual Motion Compensation Compared with The Ideal Residue (GOP2)

Sequence	OBMC	RMC
Foreman, QP=25	25.08	25.71
Hall, QP=24	30.83	32.68
Soccer, QP=25	17.53	19.23
Coast, QP=26	26.57	27.66

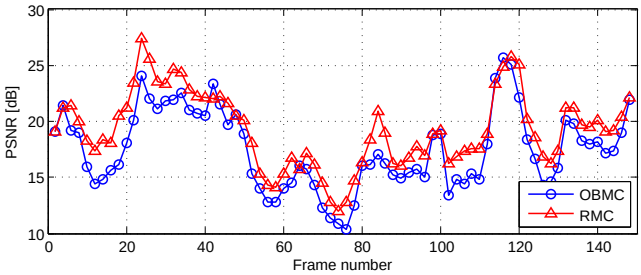


Figure 5.2: PSNR calculated between the ideal residue and residues using OBMC and RMC (using residual motion compensation), respectively, for the *Soccer* sequence (key frames QP=26)

weighting parameters are optimized after each decoded bitplane by an optimization algorithm. In principle, the optimization algorithm can be applied to any number of generated SI frames. In this work, SI frames using the OF based and OBMC based techniques are used.

Let us consider H side information frames Y_h, $h \in [1, H]$, at the decoder side for each original WZ frame X. Let $\widehat{X}$ denote the reconstructed frame. For each band, let N denote the total number of coefficients, indexed by k, i.e. $1 \leqslant k \leqslant N$, and let $[\mathbb{L}_k, \mathbb{U}_k)$ denote the decoded quantization interval for coefficient index k. The reconstructed values of coefficient index k denoted as $\widehat{x}_k$ will be optimized based on a minimum mean-squared error (MSE) estimate of source $x_k \in X$ given $[\mathbb{L}_k, \mathbb{U}_k)$ and Y_h, expressed by

$$\widehat{x}_k = \arg\min_{\widehat{x}_k} \mathrm{E}[(\widehat{x}_k - x_k)^2 | x_k \in [\mathbb{L}_k, \mathbb{U}_k), y_{1k}, ..., y_{Hk}], \tag{5.6}$$

where E denotes the expectation operator. The residue between the source X

and the SI Y_h is modeled by a Laplacian distribution $f_{X|Y_h}(x_k) = \frac{\alpha_{hk}}{2} e^{-\alpha_{hk}|x_k - y_{hk}|}$, where α_{hk} is a Laplacian parameter of the estimate between the coefficient x_k of source X and the coefficient y_{hk} of SI Y_h.

We introduce a generalized objective function for the estimate of x_k based on minimizing over the N coefficients in a given band conditioning on the given H side information Y_h, the corresponding Laplacian distributions, and decoded quantization intervals as follows

$$J_m(\widehat{X}, U) = \sum_{h=1}^{H} u_h^m \sum_{k=1}^{N} \int_{\mathbb{L}_k}^{\mathbb{U}_k} (\widehat{x}_k - x_k)^2 f_{X|Y_h}(x_k) \mathrm{d}x_k + \eta\Big(1 - \sum_{h=1}^{H} u_h\Big), \quad (5.7)$$

where $U = \{u_h\}$ denotes the sets of weights on the SI frames, i.e. $u_h \geqslant 0$ denotes the weight for the SI Y_h, where we impose the constraint $\sum_{h=1}^{H} u_h = 1$. The parameter η represents a weight on the constraint term that the sum of u_h equals one and m ($m \geqslant 1$) is a constant. The generalized objective function is optimized by selecting the variables, u_h and $\widehat{x}_k$. We shall minimize $J_m(\widehat{X}, U)$ by an iterative process, in turn adjusting u_h and $\widehat{x}_k$. The degree m will influence how the weighting parameters u_h are optimized. If $m > 1$, the weighting parameters u_h are determined by an iterative minimization, whereas for $m = 1$, the weighting parameters u_h are decided directly.

The minimum solution is achieved by minimizing the objective function J_m (5.7). The minimum is obtained when the gradient is zero, i.e. the partial derivatives of J_m of $\widehat{x}_k$ and u_h are zero. The derivatives are given by

$$\frac{\partial J_m}{\partial \widehat{x}_k} = \sum_{h=1}^{H} u_h^m \int_{\mathbb{L}_k}^{\mathbb{U}_k} 2(\widehat{x}_k - x_k) f_{X|Y_h}(x_k) \mathrm{d}x_k. \quad (5.8)$$

$$\frac{\partial J_m}{\partial u_h} = m u_h^{m-1} \sum_{k=1}^{N} \int_{\mathbb{L}_k}^{\mathbb{U}_k} (\widehat{x}_k - x_k)^2 f_{X|Y_h}(x_k) \mathrm{d}x_k - \eta, \quad (5.9)$$

Setting the partial derivatives of (5.8) and (5.9) to zero, and using the con-

straint $\sum_{h=1}^{H} u_h = 1$, we obtain:

$$\widehat{x}_k = \frac{\sum_{h=1}^{H} u_h^m \int_{\mathbb{L}_k}^{\mathbb{U}_k} x_k f_{X|Y_h}(x_k)\mathrm{d}x_k}{\sum_{h=1}^{H} u_h^m \int_{\mathbb{L}_k}^{\mathbb{U}_k} f_{X|Y_h}(x_k)\mathrm{d}x_k}. \tag{5.10}$$

$$u_h = \frac{\left(\sum_{k=1}^{N} \int_{\mathbb{L}_k}^{\mathbb{U}_k} (\widehat{x}_k - x_k)^2 f_{X|Y_h}(x_k)\mathrm{d}x_k\right)^{1/(1-m)}}{\sum_{h=1}^{H}\left(\sum_{k=1}^{N} \int_{\mathbb{L}_k}^{\mathbb{U}_k} (\widehat{x}_k - x_k)^2 f_{X|Y_h}(x_k)\mathrm{d}x_k\right)^{1/(1-m)}}, \tag{5.11}$$

The minimization of J_m is accomplished by iteratively repeating (5.10) and (5.11). The process is terminated after iteration t when the following termination criteria is satisfied

$$\max_{1\leqslant k\leqslant N}\{|\widehat{x}_k^{(t)} - \widehat{x}_k^{(t-1)}|\} < \epsilon, \tag{5.12}$$

where $\widehat{x}_k^{(t)}$ is an updated reconstructed coefficient, $\widehat{x}_k^{(t-1)}$ is the previous reconstructed coefficient, and ϵ is the predefined termination threshold. In this work $\epsilon = 0.0001$. The algorithm is outlined in Algorithm 1. Note that J_m is convex in $\widehat{x}_k$. In particular, if $m = 1$ (5.7), the weighting parameters $\{u_h\}$ as determined by (5.11) taking the limit $m \to 1$ do not require iterations, as the resulting $\widehat{x}_k$ becomes the minimum MSE reconstruction by

$$\widehat{x}_k = \frac{\sum_{h=1}^{H} u_h \int_{\mathbb{L}_k}^{\mathbb{U}_k} x_k f_{X|Y_h}(x_k)\mathrm{d}x_k}{\sum_{h=1}^{H} u_h \int_{\mathbb{L}_k}^{\mathbb{U}_k} f_{X|Y_h}(x_k)\mathrm{d}x_k}. \tag{5.13}$$

In addition, when $m \to \infty$, u_h becomes equal to $1/H$ for all SIs. Thus, for $m > 1$, we may consider the solution as an unsupervised way to mix the uniform and the MSE solutions. In this work $m = 2$.

Input: Side information $\{Y_h\}$; Laplacian parameters $\{\alpha_{hk}\}$; decoded quantization intervals $\{[\mathbb{L}_k, \mathbb{U}_k)\}$
Output: The reconstructed coefficients $\{\widehat{x}_k\}$
Initialization $u_h^{(0)}$; $\widehat{x}_k^{(0)}$ by (5.10) ;
for $t = 1$ *to* $T_{\max}$ **do**
  `// Iterating until the maximum interation`
  Compute $\widehat{x}_k^{(t)}$ by (5.10) with $u_h^{(t-1)}$;
    `// Reconstructing with the previous weights`
  Compute $u_h^{(t)}$ by (5.11) with $\widehat{x}_k^{(t)}$;
    `// Updating the weights with the updated reconstruction`
  if (5.12) *is satisfied* **then**
    `// Checking the termination condition`
    The algorithm is terminated;
  end
end

Algorithm 1: Generalized reconstruction.

5.2.3 Block Based Motion Reestimation

This section introduces a technique to reestimate a bidirectional motion field after each bitplane is decoded. The SI and the residue are then updated using the reestimated bidirectional motion. For a GOP size of two, let $\widehat{X}_{2n}^{(l,i-1)}$, $Y_{2n}^{(l,i-1)}$, and $R_{2n}^{(l,i-1)}$ denote the partially decoded frame, the reestimated SI, and the reestimated residue, at time $2n$ after decoding band $l-1$ and bitplane $i-1$ of the band l. These frames correspond to the frames, $\widehat{X}_{2n}$, Y_{2n}, and R_{2n} (Secs. 5.2.1, 5.2.2). Each frame is split into N non-overlapping 8×8 blocks indexed by k, where $1 \leqslant k \leqslant N$. Let $v_k^{(0)}$ denote motion vectors estimated by the bi-directional motion estimation in the SI generation [10] between two backward and forward decoded key frames $\widehat{X}_{2n-1}$ and $\widehat{X}_{2n+1}$, respectively. The bi-directional motion vectors $v_k^{(0)}$ can be estimated within the search range Φ by

$$v_k^{(0)} = \arg\min_{v \in \Phi} \sum\nolimits_{\text{block}} (\widehat{X}_{2n-1}(x_k - v) - \widehat{X}_{2n+1}(x_k + v))^2. \tag{5.14}$$

Let $v_k^{(l,i-1)}$ denote the motion vector reestimated after band $l-1$ and bitplane $i-1$ of band l are successfully decoded, where the first band motion vector is assigned by the bi-directional motion vector (5.14) as $v_k^{(1,0)} = v_k^{(0)}$. Let

$Y_{2n}^{(l,i-1)}(x_k)$ and $R_{2n}^{(l,i-1)}(x_k)$ denote the reestimated SI block corresponding to the motion $v_k^{(l,i-1)}$ given by

$$Y_{2n}^{(l,i-1)}(x_k) = \frac{1}{2}(\widehat{X}_{2n-1}(x_k - v_k^{(l,i-1)}) + \widehat{X}_{2n+1}(x_k + v_k^{(l,i-1)})), \tag{5.15}$$

$$R_{2n}^{(l,i-1)}(x_k) = (\widehat{X}_{2n-1}(x_k - v_k^{(l,i-1)}) - \widehat{X}_{2n+1}(x_k + v_k^{(l,i-1)})). \tag{5.16}$$

Here we reestimate the motion vectors for all blocks after decoding each bitplane, by searching for the best match in the search range Φ between the partially decoded block $\widehat{X}_{2n}^{(l,i-1)}(x_k)$ and $Y_{2n}^{(l,i-1)}(x_k)$ as:

$$v_k^{(l,i)} = v_k^{(l,i-1)} + \arg\min_{v\in\Phi} \sum\nolimits_{\text{block}} (\widehat{X}_{2n}^{(l,i-1)}(x_k) - Y_{2n}^{(l,i-1)}(x_k + v))^2. \tag{5.17}$$

The updated motion vectors $v_k^{(l,i)}$ obtained are used in the OBMC based frame interpolation [10], where they are subjected to the processes of motion smoothing, variable block size refinement, and adaptive weighted OBMC [10]. Table 5.2 shows PSNR comparisons of the original OBMC SI quality denoted OBMC and the iteratively updated SIs, after decoding the DC coefficient and after decoding all AC coefficients, denoted by SI(DC) and SI(AC), respectively, using the DVC scheme in [10]. In general, the SI(AC) and SI(DC) quality outperform the SI, especially on higher motion sequences such as *Soccer* and *Foreman*. Moreover, SI(AC) is better than SI(DC) due to the iterative improvement on each decoded bitplane and band. Consequently, updates of SI $Y_{2n}^{(l,i)}$ and noise residue $R_{2n}^{(l,i)}$ are obtained to be used for decoding the next bitplane $i+1$ of band l. They are also further used to iteratively compensate the residual motion in Section 5.2.1 and optimize the reconstruction process in Section 5.2.2.

Table 5.2: The Average PSNR [dB] Results for SI Quality using OBMC and the Motion Reestimation, SI(DC) and SI(AC) (GOP2)

Sequence	OBMC	SI(DC)	SI(AC)
Foreman, QP=25	29.26	29.98	30.30
Hall, QP=24	36.46	36.37	36.54
Soccer, QP=25	21.30	23.22	23.64
Coast, QP=26	31.83	31.85	32.06

5.3 TDWZ Using Motion and Reconstruction Reestimation

5.3.1 TDWZ Using Reestimation

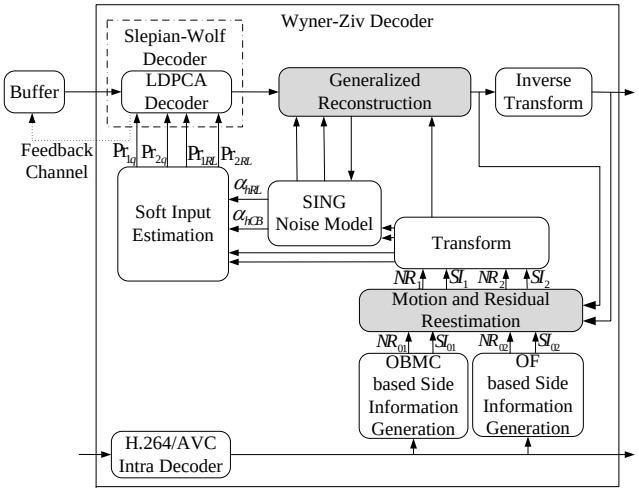


Figure 5.3: TDWZ with the motion and reconstruction reestimation (MORE 2SI).

The proposed TDWZ codec using **MO**tion and **RE**construction reestimation (MORE) is illustrated in Fig. 5.3. The new techniques, specifically the optical flow based motion reestimation (in [47]), the noise residual motion reestimation (Sec. 5.2.1), and generalized reconstruction (Sec. 5.2.2) are integrated into the SING codec (Sec. 5.1) [8]. Compared with the SING codec (Chapter 4), the MORE codec introduces two novel modules, a motion and residual reestimation and a generalized reconstruction. In addition, the MORE codec replaces the learning based OF SI generation of SING by the OF SI generation presented in [47]. The motion and residual reestimation consists of both the optical flow motion reestimation in [47] and the noise residual motion reestimation in Secs. 5.2.1 and 5.2.3. Input noise residual frames, NR_{01}, NR_{02} and SI frames, SI_{01}, SI_{02} are generated by the OBMC based SI generation [8] and the OF based SI generation (in [47]), respectively. The motion and residual reestimation module recalculates the input residual and SI frames by reestimating motion vectors as in (5.17) for SI_{01}, NR_{01}, by the block based technique (Sec. 5.2.3) and

SI_{02}, NR_{02}, by the OF based technique (in [47]). It can be noted that the recalculation of SI_{01}, NR_{01} is performed after decoding each bitplane using (5.15)-(5.16) and that of SI_{02}, NR_{02} is performed after decoding each band based on the partially decoded information. The residual frames in the MORE scheme, NR_1 and NR_2, each consists of the current residues and the additional residues compensated by (5.5). Let W be the window size specifying the number of previously decoded frames, $\widehat{X}_{2n-2\omega}$, $1 \leqslant \omega \leqslant W$. Consequently, there are W compensated residues calculated by (5.5). This compensation process is called once each band has been decoded. The output SI frames, SI_1 (5.15), SI_2, and residual frames, NR_1 ((5.15) and (5.5)) and NR_2 (in [47] and (5.5)), of the motion and residual reestimation are generated by reestimation (in [47] and 5.2) and thereafter an optimal selection is introduced in the following section (Sec. 5.3.2).

5.3.2 Selecting Side Information

The quality of the reestimated SI varies for the decoded bands. The quality also depends on the given sequence, e.g. the quality of sequences with less motion may be degraded by the OF reestimation (in [47]). Thus, we should optimally select the best quality among the inital SI and the reestimated SI. The evaluation process is performed after each band is successfully decoded, where the given decoded bitplanes and intervals are utilized for selecting the SI to decode the next band.

In order to evaluate the quality of the so far decoded side information, we also calculate an Ideal Code Length (ICL) as in (4.12) (Chapter 4) [10], which measures the number of bits required by applying ideal (arithmetic) coding to the given soft-input values if a (non-distributed) encoder would encode using the same soft-input values. Let $\Pr(b_{bpi})$ denote the soft-input values fed into the LDPCA decoder, where b_{bpi} denotes the i^{th} bit in bitplane bp, and are calculated by reading b_{bpi} as the bits after decoding. The ideal code length, L_{bp}, for bitplane bp is calculated as

$$L_{bp} = \sum_{i=1}^{N} -\log \Pr(b_{bpi}). \tag{5.18}$$

The ICL is obtained as the sum over all bitplanes. This is equivalent to a log-likelihood measure of the coded coefficients.

For evaluating the reestimated SI after decoding each band, we should not only

take the rate in terms of the ICL into account but also consider a distortion cost for the corresponding reestimated SI. The rate distortion cost C_l for a particular band l is expressed by:

$$C_l = \mathcal{R} + \lambda_l \mathcal{D}, \tag{5.19}$$

where the first term $\mathcal{R}$ is an estimated rate for the coded coefficients after decoding band l, and the second term consists of a Lagrange parameter λ_l, which is experimentally set to 0.015, multiplied as the mean square distortion $\mathcal{D}$. We estimate by the mean of the ICL (5.18) over all decoded bitplanes bp by

$$\mathcal{R} = \frac{1}{B_l} \sum_{bp=1}^{B_l} L_{bp}, \tag{5.20}$$

where B_l is the number of bitplanes used to code the given band l, and L_{bp} is the ICL (5.18) of bitplane bp. The distortion term is calculated over the N decoded coefficients $\widehat{x}_k$ in the band l on the given decoded interval $[\mathbb{L}_k, \mathbb{U}_k]$ and side information Y_h by

$$\mathcal{D} = \sum_{k=1}^{N} \int_{\mathbb{L}_k}^{\mathbb{U}_k} (\widehat{x}_k - x_k)^2 f_{X|Y_h}(x_k) \mathrm{d}x_k. \tag{5.21}$$

Consequently, the cost C_l is derived from (5.18), (5.20), and (5.21) as

$$C_l = \frac{1}{B_l} \sum_{bp=1}^{B_l} \sum_{i=1}^{N} -\log \Pr(b_{bpi}) + \lambda_l \sum_{k=1}^{N} \int_{\mathbb{L}_k}^{\mathbb{U}_k} (\widehat{x}_k - x_k)^2 f_{X|Y_h}(x_k) \mathrm{d}x_k. \tag{5.22}$$

The cost C_l (5.22) is used to determine the quality of the reestimated SI after decoding each band l. When band l is successfully decoded, i.e. B_l bitplanes and the intervals $[\mathbb{L}_k, \mathbb{U}_k]$ are given at the decoder side, the cost C_l is calculated by (5.22) for each SI Y_h. Here we calculate the cost C_l for SI_{01} and SI_1 to select the better SI as the one with the smaller cost value C_l. The selection procedure is carried out similarly for SI_{02} and SI_2.

Based on the cost C_l, the output reestimated SI frames, SI_1, SI_2, and residual frames, NR_1, NR_2, are optimally selected using (5.22) as either the initial SI_{01}, SI_{02}, NR_{01}, NR_{02}, or the reestimated SI_1, SI_2, NR_1, NR_2. The reestimated outputs SI_1, SI_2, and residual frames, NR_1 and NR_2, are transformed and thereafter used as inputs for the SING noise model, the soft input estimation

[8], and the generalized reconstruction (Sec. 5.2.2) applied on two side information, SI_1 and SI_2, i.e. $H = 2$. When the coefficients and the frame are partially reconstructed, the inverse transform converts the results to the partially decoded frames. The partially decoded coefficients and frames are also fed back to the motion and residual motion compensation to reestimate the SI and residual frames for the next process until fully completing the decoding.

5.4 Performance Evaluation

We will evaluate the RD performance of the proposed techniques for the test sequences (149 frames of) *Foreman, Hall Monitor, Soccer,* and *Coastguard* with 15Hz frame rate and QCIF format, where only the luminance component of each frame is evaluated using GOP sizes 2 and 4. For GOP size of 2, odd frames are coded as key frames using H.264/AVC Intra and even frames are coded using Wyner-Ziv coding. Four RD points are considered corresponding to four 4×4 quantization matrices Q1, Q4, Q7, and Q8 [9]. H.264/AVC Intra is here given by the intra coding mode of the H.264/AVC reference codec JM 9.5 [43] in main profile. The parameters for H.264/AVC Intra are set as by DISCOVER [9] and QP values are set to those used for the key frames in the Wyner-Ziv video coding in the DISCOVER [9].

The proposed TDWZ codec based on the SING2SI codec (Chapter 4) [8] employing motion and reconstruction reestimation (Sec. 5.3.1) is denoted by MORE(2SI). The contributions of the different techniques are evaluated, where the corresponding TDWZ codecs proposed are based on the SING2SI codec (Chapter 4) [8] using the 3OF SI generation instead of the learning based generation of SING. SING2SI(RMC) denotes the SING2SI scheme using the residual motion compensation (Sec. 5.2.1) and SING2SI(GR) additionally employs the generalized reconstruction along with the residual motion compensation (Secs. 5.2.1+ 5.2.2). SING2SI(reOBMC) and SING2SI(reOF) denote the SING2SI schemes using the noise residual motion reestimation and generalized reconstruction (Sec. 5.2), and the optical flow motion reestimation (in [47]), respectively. The proposed codecs are also compared with the DVC schemes in Sec. 2.2.4 [10] called Cross-band and SING2SI (Chapter 4) [8].

5.4.1 Rate Distortion Results

Tables 6.1-5.6 present the relative average bitrate savings and the average PSNR improvements (using the Bjøntegaard metric [35] and curve fitting through

the 4 RD points measured) over the DISCOVER codec for WZ frames and overall frames. The results are also compared with the DVC Cross-band scheme (Sec. 2.2.4) [10] and the SING2SI codec (Chapter 4) [8]. Evaluated against DISCOVER, the average PSNR improvement of the proposed MORE(2SI) scheme is overall 2.5dB and 1.2dB (or equivalently the average bitrate saving is 64.1% and 24.3%) better on WZ frames and all frames, respectively. In particular, an average Bjøntegaard improvement of 4.2dB in PSNR (equivalent to 101.8% in bitrate saving) is achieved for the difficult *Soccer* sequence on WZ frames. For the individual techniques, the improvements of the SING2SI(reOF) including the OF based motion reestimation (in [47]) are robust for the high and complex motion sequences *Foreman* and *Soccer* with the average bitrate savings 74% and 94%, respectively. On the other hand, higher improvements of SING(reOBMC) including the block-based techniques (Sec. 5.2) are achieved on the lower motion sequences *Hall Monitor* and *Coastguard*. In general, the proposed techniques combine well in the final MORE(2SI) scheme which improves performance most for the high motion *Soccer* sequence. The RD performances of the proposed methods robustly outperform the DISCOVER, Cross-band, and SING2SI codecs.

The RD performance of the MORE(2SI) codec and H.264/AVC coding is also depicted in Figs. 5.4-6.5 for WZ frames and all frames, respectively. The MORE(2SI) codec gives a better RD performance than H.264/AVC Intra coding for all four test sequences and also better than H.264/AVC No Motion for *Foreman*, *Soccer*, and *Coastguard*. The RD performance of the MORE(2SI) codec clearly outperforms those of the SING2SI (Chapter 4) [8], the Cross-band (Sec. 2.2.4) [10], and DISCOVER. For high rates, the improvement of the MORE(2SI) codec on *Soccer* is up to 6dB over the DISCOVER codec for WZ frames. For evaluating the quality of side information, the MORE(2SI) ICL (5.18) are also depicted and these results further reduce the gap to H.264/AVC Motion. This shows that the performance gap between the TDWZ video coding and conventional predictive video coding would be further reduced if more efficient Slepian-Wolf coding is developed.

Table 5.3: Bjøntegaard Relative Bit-rate Savings (%) over DISCOVER for WZ Frames (QCIF, 15Hz, GOP2)

Sequence	Cross-band	SING2SI	SING2SI (RMC)	SING2SI (GR)	SING2SI (reOBMC)	SING2SI (reOF)	MORE (2SI)
Foreman	14.19	35.43	39.36	41.74	44.65	**74.24**	74.03
Hall	8.59	22.71	35.69	37.07	**37.08**	24.20	36.21
Soccer	26.72	62.70	68.58	71.06	74.41	94.04	**101.75**
Coast	11.61	24.98	38.14	39.42	39.97	29.84	**44.44**
Average	14.92	36.46	45.44	47.32	49.03	55.58	**64.10**

Table 5.4: Bjøntegaard PSNR Improvements (dB) over DISCOVER for WZ Frames (QCIF, 15Hz, GOP2)

Sequence	Cross-band	SING2SI	SING2SI (RMC)	SING2SI (GR)	SING2SI (reOBMC)	SING2SI (reOF)	MORE (2SI)
Foreman	0.65	1.52	1.66	1.75	1.85	**3.09**	3.00
Hall	0.39	0.99	1.43	**1.47**	1.46	1.06	1.42
Soccer	1.33	2.70	2.93	3.02	3.18	4.01	**4.19**
Coast	0.36	0.76	1.11	1.14	1.16	0.91	**1.28**
Average	0.64	1.49	1.78	1.84	1.91	2.27	**2.47**

Table 5.5: Bjøntegaard Relative Bit-rate Savings (%) over DISCOVER for all Frames (QCIF, 15Hz, GOP2)

Sequence	Cross-band	SING2SI	SING2SI (RMC)	SING2SI (GR)	SING2SI (reOBMC)	SING2SI (reOF)	MORE (2SI)
Foreman	5.98	13.63	14.94	15.79	16.74	**26.35**	26.22
Hall	2.55	5.52	7.89	**8.19**	8.18	5.98	8.05
Soccer	14.64	32.83	35.66	36.74	38.30	46.93	**50.15**
Coast	4.08	7.70	11.19	11.55	11.72	9.14	**12.90**
Average	6.25	14.92	17.42	18.07	18.74	22.10	**24.33**

Table 5.6: Bjøntegaard PSNR Improvements (dB) over DISCOVER for all Frames (QCIF, 15Hz, GOP2)

Sequence	Cross-band	SING2SI	SING2SI (RMC)	SING2SI (GR)	SING2SI (reOBMC)	SING2SI (reOF)	MORE (2SI)
Foreman	0.33	0.75	0.82	0.87	0.91	**1.45**	1.43
Hall	0.19	0.40	0.57	**0.59**	0.59	0.44	0.58
Soccer	0.73	1.51	1.63	1.67	1.75	2.14	**2.26**
Coast	0.19	0.37	0.53	0.55	0.56	0.44	**0.61**
Average	0.33	0.76	0.89	0.92	0.95	1.12	**1.22**

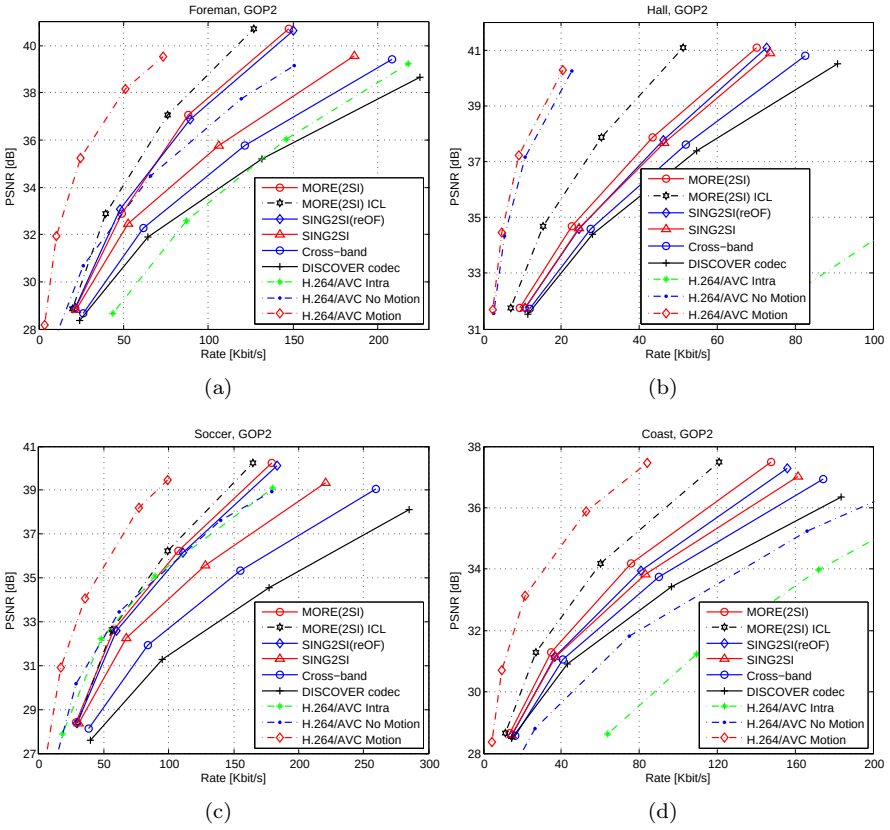


Figure 5.4: PSNR vs. rate for the proposed codec for WZ frames (QCIF, 15Hz, GOP2).

The proposed scheme MORE(2SI) is additionally tested on the four test sequences (299 frames, QCIF at 30Hz of) *Foreman*, *Soccer*, *Hall Monitor*, and *Coastguard* using GOP size 4. The two key frames are also coded using H.264/AVC Intra, thereafter the middle frame is WZ decoded based on the two decoded key frames. Finally, the two remaining frames are WZ decoded based on the nearest decoded key frame and the decoded middle frame. The RD performance of the MORE(2SI) codec in Figs. 5.6-5.7 is better than those of the Cross-band (Sec. 2.2.4) [10] and the SING2SI (Chapter 4) [8]. The MORE(2SI) codec achieves better RD performance than H.264/AVC Intra for all test sequences and outperform H.264/AVC No Motion on *Foreman*, *Soccer*, and *Coastguard*. Compared with DISCOVER, the average PSNR improvement is 2.2 dB and 1.6 dB (or equivalently the average Bjøntegaard bitrate saving is 56.3% and 34%) for WZ frames and all frames, respectively. For the diffi-

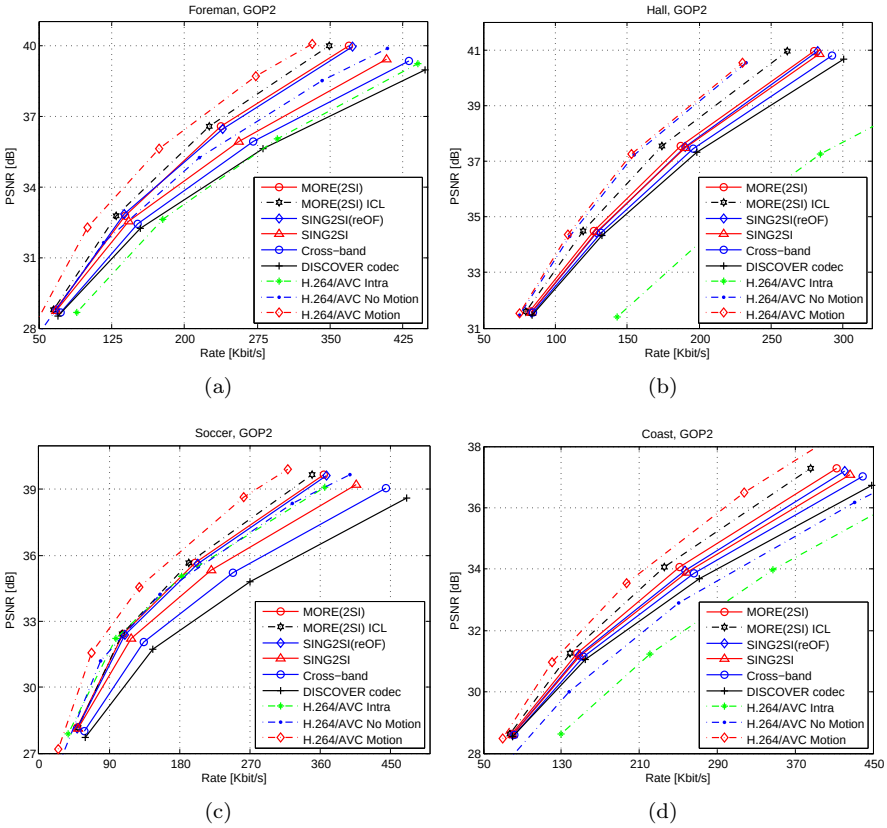


Figure 5.5: PSNR vs. rate for the proposed codec for all frames (QCIF, 15Hz, GOP2).

cult sequence Soccer, the improvement in PSNR is 3.6 dB (or equivalently the bitrate saving is 87.2%) for WZ frames.

5.4.2 Performance Comparisons

Compared with Cross-band and SING2SI (Table 6.1), bitrate savings of MORE(2SI) are 75% and 39% for *Soccer* on the WZ frames, respectively. The average bitrate savings of the MORE(2SI) are 49.2% and 27.6% over Cross-band and SING2SI. It may be noted that the encoding is the same in all cases.

Besides comparison with Cross-band (Sec. 2.2.4) [10] and SING2SI (Chapter

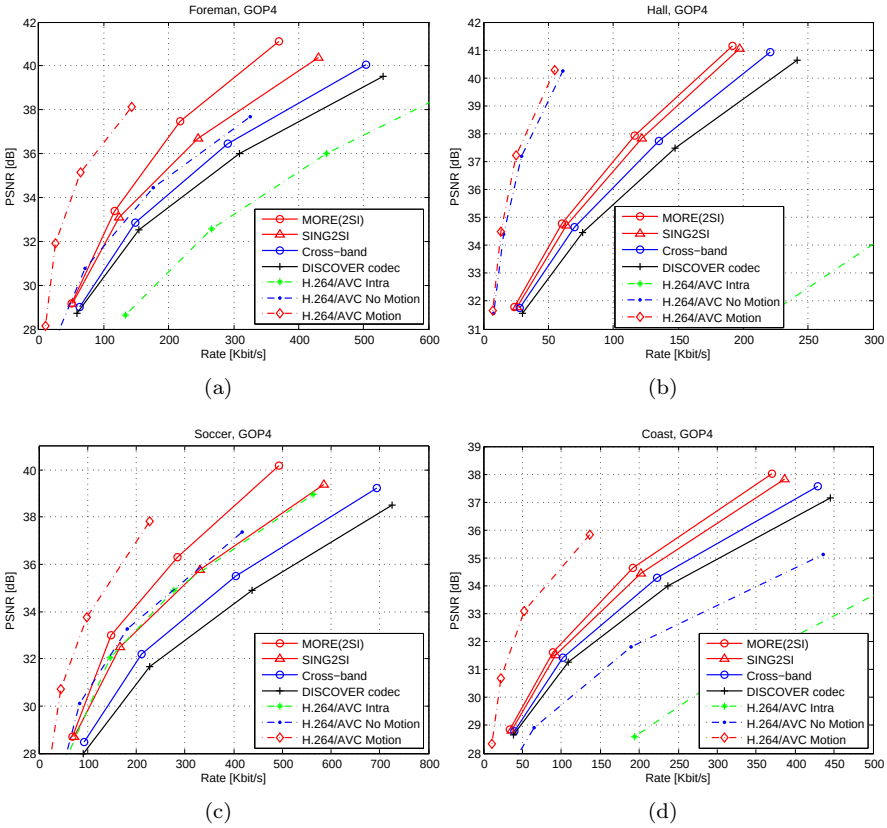


Figure 5.6: PSNR vs. rate for the proposed MORE(2SI) codec for WZ frames (QCIF, 30Hz, GOP4).

4) [8], we also compare with the results for a number of recent DVC codecs [20–23, 40, 44, 46]. The comparison will use the DISCOVER results as common reference, reporting gains over DISCOVER. This Section reports results for the high motion *Foreman* and *Soccer* as common test sequences for which comparison is feasible. The RD results may be compared with RD results of the other recent DVC codecs, some of which [20, 21, 46] utilized the partially reconstructed information to update the SI and residue during decoding. The following comparisons are evaluated for GOP2, QCIF, and 15Hz frame rate at 350 Kb/s for all frames. At 350Kb/s, the improvements of MORE(2SI) compared with DISCOVER are 2.5dB for *Foreman* (Fig. 6.5(a)) and 3dB for

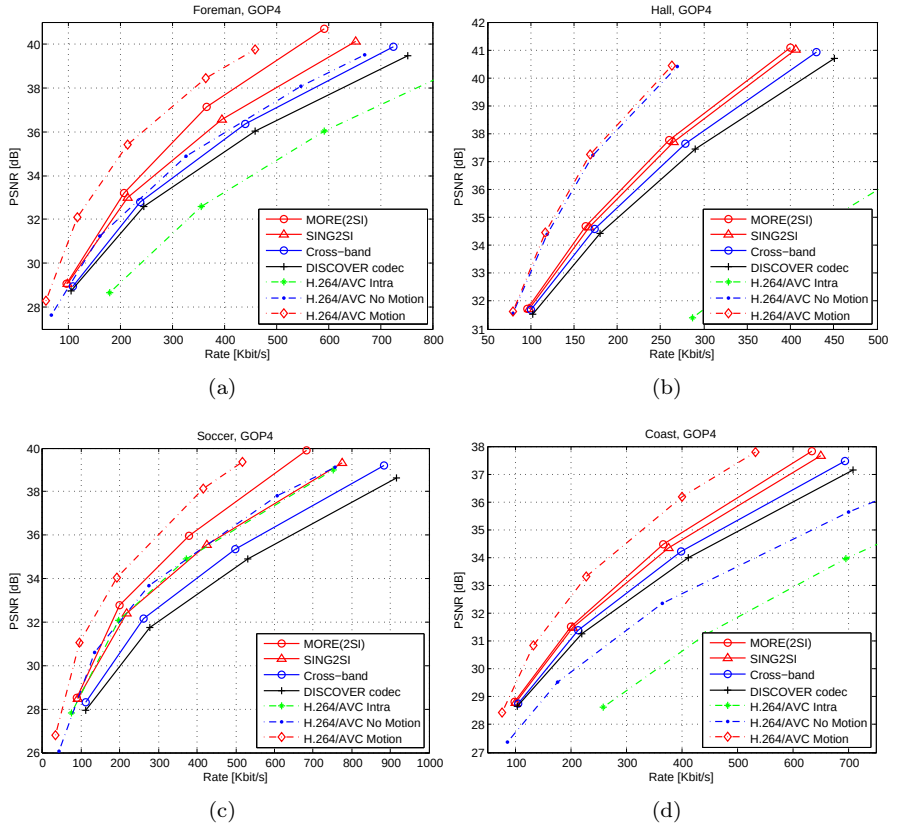


Figure 5.7: PSNR vs. rate for the proposed MORE(2SI) codec for all frames (QCIF, 30Hz, GOP4).

Soccer (Fig. 6.5(d)). Compared with DISCOVER, the results in [21] utilizing motion reestimation show an improvement of 1.8dB for *Foreman* and 1.5dB for *Soccer*. Another work employing motion reestimation [46] gains 1dB for both *Foreman* and *Soccer*. The results in [20] show an improvement of 0.4 dB for *Foreman* and 0.7 dB for *Soccer*. Improvements of 0.5dB for *Foreman* and 0.4dB degradation for *Soccer* are reported in [40]. More recently, both the scheme in [22] and the hash-based DVC codec in [23] show an improvement 0.4 dB for *Foreman* and 0.5 dB for *Soccer*. Furthermore, the TRACE method [44] reports 1.6% bit-rate saving for *Foreman* (at 30Hz) compared with [39]. Thus, the proposed MORE(2SI) codec robustly outperforms all these codecs as well

as H.264/AVC No Motion for both *Foreman* and *Soccer*. To the best of our knowledge, there are no other DVC codecs in the literature reporting better (Bjøntegaard) RD performance for both *Foreman* and *Soccer* compared with H.264/AVC No Motion.

For GOP size 4, the results are compared with the results in [8,10]. The average bitrate savings of the MORE(2SI) are 42.5% and 18.8% (or equivalently the improvement in PSNR is 1.6dB and 0.7dB) over Cross-band and SING2SI for WZ frames. For the difficult sequence Soccer, the bitrate saving is 69.5% and 32.9% (or equivalently the improvement in PSNR is 2.7dB and 1.3dB) on the WZ frames, compared with Cross-band and SING2SI, respectively.

5.5 Summary

Motion reestimation using optical flow was introduced in TDWZ DVC to take advantage of the partially decoded information. More accurate side information and residual frames were updated during decoding. Furthermore, residual motion compensation, using motion updating, generated additional residues to exploit the correlation between the previously decoded and current noise residues. Also, a generalized reconstruction algorithm was proposed to improve the multi-hypothesis reconstruction by refining weighting parameters. The proposed techniques were integrated to form the novel MORE scheme. Experimental results show that the coding efficiency of the proposed MORE scheme can robustly improve the RD performance of TDWZ DVC without changing the encoder. For a GOP size of 2, an average Bjøntegaard improvement in PSNR of 2.5dB (or equivalent an average bitrate saving of 64%) and up to 6dB improvement were achieved by the MORE(2SI) codec for WZ frames compared with the DISCOVER codec.

Chapter 6

Adaptive Mode Decision

An adaptive mode decision technique for DVC is proposed in this chapter to control and take advantage of skip mode and intra mode in DVC. The adaptive mode decision is not only based on quality of key frames but also the estimated rate of WZ frames. To improve noise distribution estimation for a more accurate mode decision, a residual motion compensation is proposed to estimate a current noise residue based on a previously decoded frame.

The DISCOVER codec [9] brought some improvements of the coding efficiency, thanks to more accurate side information generation and correlation noise modeling. Other researchers have improved upon this approach, for example, by developing advanced refinement techniques [10, 20]. The rate distortion (RD) performance of TDWZ has been improved [10] using a cross-band noise refinement technique. Despite advances in practical TDWZ video coding, the RD performance of TDWZ video coding is still not matching that of conventional video coding approaches such as H.264/AVC. Including different coding modes as in conventional video compression may be a promising solution for further improving the RD performance of DVC.

Some previous works [17, 48, 49] propose to exploit different coding modes entirely at the decoder. In [48, 49], it was proposed to skip or decide between skipping or WZ coding for coefficient bands or bitplanes. They decided the modes based on a threshold using the estimated rate and distortion. More theoretically, the work in [17] has developed techniques for rate-distortion based decoder-side mode decision. The decoder-side mode decision takes the side information position in the quantization bin into account to determine the coding modes at the coefficient and bitplane levels.

To take advantage of both the refinement technique in [10] and the decoder-side mode decision in [17], this chapter proposes an adaptive mode decision technique for TDWZ video coding. The mode decision uses estimated rate to form an adaptive mode decision and develop a residual motion compensation to generate a more accurate correlation noise. The proposed techniques are combined based on the DVC codecs the Cross-band in Sec. 2.2.4 [10] and the MORE in Chapter 5 to enhance the RD performance of the TDWZ scheme.

The rest of this chapter is organized as follows. In Section 6.1, the proposed adaptive mode decision for DVC is presented. The adaptive mode decision DVC architectures based on the Cross-band codec in Sec. 2.2.4 [10] and the MORE codec (Chapter 5) proposed are described in Section 6.2. Section 6.3 evaluates and compares the performance of our approach to other existing methods.

6.1 Adaptive Mode Decision for Distributed Video Coding

The techniques for mode decision as employed in our codec extend the method in [17]. Let X denote the original Wyner-Ziv frame and Y denote the side information frame. The cost for WZ coding a coefficient X_k with index k in a particular coefficient band is defined as [17]:

$$C_{\mathrm{WZ}}^{k} = H(Q(X_k)|Y_k = y_k) + \lambda \mathrm{E}[|X_k - \widehat{X}_k||Y_k = y_k]. \tag{6.1}$$

The first term in this sum denotes the conditional entropy of the quantized coefficient $Q(X_k)$ given the side information. The second term consists of the Lagrange parameter multiplied by the mean absolute distortion between the original coefficient X_k and its reconstruction $\widehat{X}_k$, given the side information. Entropy and distortion are calculated as in [17].

The cost for skipping the coefficient X_k is given by [17]:

$$C_{skip}^{k} = \lambda \frac{1}{\alpha}. \tag{6.2}$$

If $C_{skip}^{k} < C_{WZ}^{k}$ for all coefficients in a coefficient band, all bitplanes in the coefficient band are skipped and the side information is used as the result. In the other case, bitplane-level mode decision is performed to decide between bitplane-level skip, intra, or WZ coding as described in [17]. The coding mode for each bitplane is communicated to the encoder through the feedback channel.

One of the contributions in this chapter is to extend the method above. Instead of using a sequence-independent formula for λ as in [17], we propose to vary the Lagrange parameter depending on the sequence characteristics.

As a first step, results are generated for a range of lambdas and WZ quantization points, using the training sequences *Foreman, Coastguard, Hall Monitor,* and *Soccer* (QCIF, 15Hz, GOP2). Wherever necessary, the intra QP of the key frames is adjusted so that the qualities of WZ frames and intra frames are comparable (i.e., within a 0.3dB difference) for each of the RD points. For each sequence and WZ quantization matrix, the optimal lambda(s) are identified by selecting the set providing the best RD curve. These points are then used to create a graph of (optimal) lambdas as a function of the intra QP, as in Fig. 6.1. For each test sequence, the points were fitted with a continuous exponential function. This results in an approximation of the optimal lambda as a function of the intra QP, for each test sequence, i.e.:

$$\lambda = ae^{-b\cdot \mathrm{QP}}, \tag{6.3}$$

where QP denotes the intra quantization parameter of the key frames, and a and b are constants. The optimal λ is obtained by the work in [17] with fixed $a = 7.6$ and $b = 0.1$ for all sequences.

As shown in Fig. 6.1, the optimal λ differs between the sequences. Typically, for sequences with less motion (such as *Hall Monitor*), the optimal λ is lower to give more weight to the rate term in (6.1) and consequently encourage skip mode. On the other hand, for sequences with complex motion such as *Soccer*, the distortion introduced in the case of skip mode is significant due to errors in the side information, so that higher values for λ give better RD results.

The results in Fig. 6.1 are exploited to estimate the optimal λ on a frame-by-frame basis during decoding. The approach taken is - relatively simple - to look at the rate. Apart from the graph (Fig. 6.1) we also store the average rate per WZ frame associated with each of the points. For sequences with simple motion characteristics (e.g., *Hall Monitor, Coastguard*), for the same intra QP, the WZ rate is typically lower than for more complex sequences such as *Foreman* and *Soccer*. Therefore, during decoding, we first estimate the WZ rate and compare this estimate with the results in Fig. 6.1 to estimate the optimal lambda. Specifically, the WZ rate r_i for the current frame is estimated as the median (med) of the WZ rates r_{i-3}, r_{i-2} r_{i-1} of the three previously decoded WZ frames (as in [24]):

$$r_i = \mathrm{med}(r_{i-1}, r_{i-2}, r_{i-3}). \tag{6.4}$$

By comparing with Fig. 6.1, we then obtain an estimate of the optimal lambda parameter for the current WZ frame to be decoded through interpolation:

$$\lambda_{r_i} = \frac{r_i - r_1}{r_2 - r_1}\lambda_{r_1} + \frac{r_2 - r_i}{r_2 - r_1}\lambda_{r_2}, \tag{6.5}$$

where $r_1 \leqslant r_i \leqslant r_2$ and r_1, r_2 are the rate points for the training sequences with the corresponding λ_{r_1}, λ_{r_2}, respectively.

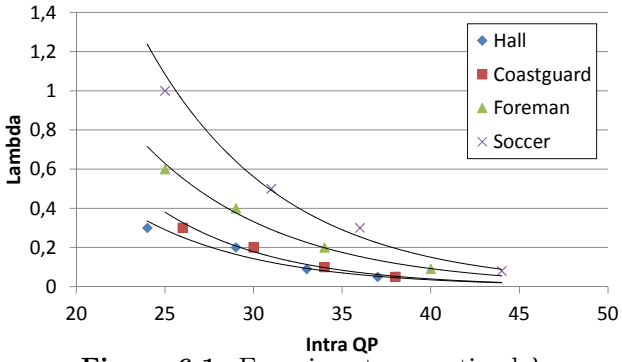


Figure 6.1: Experiments on optimal λ.

6.2 The Adaptive Mode Decision DVC Architectures

6.2.1 The Adaptive Mode Decision Cross-band Codec

The architecture of an efficient TDWZ video codec with a feedback channel [1,9] is depicted in Fig. 6.2. The input video sequence is split into key frames and Wyner-Ziv frames, where the key frames are intra coded using conventional video coding techniques such as H.264/AVC intra coding. The WZ frames are transformed (4×4 DCT), quantized and decomposed into bitplanes. Each bitplane is in turn fed to a rate-compatible LDPCA encoder [26] from most significant bitplane to least significant bitplane. The parity information from the output of the LDPCA encoder is stored in a buffer from which bits are requested by the decoder through a feedback channel.

At the decoder side, OBMC [10] is applied to generate a prediction of each WZ frame available at the encoder-side. This prediction is referred to as the side information (Y). The decoder also estimates the noise residue (R_0) between the SI and the original frame at the encoder. This noise residue is used to

derive the noise parameter α_0 that is used to calculate soft-input information (conditional probabilities $\Pr_0$) for each bit in each bitplane. Given the SI and correlation model, soft input information is calculated for each bit in one bitplane. This serves as the input to the LDPCA decoder. For each bitplane (ordered from most to least significant bitplane), the decoder requests bits from the encoder's buffer via the feedback channel until decoding is successful (using a CRC as confirmation). After all bitplanes are successfully decoded, the WZ frame can be reconstructed through centroid reconstruction followed by inverse transformation.

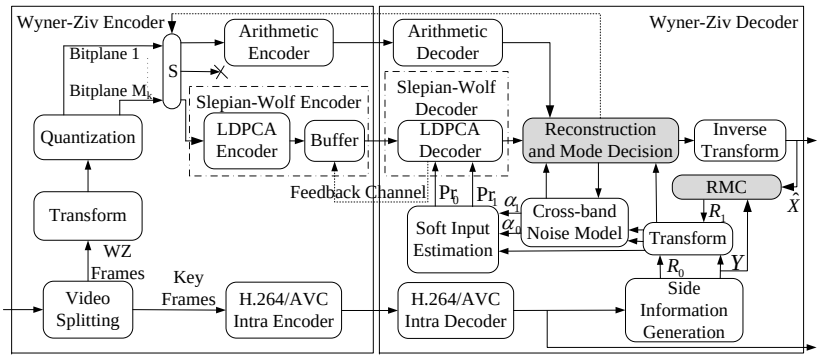


Figure 6.2: Adaptive mode decision TDWZ video architecture.

To improve RD performance of TDWZ video coding, a cross-band noise model in Sec. 2.2.4 [10] utilizing cross-band correlation based on the previously decoded neighboring bands and a mode decision technique [17] have been introduced. In this chapter, we additionally propose the Adaptive Mode Decision (AMD) by adapting rate (Sec. 6.1) and compensating residual motions (Sec. 5.2.1) to further improve the RD performance.

The proposed techniques including the novel adaptive mode decision in Section 6.1 and the novel residual motion compensation in Section 5.2.1 are integrated in the cross-band DVC scheme [10] as shown in Fig. 6.2. The mode decision, S, determines the three modes skip, arithmetic, or WZ coding of each bitplane to be coded. The mode information is updated and sent by the decoder to the encoder after each bitplane is completely processed. The Residual Motion Compensation (RMC) technique (Sec. 5.2.1) generates the additional residue R_1 along with the original residue R_0 generated by the OBMC technique [10] of the side information generation. Thereafter, the cross-band noise model in Sec. 2.2.4 [10] produces the parameters α_0, α_1 for estimating corresponding

soft inputs Pr_0, Pr_1 for the multiple input LDPCA decoder [8]. When all bitplanes are decoded, the coefficients are reconstructed and the inverse transform converts the results to the decoded WZ frames $\widehat{X}$. These frames $\widehat{X}$ are also used along with SI frame Y for the RMC technique to generate the residual frame R_1 for the next frame to be decoded.

6.2.2 The Adaptive Mode Decision MORE2SI Codec

In order to enhance the RD performance of the MORE2SI codec (Chapter 5), the proposed MORE(2SI) scheme is additionally enhanced by integrating an adaptive mode decision (AMD) for the RD points with lowest rate. It is applied for two RD points for *Hall Monitor* and one for *Foreman*, *Soccer*, and *Coastguard* and denoted by MORE2SI(AMD). Mode decision has been proposed to control and take advantage of skip mode and intra mode in DVC [17]. The adaptive mode decision is not only based on the decoder-side mode decision as in [17], but also (decoder side) estimated rate of Wyner-Ziv (WZ) frames to obtain a Lagrange parameter [17]. Figure 6.3 depicts the MORE2SI(AMD) architecture, which includes the MORE2SI scheme (Fig. 5.3) and the AMD technique (Sec. 6.1) determining the three modes skip, arithmetic, or WZ coding of each bitplane.

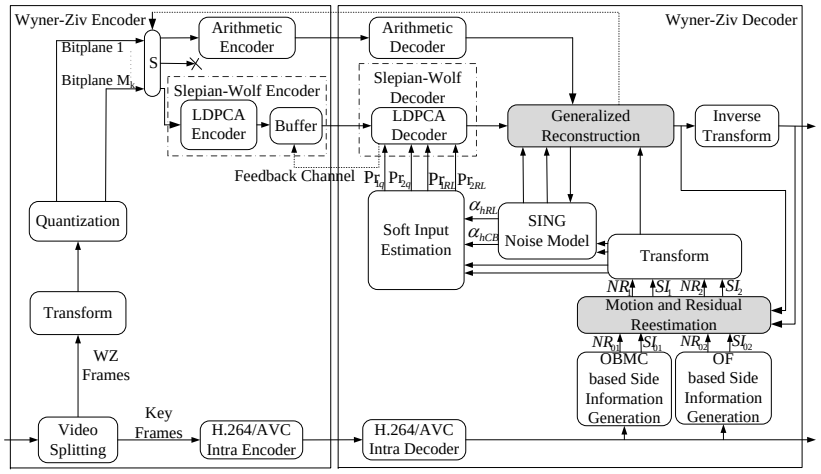


Figure 6.3: Adaptive mode decision MORE video architecture.

6.3 Performance Evaluation

Table 6.1: Bjøntegaard relative bit-rate savings (%) of the proposed AMD techniques over DISCOVER for WZ and all frames

Sequence	Cross-band		MD		AMD		AMDMotion	
	WZ	All	WZ	All	WZ	All	WZ	All
Coastguard	11.61	4.08	13.69	4.61	24.01	5.91	32.62	7.50
Foreman	14.19	5.98	16.88	6.95	21.57	8.42	24.47	9.46
Hall Monitor	8.59	2.55	11.54	3.03	39.68	5.96	59.42	8.18
Mother-daughter	13.51	3.98	21.14	5.44	44.75	8.31	57.58	10.04
Silent	17.33	5.77	22.94	6.58	30.96	7.77	38.82	9.50
Soccer	26.72	14.64	26.81	15.36	26.95	15.49	29.78	16.97
Stefan	2.32	1.15	4.11	2.40	4.26	2.34	5.96	3.20
Average	13.47	5.45	16.73	6.34	27.45	7.74	35.52	9.26

Table 6.2: Bjøntegaard PSNR improvements (dB) of the proposed AMD techniques over DISCOVER for WZ and all frames

Sequence	Cross-band		MD		AMD		AMDMotion	
	WZ	All	WZ	All	WZ	All	WZ	All
Coastguard	0.36	0.19	0.41	0.22	0.65	0.27	0.85	0.34
Foreman	0.65	0.33	0.75	0.38	0.91	0.46	1.02	0.51
Hall Monitor	0.39	0.19	0.51	0.22	1.39	0.41	1.91	0.56
Mother-daughter	0.49	0.22	0.62	0.29	1.11	0.44	1.44	0.53
Silent	0.81	0.36	1.02	0.40	1.29	0.48	1.52	0.58
Soccer	1.33	0.73	1.29	0.75	1.28	0.75	1.42	0.82
Stefan	0.08	0.05	0.15	0.12	0.17	0.12	0.26	0.17
Average	0.59	0.30	0.68	0.34	0.97	0.42	1.20	0.50

Table 6.3: Bjøntegaard relative bit-rate savings (%) of the DVC schemes over DISCOVER for WZ and all frames

Sequence	Cross-band		SING		MORE		MORE(AMD)	
	WZ	All	WZ	All	WZ	All	WZ	All
Foreman	14.19	5.98	35.43	13.63	74.03	**26.22**	**74.03**	26.09
Hall	8.59	2.55	22.71	5.52	36.21	8.05	**55.85**	**8.82**
Soccer	26.72	14.64	62.70	32.83	**101.75**	**50.15**	100.16	49.46
Coast	11.61	4.08	24.98	7.70	44.44	**12.90**	**45.59**	12.88
Average	14.92	6.25	36.46	14.92	64.10	**24.33**	**68.91**	24.31

The RD performance of the proposed techniques are evaluated for the QCIF test sequences (149 frames of) *Coastguard, Foreman, Hall Monitor, Mother-daughter, Silent, Soccer,* and *Stefan* with 15Hz frame rate. The GOP size is 2, where odd frames are coded as key frames using H.264/AVC Intra and even frames are coded using Wyner-Ziv coding. Four RD points are considered corresponding to four 4×4 quantization matrices Q1, Q4, Q7, and Q8 [9]. H.264/AVC Intra corresponds to the intra coding mode of the H.264/AVC reference codec JM 9.5 [43] in main profile. Only the luminance component of each frame is evaluated. The proposed techniques are integrated in the DVC scheme in [10], using the adaptive rate mode decision as in Section 6.1 and combining with the residual motion compensation as in Section 5.2.1 denoted

Table 6.4: Bjøntegaard PSNR improvements (dB) of the DVC schemes over DISCOVER for WZ and all frames

Sequence	Cross-band		SING		MORE		MORE(AMD)	
	WZ	All	WZ	All	WZ	All	WZ	All
Foreman	0.65	0.33	1.52	0.75	**3.00**	**1.43**	2.93	1.41
Hall	0.39	0.19	0.99	0.40	1.42	0.58	**1.95**	**0.61**
Soccer	1.33	0.73	2.70	1.51	**4.19**	**2.26**	4.18	2.23
Coast	0.36	0.19	0.76	0.37	**1.28**	**0.61**	1.24	0.60
Average	0.64	0.33	1.49	0.76	2.47	1.22	**2.58**	**1.22**

by AMD and AMDMotion, respectively. Results for the DVC scheme in Sec. 2.2.4 [10] (Cross-band) and the mode decision in [17] integrated in the Cross-band DVC [10], denoted by Cross-band and MD, respectively, are also given.

Table 6.1 presents the average bitrate savings and equivalently the average PSNR improvements using the Bjøntegaard metric [35] compared with the DISCOVER codec for WZ frames as well as for all frames. Compared with DISCOVER, the average bitrate saving for the proposed AMDMotion scheme is 35.5% and 9.26% for WZ frames and all frames, respectively. In particular, the performance improvement is 59.4% and 8.18% (or equivalently the average improvement in PSNR is 1.91 dB and 0.56 dB) for WZ frames and overall frames for the low motion *Hall Monitor* sequence. Compared with the Cross-band DVC scheme (Sec. 2.2.4) [10], an average bit-rate saving (Bjøntegaard) of 22.1% is observed on the WZ frames. It is clear that AMD outperforms MD with an average relative bitrate saving on WZ frames of 27.5% compared with 16.7%.

The RD performance of the proposed AMD and AMDMotion codecs and H.264/AVC coding is also depicted in Fig. 6.4 for WZ frames and all frames. The AMDMotion codec gives a better RD performance than H.264/AVC Intra coding for all the sequences except *Soccer* and *Stefan* and also better than H.264/AVC No Motion for *Coastguard*. Furthermore, the proposed AMDMotion codec improves performance in particular the lower motion sequences *Hall Monitor, Silent,* and *Mother-daughter*. In general, the RD performance of the AMDMotion codec clearly outperforms those of the Cross-band scheme (Sec. 2.2.4) [10] and DISCOVER.

Furthermore, experiments were conducted enhancing the proposed MORE(2SI) scheme by integrating an adaptive mode decision (AMD) for the RD points with lowest rate. It is applied for two RD points for *Hall Monitor* and one for *Foreman*, *Soccer*, and *Coastguard*. Furthermore, the MORE2SI(AMD) codec only using skip mode achieved 68.9% in average bitrate saving (or equivalent the average improvement in PSNR is 2.6 dB) on WZ frames for GOP2 improving the 64.1% of MORE(2SI) (Tables 6.3-6.4). The improvement over MORE(2SI) was mainly achieved by a significant improvement of the RD performance for the low motion sequence *Hall Monitor* with an average bitrate saving of 55.8% compared with 36.2% that of the MORE(2SI) scheme (Chapter 5). The RD performance of the proposed DVC codecs and H.264/AVC coding is also depicted in Figs. 6.6-6.7 for *Hall Monitor* for WZ frames and all frames.

6.4 Summary

Adaptive mode decision DVC with residual motion compensation was introduced to utilize skip, intra, and WZ modes based on rate estimation and combined with a more accurate correlation noise estimate. The adaptive mode decision used the estimated rate to more accurately determine the modes during decoding. Moreover, the residual motion compensation generated an additional residue to take advantage of correlation between the previously decoded and current noise residues. Experimental results show that the coding efficiency of the proposed AMDMotion scheme can robustly improve the RD performance of TDWZ DVC without changing the encoder. For a GOP size of 2 the average bitrate saving of the AMDMotion codec is 35.5% (or equivalent the average improvement in PSNR is 1.2dB) on WZ frames compared with the DISCOVER codec. On the four test sequences, the average bitrate saving of the MORE(AMD) is 69% (or equivalent the average improvement in PSNR is 2.6 dB) on WZ frames compared with the DISCOVER codec.

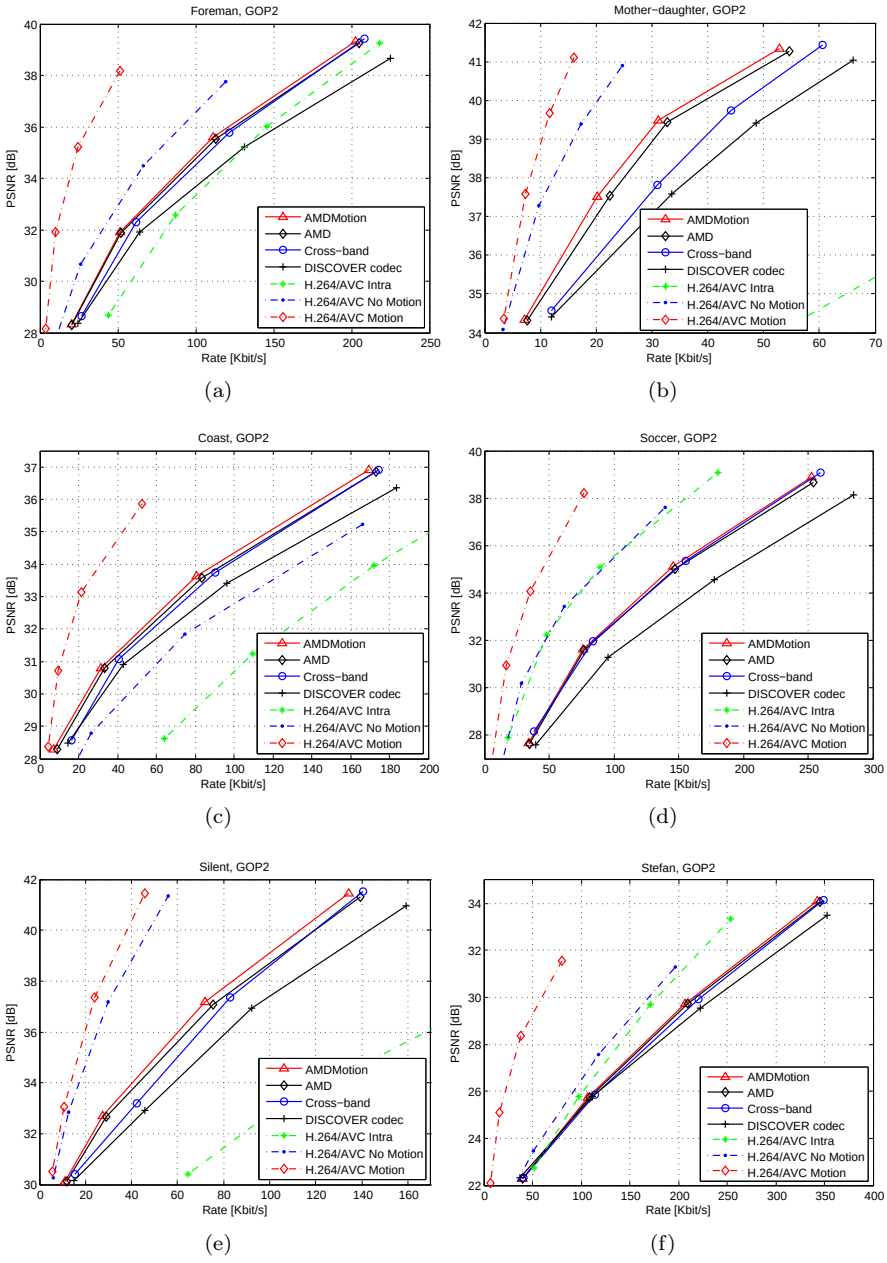


Figure 6.4: PSNR vs. rate for the proposed AMD codecs for WZ frames.

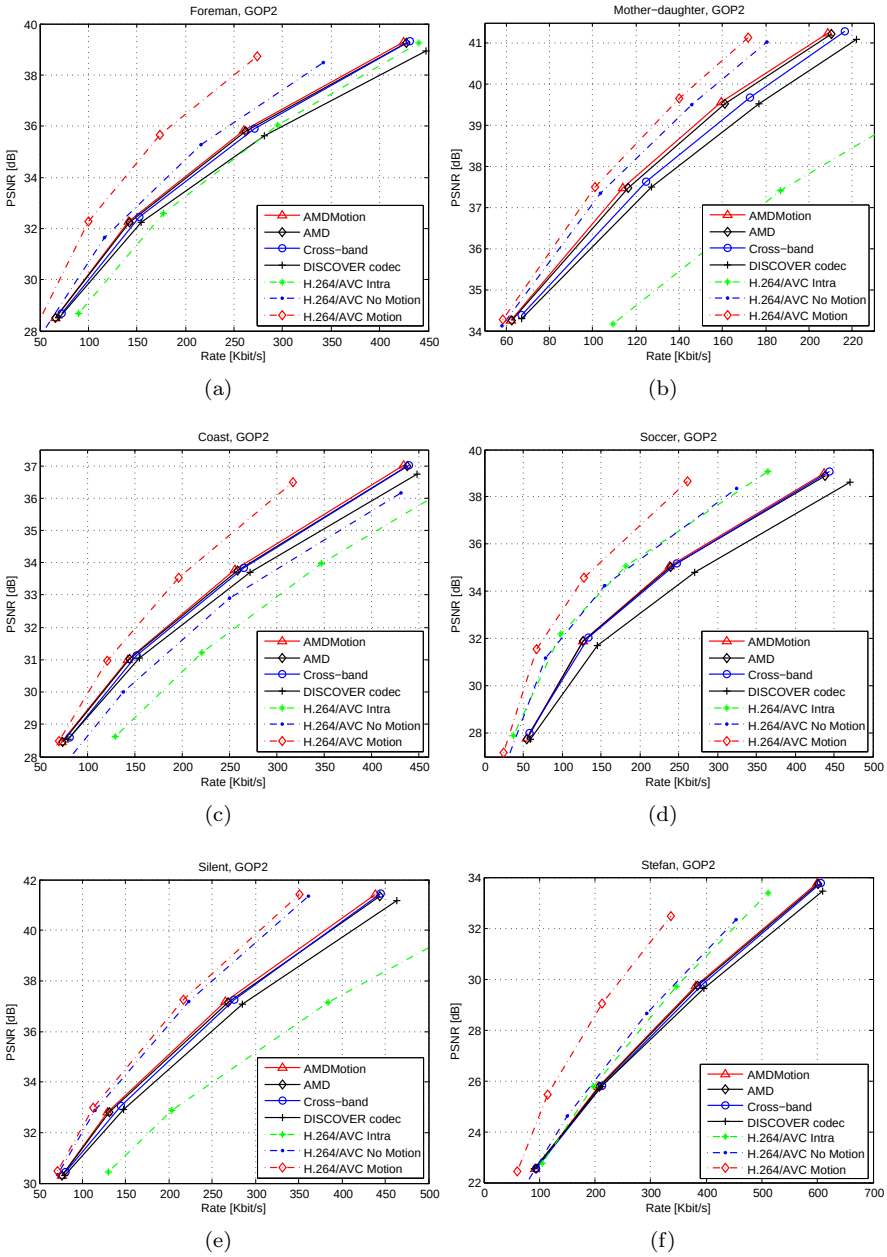
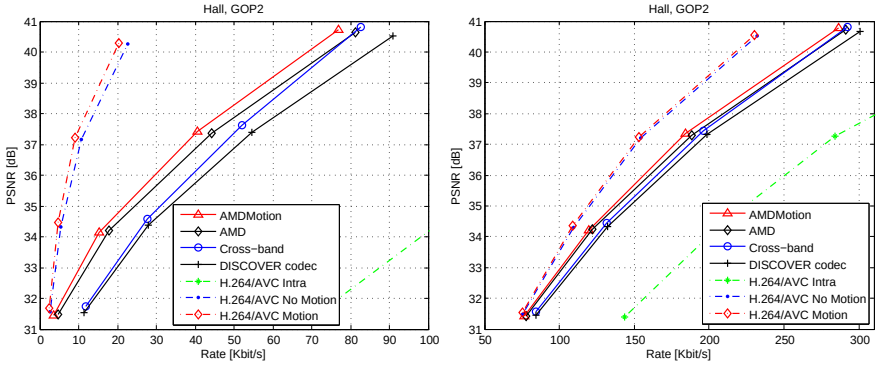
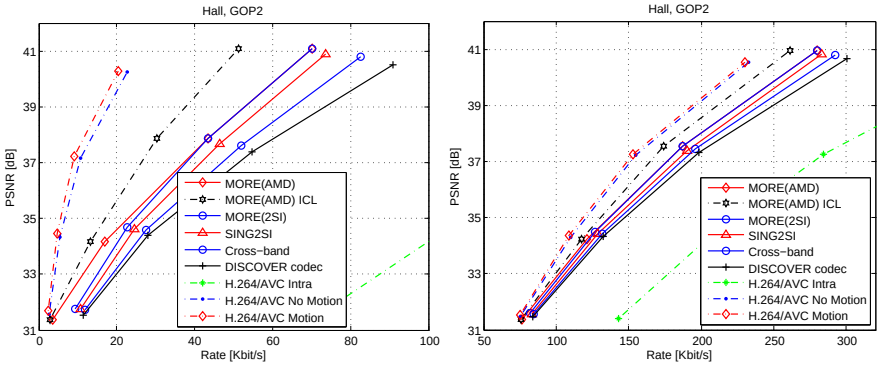


Figure 6.5: PSNR vs. rate for the proposed AMD codecs for all frames.



(a) WZ frames (b) All frames

Figure 6.6: PSNR vs. rate for the proposed AMD codecs for *Hall*.



(a) WZ frames (b) All frames

Figure 6.7: PSNR vs. rate for the proposed DVC schemes for *Hall*.

Chapter 7

Conclusion

In this thesis, theoretical and practical issues of DVC were investigated and iterative improvement algorithms were proposed to improve the compression performance of DVC. The backgrounds of theoretical and practical results in DVC were disscussed in Chapter 2 as a starting point for the contributions in the following Chapters. A Wyner-Ziv video codec with parallel iterative LDPC decoding was proposed in Chapter 3, where the technique took bitplane correlation into account by iteratively refining the soft-input for each bitplane and updating the noise distribution during decoding. In Chapter 4, the TDWZ video coding was also improved using optical flow and clustering of DCT blocks. Optical flow was used for frame interpolation generating side information, which was adopted in a multi-hypothesis scheme to compensate weaknesses of block based methods. Adaptive noise modeling using clustering was introduced additionally utilizing residues of previously decoded frames and generating a number of noise residual distributions within a frame for adaptive optimization of the soft side information during decoding. Furthermore, in Chapter 5, motion reestimation using optical flow was proposed to take advantage of the partially decoded information to generate more accurate side information and residual frames. The residual motion compensation following the motion updating generated additional residues to exploit the correlation between the previously decoded and current noise residues. A generalized reconstruction algorithm was proposed to optimize the multi-hypothesis reconstruction by refining the weighting parameters. Finally, an adaptive mode decision DVC with residual motion compensation was introduced in Chapter 6 to utilize skip, intra, and WZ modes based on rate estimation and combined with a more accurate correlation noise estimate. The adaptive mode decision used the estimated rate to more accurately determine the modes during decoding.

The proposed iterative improvement algorithms provide numerous solutions to

improve the RD performance for the DVC scheme, in which the OF techniques are integrated. On test sequences with GOP size of 2, the proposed DVC scheme with parallel iterative LDPC decoding (Chapter 3) can improve the coding efficiency of TDWZ in terms of WZ rate savings up to 5.6% compared with the available TDWZ video codec [13]. For the proposed SING scheme (Chapter 4), the average bitrate saving of the SING(3SI) codec is 37% (or equivalent the average improvement in PSNR is 1.5 dB) on WZ frames compared with the DISCOVER codec. The most improvement is obtained by the MORE scheme (Chapter 5), where the average improvement in PSNR is 2.5 dB (or equivalent the average bitrate saving of 64%) of the MORE(2SI) codec on WZ frames compared with the DISCOVER codec. In addition, on the number of test sequences, the average bitrate saving of the AMDMotion codec (Chapter 6) based on the Cross-band scheme (Sec. 2.2.4) [10] with the adaptive mode decision is 35.5% (or equivalent the average improvement in PSNR is 1.2 dB) on WZ frames compared with the DISCOVER codec. Finally, on the four test sequences, the the average bitrate saving of MORE(AMD) based on the MORE scheme (Chapter 5) with the adaptive mode decision is 69% (or equivalent the average improvement in PSNR is 2.6 dB) on WZ frames compared with the DISCOVER codec. The experimental results show that the proposed DVC codec gives the better RD performance than the low-complex H.264/AVC Intra for all four test sequences and also the better RD performance than H.246/AVC No Motion for *Foreman*, *Soccer*, and *Coastguard*.

The experimental results have proved that the proposed algorithms in this thesis were efficient for improving the coding performance of the TDWZ video. Besides classic issues of side information generation and the accuracy of noise modeling, the work may be extended in the future:

- **Slepian-Wolf Coding with Multiple LDPC Decoders**: In a Wyner-Ziv video coding structure, the Slepian-Wolf codec plays an important role. The experimental results indicate that the DVC scheme with multiple LDPCA decoders improves the coding performance in terms of bitrate. How to estimate and adapt the multiple LDPCA decoders is still challenging.

- **Adaptive Mode Decisions**: Integrating adaptive mode decisions with advanced refinement techniques is one promising approach to take advantage of skip and intra modes to further improve the DVC coding efficiency.

- **Feedback Channel Constraints with Refinements**: Improving the coding efficiency of more practical DVC codecs with flexible number of requests and refinement techniques could be one of research directions.

- **Reconstruction with Postprocessing**: Improving the reconstruction process using postprocessing may be a solution to overcome the drawbacks of reconstructing in DVC which the reconstruction is only carried out on individual coefficients.
- **DVC over Error Channel**: In real situations, when syndromes are transmitted in error prone environment, the DVC over error channel could be a challenging issue.

Appendix A

The fuzzy C-means (FCM) clustering

Consider a given finite set R, with elements $R_k \in \mathfrak{R}^{16}$ i.e. the set of 16-dimensional real numbers called the feature space, i.e. $\mathrm{R} = \{R_1, R_2, ..., R_N\}$ with feature vectors $R_k = \{R_k^1, R_k^2, ..., R_k^{16}\}$. Let $\mathrm{V} = \{V_1, V_2, ..., V_M\}$ be the cluster centers, $V_i \in \mathfrak{R}^{16}$. A feature vector R_k belongs to a specific cluster V_i that is given by the membership value u_{ik} which can be represented by a matrix $\mathrm{U} \in \mathfrak{R}_{MN}$, where $\mathfrak{R}_{MN}$ is the set of real $M \times N$ matrices. The FCM algorithm iteratively optimizes the standard FCM objective function defined as:

$$J_m(\mathrm{U}, \mathrm{V}) = \sum_{k=1}^{N} \sum_{i=1}^{M} u_{ik}^m d_{ik}^2, \tag{A.1}$$

where $d_{ik}^2 = \|R_k - V_i\|^2$ represents the squared Euclidean distance between the feature vector R_k and center V_i, $m \geqslant 1$ is the degree of fuzzification. The optimization is initiated using the constraint $\sum_{i=1}^{M} u_{ik} = 1$.

Local minimization of the objective function $J_m(\mathrm{U}, \mathrm{V})$ is accomplished by iteratively adjusting the values of u_{ik} and V_i according to the following equations:

$$u_{ik} = \frac{1}{\sum_{j=1}^{M} \left(\frac{d_{ik}}{d_{jk}} \right)^{2/(m-1)}}, \tag{A.2}$$

$$V_i = \frac{\sum_{k=1}^{N} u_{ik}^m R_k}{\sum_{k=1}^{N} u_{ik}^m}. \tag{A.3}$$

As J_m is iteratively minimized, V_i becomes more stable. Iteration of feature vector groupings is terminated at iteration t when the termination measurement $\max_{1 \leqslant i \leqslant M} \{\|V_i^{(t)} - V_i^{(t-1)}\|\} < \epsilon$ is satisfied, where $V_i^{(t)}$ is an updated center, $V_i^{(t-1)}$ is the previous center, and ϵ is the predefined termination threshold. Finally, all feature vectors are classified into clusters by assigning a feature vector R_k to the cluster V_j for $u_{jk} = \max_{1 \leqslant i \leqslant M} \{u_{ik}\}$. The FCM algorithm converges to a minimum or a saddle point [50].

Appendix B

The cluster-based variance

Lemma: Let R be a data set where R is classified into non-overlapping sub-sets. The variance σ^2 of a set R is higher than the expected variance of the sub-sets.

Proof: Assume $\mathrm{R} = \{R_k\}, 1 \leqslant k \leqslant N$ is separated into M clusters, for instance, cluster j $(1 \leqslant j \leqslant M)$ includes N_j elements that are denoted by $R_{j(i)} (1 \leqslant i \leqslant N_j)$, where $\sum_j N_j = N$. σ^2 and σ_j^2 are the variances of R and a set j including N_j elements $R_{j(i)}$ given j, respectively. What we need to prove is:

$$\sigma^2 \geqslant \frac{1}{N} \sum_j \sum_i (R_{j(i)} - \mathrm{E}_j[R_{j(i)}])^2, \tag{B.1}$$

where $\mathrm{E}_j[.]$ is the expectation operator of elements given j, this means the elements $R_{j(i)}$ are included in a set j.

Equation (B.1) is equivalent to

$$\begin{aligned} \sigma^2 &\geqslant \frac{1}{N} \sum_j N_j \sigma_j^2 \\ \Leftrightarrow N(\mathrm{E}[R^2] - E[R]^2) &\geqslant \sum_j N_j (\mathrm{E}_j[R_{j(i)}^2] - E_j[R_{j(i)}]^2), \end{aligned} \tag{B.2}$$

where $N\mathrm{E}[R^2] = \sum_j N_j \mathrm{E}_j[R_{j(i)}^2]$ because $N\mathrm{E}[R^2] = \sum_k R_k^2$ and $\sum_j N_j \mathrm{E}_j[R_{j(i)}^2] = \sum_j \sum_i R_{j(i)}^2$.

Equation (B.2) is equivalent to

$$\begin{aligned}
\sum_j N_j \mathrm{E}_j[R_{j(i)}]^2 &\geqslant N\mathrm{E}[R]^2 \\
\Leftrightarrow N\sum_j N_j \left(\frac{\sum_i R_{j(i)}}{N_j}\right)^2 &\geqslant (\sum_k R_k)^2 \\
\Leftrightarrow (\sum_j (\sqrt{N_j})^2)\left(\sum_j \left(\frac{\sum_i R_{j(i)}}{\sqrt{N_j}}\right)^2\right) &\geqslant (\sum_j (\sum_i R_{j(i)}))^2, \qquad \text{(B.3)}
\end{aligned}$$

which is true due to the Cauchy-Schwarz inequality for any real number $N_j > 0$ and $R_{j(i)}$. The two sides are equal if and only if the ratios $\frac{\sum_i R_{j(i)}}{N_j}$ are equal.

Appendix C

Publications

The selected Ph.D. publication contributions are fully reported.

TIP12: Huynh Van Luong, Lars Lau Rakêt, Xin Huang, and Søren Forchhammer, "Side Information and Noise Learning for Distributed Video Coding using Optical Flow and Clustering," *IEEE Transactions on Image Processing*, vol. 21, no. 12, pp. 4782-4796, Dec. 2012.

ICASSP11: Huynh Van Luong, Xin Huang, and Søren Forchhammer, "Multiple LDPC Decoding using Bitplane Correlation for Trainsform Domain Wyner-Ziv Video Coding," in *IEEE International Conference on Acoustics, Speech and Signal Processing 2011 (ICASSP 2011)*, Prague, Czech Republic, May 2011.

ICIP11: Huynh Van Luong, Xin Huang, and Søren Forchhammer, "Parallel Iterative Decoding of Transform Domain Wyner-Ziv Video using Cross Bitplane Correlation," in *IEEE International Conference on Image Processing 2011 (ICIP 2011)*, Brussels, Belgium, Sep. 2011.

MMSP11-1: Huynh Van Luong, Xin Huang, and Søren Forchhammer, "Adaptive Noise Model for Transform Domain Wyner-Ziv Video using Clustering of DCT Blocks," in *IEEE International Workshop on Multimedia Signal Processing 2011 (MMSP 2011)*, Hangzhou, China, Oct. 2011.

MMSP11-2: Xin Huang, Lars Lau Rakêt, Huynh Van Luong, Mads Nielsen, Francois Lauze, and Søren Forchhammer, "Multi-hypothesis Transform Domain Wyner-Ziv Video Coding including Optical Flow," in *IEEE International Workshop on Multimedia Signal Processing 2011 (MMSP 2011)*, Hangzhou, China, Oct. 2011. (Top 10% paper award)

PCS12: Huynh Van Luong and Søren Forchhammer, "Noise Residual Learning for Noise Modeling in Distributed Video Coding," in *Picture Coding Symposium 2012 (PCS 2012)*, Krakow, Poland, May 2012.

SPIE12: Lars Lau Rakêt, Jacob Søgaard, Matteo Salmistraro, Huynh Van Luong, and Søren Forchhammer, "Exploiting the Error-Correcting Capabilities of Low Density Parity Check Codes in Distributed Video Coding using Optical Flow," in *SPIE Optics+ Photonics, Optical Engineering+Applications*, San Diego, California, Aug. 2012.

4782 IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 21, NO. 12, DECEMBER 2012

Side Information and Noise Learning for Distributed Video Coding Using Optical Flow and Clustering

Huynh Van Luong, Lars Lau Rakêt, Xin Huang, and Søren Forchhammer, *Member, IEEE*

***Abstract*— Distributed video coding (DVC) is a coding paradigm that exploits the source statistics at the decoder side to reduce the complexity at the encoder. The coding efficiency of DVC critically depends on the quality of side information generation and accuracy of noise modeling. This paper considers transform domain Wyner–Ziv (TDWZ) coding and proposes using optical flow to improve side information generation and clustering to improve the noise modeling. The optical flow technique is exploited at the decoder side to compensate for weaknesses of block-based methods, when using motion-compensation to generate side information frames. Clustering is introduced to capture cross band correlation and increase local adaptivity in the noise modeling. This paper also proposes techniques to learn from previously decoded WZ frames. Different techniques are combined by calculating a number of candidate soft side information for low density parity check accumulate decoding. The proposed decoder side techniques for side information and noise learning (SING) are integrated in a TDWZ scheme. On test sequences, the proposed SING codec robustly improves the coding efficiency of TDWZ DVC. For WZ frames using a GOP size of 2, up to 4-dB improvement or an average (Bjøntegaard) bit-rate savings of 37% is achieved compared with DISCOVER.**

***Index Terms*— Adaptive noise, distributed video coding, multihypothesis, noise residual learning, optical flow.**

I. Introduction

DISTRIBUTED video coding is an interesting instance of distributed source coding where the video redundancy is partly or fully exploited at the decoder side. In recent years, conventional video coding has been challenged by some emerging applications, such as video surveillance and video sensor networks, which require a relatively low cost encoder with high coding efficiency. DVC [1], [2] has been proposed as a solution. DVC is based on two information theoretic results, namely the Slepian-Wolf Theorem [3] and the Wyner-Ziv Theorem [4], promising efficient lossy coding of correlated source data sets when independent encoding and joint decoding are performed utilizing the correlation between the sources only at the decoder side.

Manuscript received January 22, 2012; revised May 31, 2012; accepted August 6, 2012. Date of publication August 27, 2012; date of current version November 14, 2012. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. James E. Fowler.

H. V. Luong, X. Huang, and S. Forchhammer are with the Department of Photonics Engineering, Technical University of Denmark, Lyngby 2800, Denmark (e-mail: hulu@fotonik.dtu.dk; xhua@fotonik.dtu.dk; sofo@fotonik.dtu.dk).

L. L. Rakêt is with the Department of Computer Science, University of Copenhagen, Copenhagen 2100, Denmark (e-mail: larslau@diku.dk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2012.2215621

Transform Domain Wyner-Ziv (TDWZ) video coding [1] is one efficient approach to DVC. The coding efficiency is highly dependent on the accuracy of side information at the decoder. A soft-input estimate is calculated at the Wyner-Ziv decoder, obtained by side information frame generation and noise modeling calculated using reference frames [5]–[7]. Although the quality of side information frames and the accuracy of the noise model [5] have been improved [6], [7], the coding efficiency of TDWZ coding trails that of conventional video coding solutions, such as H.264/AVC, most notably for high motion sequences. We shall consider techniques which can enhance the performance of these basic TDWZ schemes and thereafter integrate the proposed techniques in the DVC codec in [7] to enhance performance. As one technique for improved performance, multiple side information based TDWZ has been proposed [8], [9]. In [8], two different frame interpolation methods are employed, but the Wyner-Ziv decoder only considers the average of the two estimates for decoding and reconstruction. In [9], the results of frame interpolation and frame extrapolation are combined using weighting to generate multiple soft-inputs to the decoder in a TDWZ scheme. However, the contribution brought by frame extrapolation is limited and only used for the soft inputs, while for the reconstruction part, only the frame interpolation is used. Providing multiple soft inputs to the Slepian-Wolf (SW) decoder may be seen as a generic way to introduce adaptivity in SW coding and thereby in TDWZ.

In order to enhance performance and reduce the rate-distortion gap between TDWZ and conventional video coding, which is especially pronounced in high motion sequences, a multiple-input TDWZ decoder is used in this paper. Multiple versions of soft side information are generated by applying both block based and optical flow based side information generation techniques using frame interpolation. The intuition is that optical flow based frame interpolation can generate side information which is different and to some extent may compensate the weaknesses in block based methods, if the scheme allows the techniques to efficiently compensate each other. Optical flow has previously been used in a DVC scheme [10], where the optical flow was calculated using the classical method of Lucas and Kanade [11], which is a local method that can be considered as a limit of block matching. In this paper we propose to use a global method for optical flow based on an TV-L^1 energy, which should complement block-based approaches better.

Furthermore, in contrast to previous multiple soft-input DVC methods [9], the decoding and reconstruction are based

on a weighted joint distribution. In this way, the proposed multi-hypothesis based TDWZ decoder will not only reduce the required bitrate for decoding but also improve the quality of reconstructed frames.

The noise estimation is also an important aspect influencing the coding performance. The decoder needs to estimate the correlation between the corresponding source and the side information, which can be obtained through frame interpolation at the decoder side. The accuracy of the correlation has a significant impact on the compression performance of DVC. Our goal is to improve coding efficiency by improving the adaptive noise modeling and by better learning of the correlation between source and side information using both spatial and temporal correlation. Several noise models [5], [7], [12] have been proposed using the Laplacian distribution for the DCT coefficients. The advanced noise models operate with different granularity levels, e.g. frame level, band level, and coefficient level. Estimating the correlation noise has been enhanced by utilizing the correlation of coefficients in each residual frame [5], [12], [13] and noise residual refinement [7] in the transform domain.

The technique in [13] estimates the correlation noise by first classifying blocks within a frame. A residual energy between source and side information of a given block is used to classify blocks, and for each class a predefined value of the Laplacian parameter is assigned. In [7], the reconstructed bands were used to influence the noise model for subsequent bands by classifying the reconstructed band into two categories. The cross-band correlation was only based on 1-2 already decoded neighboring bands. Furthermore, two categories may not be enough to fully utilize the correlation. The noise residue refinement [7] updates the estimated noise residue for noise modeling and side information quality during decoding. More recently, an initial work on an adaptive noise model using clustering of DCT blocks was presented [14] to explore cross-band correlation. This technique not only utilizes the correlation over all bands but takes the decoded bands into account to influence the decoding of subsequent bands. In a recent paper [15], adaptive correlation is performed integrated in joint bitplane decoding.

In order to further improve the noise estimation, this paper proposes a refinement technique that utilizes clustering of DCT blocks for cross-band correlation and enhances performance by using the correlation of neighbor coefficients to refine the Laplacian parameter of the coefficient considered, and thereafter, updates the noise parameters. To utilize the temporal redundancy, we shall use residuals of already decoded (WZ) frames to influence the noise distribution of the current frame. As a last enhancement of the noise model, adaptive optimization of the number of clusters in the noise model is addressed to adaptively get the best soft side information during decoding. These improvements of noise modeling are finally combined with the side information generation using optical flow. The techniques are combined using a multiple soft input decoding approach.

The rest of this paper is organized as follows. In Section II, the architecture considered for TDWZ video coding is presented, including the version in [7], which we take as starting point. A global optical flow technique is presented in Section III. The use of clustering in DVC noise modeling is presented in Section IV along with the new learning techniques proposed. A new TDWZ video scheme is presented in Section V based on enhancing the basic TDWZ with optical flow in a multi-hypothesis set-up and the new clustering for enhanced noise modeling. Section VI presents simulation results, analyzes the contributions of the different techniques and compares the performance with reference methods.

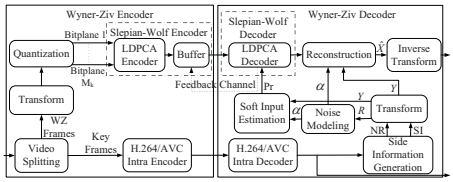


Fig. 1. Transform domain Wyner–Ziv video codec architecture.

II. Transform Domain Wyner–Ziv Video Codecs

A popular and efficient approach to DVC is TDWZ video coding with a feedback channel [1], where the decoder controls the rate by requests over a feedback channel. The DISCOVER codec [5] improved performance of the initial TDWZ architecture. More recently, TDWZ video coding with a cross-band noise model was proposed [7] to further improve the coding efficiency by utilizing the cross-band correlation.

A. Transform Domain Wyner–Ziv Video

The architecture of a TDWZ video codec [5] is depicted in Fig. 1. In this system, the sequence of frames is split into key frames and so-called Wyner-Ziv frames. Key frames are intra coded using conventional video coding techniques such as H.264/AVC intra coding. The Wyner-Ziv frames are transformed (4×4 DCT), quantized and decomposed into bitplanes. Each bitplane is fed to a rate-compatible low density parity check accumulate (LDPCA) encoder [16] from most significant bitplane to least significant bitplane. The corresponding error correcting information is stored in a buffer and requested by the decoder through a feedback channel.

The Wyner-Ziv frame is predicted at the decoder side by using already decoded frames as references. The predicted frame, called the Side Information (SI) frame, is an estimate of the original Wyner-Ziv frame. Given the available SI, soft-input information (conditional probabilities Pr for each bit) within each bitplane is estimated using a noise model. Thereafter the LDPCA decoder starts to decode the bitplanes selected by the quantizer, ordered from most to least significant bitplane, to correct the bit errors. The decoder requests bits from the buffer until the bitplane is decoded. Thereafter CRC check bits are sent for confirmation. After all the bitplanes are successfully decoded, the Wyner-Ziv frame can be decoded through combined de-quantization and reconstruction followed by an inverse transform.

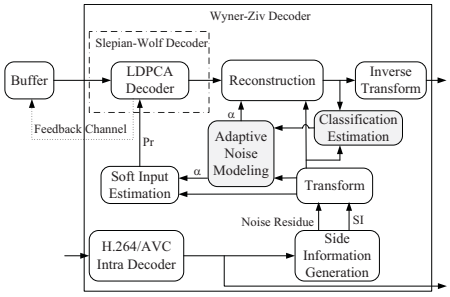


Fig. 2. Transform domain Wyner–Ziv video codec architecture with a cross-band-based adaptive noise model.

B. TDWZ Video With Cross-Band Noise Model

Coefficient level noise models [7], [12] using coefficient classifications have been proposed to further improve the coding efficiency of TDWZ. The noise model in [12] classifies coefficients into two different categories based on motion estimated residues and assigns a predefined parameter for each category. A cross-band noise model [7] was introduced utilizing cross-band correlation based on the previously decoded neighboring bands. This decoder side cross-band noise model [7], which was proposed to improve RD performance of TDWZ video coding, is shown in Fig. 2. The decoder noise model includes a classification module, which is used by the adaptive noise model. The classification utilizes successfully decoded neighboring lower frequency bands to evaluate the higher frequency bands and classifies coefficients into different categories reflecting their reliability. The adaptive noise model uses a modified maximum likelihood estimator, which is applied to the different reliability classes in order to calculate a higher level noise parameter first. Thereafter, a lower level noise parameter is adaptively determined for each coefficient. Furthermore, a bitplane level noise residue refinement (NRR) scheme was applied in the cross-band decoder to adaptively refine the quality of side information frame during decoding. An overlapping block motion compensation scheme (OBMC) was used for side information generation [7]. In this paper, the scheme presented in [7] is adopted as the baseline cross-band codec.

III. Optical Flow Side Information Generation

To improve OBMC based performance, optical flow is also considered for side information generation. Optical flow estimation concerns the determination of apparent (projected) motion. Given a set of images I_{-1} and I_1 in pixel domain, we want to estimate the dense flow field v such that $I_1(\boldsymbol{x}+v(\boldsymbol{x}))$ is close to $I_{-1}(\boldsymbol{x})$ with respect to some suitable measure, where $\boldsymbol{x}$ denotes a point in the image.

A. Duality-Based TV-L^1 Optical Flow

One of the most successful approaches to optical flow estimation is to recover the flow as the minimizer of an energy (see e.g. Baker et al. [17]). Typically the problem is considered as having a spatially continuous domain, and using variational methods, the flow v is recovered as a minimizer of an energy of the form

$$E(v) = \lambda F(I_{-1}, I_1, v) + G(v) \tag{1}$$

where F is a positive functional measuring data fidelity, G is a regularization term and λ is the parameter that determines the tradeoff between data fidelity and regularity. Many energies of this type have been suggested throughout the years (e.g. [18]–[22]), and a large variety of resolution strategies exist. Block based methods contrast regularized optical flow by the lack of a specific regularization term, since regularity is imposed by means of block sizes. Due to this limited reach of the block regularization, one may not necessarily be able to determine motion in untextured areas, and the motion ambiguity caused by the aperture problem may create problematic estimates. By including an explicit regularization term, the regularization will automatically reach throughout the image and give better motion estimates in untextured areas. The problem of untextured areas may sometimes cause problems in an interpolation setup as the intermediate frame is constructed by following motion vectors, and a wrong match in the surrounding images may create unwanted artifacts in the interpolated frame. The continuous formulation (1) of the optical flow energy may also have an advantage for motion estimation when objects are severely deformed, e.g. a face changing expression. Here the rigidity of the blocks may not be able to obtain a good match, while the continuous formulation of optical flow methods may be able to handle this better.

Here we will focus on the TV-L^1 energy (in the sense of Rakêt et al. [23]), where data fidelity between two frames I_{-1} and I_1 is measured by the L^1-norm of the difference:

$$F(I_{-1}, I_1, v) = \int \|I_1(\boldsymbol{x} + v(\boldsymbol{x})) - I_{-1}(\boldsymbol{x})\| \, \mathrm{d}\boldsymbol{x} \tag{2}$$

and the regularization term G penalizes the total variation of the estimated motion:

$$G(v) = \int \|\mathscr{D}v(\boldsymbol{x})\| \, \mathrm{d}\boldsymbol{x} \tag{3}$$

which is to be understood as the integral of the Frobenius norm of the derivative of v [24].

From the results of the Middlebury Optical Flow Database [17], one can see that both TV-L^1 based methods as well as block based methods typically do quite well, when it comes to interpolation quality. It is however also evident that the two approaches produce quite different motion fields, resulting in different types of interpolation errors, which means that the two methods often complement each other very well.

1) Minimization: In the following we will describe the minimization procedure. Because of the high degree of nonlinearity direct minimization of E is not feasible, so we have to relax the energy. First we relax the data fidelity term by replacing $I_1(\boldsymbol{x} + v) - I_{-1}(\boldsymbol{x})$ with its first order Taylor approximation $\rho(v)$ around a given estimate of the flow $\boldsymbol{v}_0$. We see that ρ is linear in v

$$\rho(v)(\boldsymbol{x}) = I_1(\boldsymbol{x} + \boldsymbol{v}_0) - I_{-1}(\boldsymbol{x}) + J_{I_1}(\boldsymbol{x} + \boldsymbol{v}_0)(v(\boldsymbol{x}) - \boldsymbol{v}_0) \tag{4}$$

where J_{I_1} is the Jacobian of I_1. Note that in the case of intensity data, the Jacobian is merely the transpose of the gradient ∇I_1. We furthermore introduce an additional relaxation in the form of an auxiliary variable u that splits the data fidelity and regularization functions in two quadratically coupled energies:

$$E_1(v) = \lambda \int \|\rho(v)(\boldsymbol{x})\| \, \mathrm{d}\boldsymbol{x} + \frac{1}{2\theta} \int \|v(\boldsymbol{x}) - u(\boldsymbol{x})\|^2 \, \mathrm{d}\boldsymbol{x} \quad (5)$$

$$E_2(u) = \frac{1}{2\theta} \int \|v(\boldsymbol{x}) - u(\boldsymbol{x})\|^2 \, \mathrm{d}\boldsymbol{x} + \int \|\mathscr{D}u(\boldsymbol{x})\| \, \mathrm{d}\boldsymbol{x}. \quad (6)$$

This relaxation, which was first proposed by Zach et al. [25], has a number of advantages, most notably that the two problems can be solved pointwise. This makes the solution very easy to implement on massively parallel processors like graphics processing units.

For grayscale images the pointwise minimizer of (5) is given by

$$v(\boldsymbol{x}) = u(\boldsymbol{x}) - \pi(u)(\boldsymbol{x}) \quad (7)$$

where

$$\pi(u)(\boldsymbol{x}) = \begin{cases} -\lambda\theta \nabla I_1(\boldsymbol{x} + \boldsymbol{v}_0) \\ \qquad \text{if } \rho(u)(\boldsymbol{x}) < -\lambda\theta |\nabla I_1(\boldsymbol{x} + \boldsymbol{v}_0)|^2 \\ \lambda\theta \nabla I_1(\boldsymbol{x} + \boldsymbol{v}_0) \\ \qquad \text{if } \rho(u)(\boldsymbol{x}) > \lambda\theta |\nabla I_1(\boldsymbol{x} + \boldsymbol{v}_0)|^2 \\ \frac{\rho(u)(\boldsymbol{x})}{|\nabla I_1(\boldsymbol{x}+\boldsymbol{v}_0)|} \nabla I_1(\boldsymbol{x} + \boldsymbol{v}_0) \\ \qquad \text{if } |\rho(u)(\boldsymbol{x})| \leq \lambda\theta |\nabla I_1(\boldsymbol{x} + \boldsymbol{v}_0)|^2. \end{cases} \quad (8)$$

The minimization of the total variation regularization energy (6) is done using the dual method of Chambolle [24], [26].

2) Algorithmic Setup: The algorithmic setup is of great importance for the quality of the estimated optical flow. Using the pointwise solution to the continuous problem we will estimate the flow vectors at all pixel positions. The outline of the algorithm is given in Algorithm 1. The basic setup consists of iteratively solving (5) and (6) in a coarse-to-fine pyramid scheme. We use $\ell_{\max} = 20$ pyramid levels with a downscaling factor of 0.83, and at each pyramid level we perform $w_{\max} = 50$ warps [19], where motion is re-estimated with the image I_1 warped to I_0 using the current motion estimate. The image pyramid is built by downsampling the original images using bilinear interpolation. Each image in the pyramid is given as the image, at one level lower, obtained by first smoothing with a Gaussian of standard deviation 0.4, and thereafter downsampling. When going from a coarser level to a finer, flows are upsampled using bilinear interpolation and the flow vectors are multiplied by the reciprocal of the downscaling factor to correctly fit the finer level. Throughout the experiments we have used $\theta = 0.2$, and the remaining parameters will be estimated adaptively for the given image sequence. For more information, see Section III-C.

Algorithm 1 Computation of TV-L^1 Optical Flow

Data: Two images I_{-1} and I_1
Result: Optical flow field u from I_{-1} to I_1
for $\ell = \ell_{\max}$ *to* 0 **do**
  //Pyramid levels
  Downsample the images I_{-1} and I_1 to current pyramid level
  for $w = 0$ *to* $w_{\max}$ **do**
    //Warping
    Compute v pointwise as the minimizer (7) of E_1 (5)
    for $i = 0$ *to* $i_{\max}$ **do**
      //Inner iterations
      Compute u as the minimizer of E_2 (6) (Section 3.2 in [24])
    for $f = 0$ *to* $f_{\max}$ **do**
      //Median filtering
      Apply a 3×3 median filter on u
  Upscale v and u to next pyramid level

B. Motion Compensated Frame Interpolation

If the two images I_{-1} and I_1 are keyframes, we are interested in estimating the in-between Wyner-Ziv frame I_0. Given the (forward) optical flow v_f, the simplest conceivable approach would be to assume that the true motion follows the estimated motion vectors linearly through I_0 and then fill in I_0 according to

$$I_0(\boldsymbol{x} + {}^1\!/_2 v_f(\boldsymbol{x})) = \frac{1}{2}(I_{-1}(\boldsymbol{x}) + I_1(\boldsymbol{x} + v_f(\boldsymbol{x}))). \quad (9)$$

However since $v(\boldsymbol{x})$ is a real valued vector, $\boldsymbol{x} + {}^1\!/_2 v_f(\boldsymbol{x})$ is typically not a pixel position. We solve this by temporally warping the flow to I_0 [17], [27], which is done by determining a new flow v_f^0 from I_0 to I_1 under the assumption that the motion vectors pass linearly through the Wyner-Ziv frame I_0. For every pixel position $\boldsymbol{x}$, v_f^0 is approximated by

$$v_f^0(\text{round}(\boldsymbol{x} + {}^1\!/_2 v_f(\boldsymbol{x}))) = {}^1\!/_2 v_f(\boldsymbol{x}) \quad (10)$$

where the round function rounds the argument to nearest pixel value in the domain. There are some drawbacks to this approach. First, if the area around $\boldsymbol{x}$ in I_{-1} is occluded in I_1, there will probably be multiple flow candidates assigned at the point round $(\boldsymbol{x} + {}^1\!/_2 v_f(\boldsymbol{x}))$. In the converse situation, i.e. dis-occlusion from I_{-1} to I_1 there may be pixels that are not hit by a flow vector, thus leaving holes in the flow. The first problem can easily be solved by choosing the candidate flow vector with the best data fidelity, i.e. the candidate v_f for which $\|I_1(\boldsymbol{x} + v_f(\boldsymbol{x})) - I_0(\boldsymbol{x})\|$ has the smallest value. For the problem of dis-occlusions the solution is not so simple. Here we will simply fill the holes in the flow field by an outside-in filling strategy. The same approach can of course be taken with the backward flow v_b (i.e. the flow from I_1 to I_{-1}), and as our final interpolated frame we will use the average

$$I_0(\boldsymbol{x}) = \frac{1}{2}\left(I_{-1}\left(\boldsymbol{x} + v_b^0(\boldsymbol{x})\right) + I_1\left(\boldsymbol{x} + v_f^0(\boldsymbol{x})\right)\right). \quad (11)$$

For later use we define a residual between the backward and forward warped frames in pixel domain

$$r_0(\boldsymbol{x}) = I_{-1}\left(\boldsymbol{x} + v_b^0(\boldsymbol{x})\right) - I_1\left(\boldsymbol{x} + v_f^0(\boldsymbol{x})\right). \quad (12)$$

Applying 4×4 DCT to I_0 (11) and r_0 (12) gives the side information and residual in transform domain.

C. Learning

This section describes the process in which some of the parameters of the optical flow algorithm are learned from the data. Optical flow algorithms often have a large number of parameters which are typically hand-tuned using benchmark data and then fixed. The perhaps most successful method for adaptive estimation is the optimal prediction principle by Zimmer et al. [22], where the prediction quality of the estimated motion field is used for estimating the data fidelity weight λ. Here we will present a scheme which, similarly to the optimal prediction principle, relies on the temporal correlation between the previously decoded Wyner-Ziv frame I_{-2} and the current I_0, such that the scheme does not need to consider future frames for prediction evaluation. The scheme is generic in the sense that, in principle, all algorithm parameters can be estimated in this process. As previously mentioned we have three free parameters, λ, $i_{\max}$ and $f_{\max}$, all of which are related to the smoothness of the estimated flow. The parameter λ controls the trade-off between data fidelity and regularity, a low value means higher weight to the total variation term which in turn means a smoother estimate, and vice versa. The parameter $i_{\max}$ determines the convergence of the solution of (6), and while it in principle should be high enough to guarantee convergence, we have found that varying values improves results. Finally the parameter $f_{\max}$ determines the number of times that a 3×3 median filter is applied to the flow u (in each warp). A median filter is a good way of removing strong outliers, e.g. caused by a bad data fit. Figure 3 shows the effect of these three parameters on the estimated motion. For the given frames, the motion blur and slight intensity shift means that too high a weight on data fidelity, Fig. 3(d), will cause motion artifacts. On the other hand choosing a λ-value that does not cause artifacts, Fig. 3(c), results in an estimate where the motion of the two rightmost players merges. By imposing intermediate median filter steps, the strong outliers that propagate these artifacts throughout the image pyramid are removed, and a higher lambda can be chosen. This can be seen in Fig. 3(e) where the motion field has few artifacts, and the motion of the two rightmost players is clearly separated. However, a median filtering is not always desirable as it may remove small-object motion.

1) GOP Size 2: For a GOP size of 2, we propose to estimate the free parameters as follows. Let a decoded frame be denoted $\widehat{I}$. The side information for I_0 is calculated based on $\widehat{I}_{-1}$ and $\widehat{I}_1$, and used to decode $\widehat{I}_0$. Validating against the reconstructed frame $\widehat{I}_0$, we will find the set of optical flow parameters for which the interpolation (11) has the lowest mean square error. These parameters will then be passed on and used for calculating the next frame, assuming that the type of motion in the next frame, and hence the parameters, are similar to the optimal choice for $\widehat{I}_0$.

For the first Wyner-Ziv frame we cannot learn from the previous, so we fix the parameters $\lambda = 70$, $i_{\max} = 5$ and $f_{\max} = 2$. From then on we will evaluate all combinations of $\lambda \in \{0, 10, \ldots, 130\}$, $i_{\max} = 5, 10, 15, 20$ and $f_{\max} = 1, 2$, resulting in 112 distinct flow fields. Figure 4 shows the optimal λ parameters for four test sequences. We see that the temporal correlation between the frames is more clear in the *Coastguard* and *Hall* sequences, than in *Foreman* and *Soccer* that contain stronger motion. But as is visible in Fig. 3, the choice of good λ values depends on both $i_{\max}$ and $f_{\max}$, and even for *Coastguard* and *Foreman*, we have found that the estimation procedure increases side information quality. We have made a CUDA C implementation for computing optical flow on graphics hardware. Using this we are able to compute a single optical flow in less than 200 ms on an single NVIDIA Tesla C2050 GPU. By taking advantage of the parallel nature of the optical flow computations, this further makes the OF learning process for each Wyner-Ziv frame feasible in a matter of seconds. At a slight cost in accuracy (in particular lowering the number of levels in the coarse-to-fine pyramid), the computation of optical flows in QCIF sequences can be done quite a bit faster than realtime [25].

2) Hierarchical GOP Size 4: Using a hierarchical GOP size 4, we have three Wyner-Ziv frames for which we need to generate side information. We proceed by decoding the middle frame first and thereafter use its reconstruction to find the optimal set of parameters, as for GOP size 2. The parameters $i_{\max}$ and $f_{\max}$ are then used for the optical flow based side information generation for the two remaining frames of the GOP. As the temporal distance has been halved, we simply increase the data fidelity weight λ (by a factor of 1.2). This scheme was found to outperform a scheme similar to the one presented in the previous section, while at the same time reducing the relative computational complexity by only testing the parameter sets for one out of three WZ frames in a GOP of size 4.

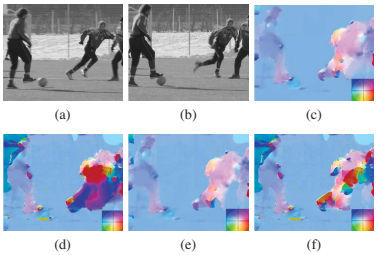


Fig. 3. Frames I_{-1} and I_1 from the *Soccer* sequence and corresponding color-coded motion fields [Fig. 3(c)–(f)], for different combinations of the parameters λ, $i_{\max}$, and $f_{\max}$. Unless otherwise mentioned, $i_{\max} = 5$ and $f_{\max} = 0$. (a) I_{-1}. (b) I_1. (c) $\lambda = 20$. (d) $\lambda = 100$. (e) $\lambda = 100$, $f_{\max} = 2$. (f) $\lambda = 100$, $i_{\max} = 20$.

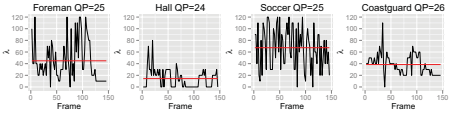


Fig. 4. Optimal values of the smoothness parameter λ for the four test sequences, along with the average λ value (line).

TABLE I
AVERAGE PSNR [dB] RESULTS FOR DIFFERENT SIDE INFORMATION GENERATION METHODS (GOP2)

Sequence	Extra	OBMC	OF	OF(learning)
Foreman, QP = 25	25.20	29.26	29.28	29.63
Hall, QP = 24	33.24	36.46	32.28	35.71
Soccer, QP = 25	19.26	21.30	22.43	22.93
Coast, QP = 26	28.55	31.83	30.92	30.99

3) Side Information Generation Evaluation: The performance of the proposed optical flow learning based frame interpolation scheme, called OF(learning) is evaluated for GOP2 and compared with the optical flow interpolation [28], the block based frame interpolation [7] and the extrapolation described in [28], named OF, OBMC, and Extra, respectively. The quality of interpolated frames is measured by average Peak Signal-to-Noise Ratio (PSNR) over the set of test sequences, *Foreman*, *Soccer*, *Coastguard* and *Hall* at 15 frames per second, QCIF format, and GOP size 2. Key frames are coded with H.264/AVC intra and QPs are chosen as in [5]. In Table I, it can be seen that the OBMC based frame interpolation method gives the best performance on *Hall* and *Coast*. However, the optical flow based frame interpolation outperforms the OBMC scheme [7] on the high motion sequences, especially *Soccer*. The proposed OF(learning) method outperforms the OF method on all test sequences. In addition, for the OBMC scheme, increasing the search range was evaluated. However, the results only improved slightly by 0.05 dB on average when increasing the search range. The proposed OF(learning) was still better on the high motion sequences. Later we shall combine OF side information generation with OBMC to improve the performance of TDWZ coding.

IV. NOISE RESIDUAL LEARNING FOR ADAPTIVE NOISE MODEL

We consider the difference between the original Wyner-Ziv frame X and the side information frame Y. The residual difference, Z, between the transformed coefficients of the WZ frame and the interpolated frame will be modeled by a Laplacian distribution with probability density function $f(z) = (\alpha/2)\exp(-\alpha|z|)$ with variance $\sigma^2 = 2/\alpha^2$.

Rate distortion bounds for simple source models may be derived [29]. Assuming quadratic distortion D and a memoryless source with variance σ^2 and entropy power Q, the upper and lower rate distortion bounds are [29]

$$\frac{1}{2}\log\frac{Q}{D} \le \Re(D) \le \frac{1}{2}\log\frac{\sigma^2}{D} \tag{13}$$

where $\Re(D)$ denotes the rate at distortion D, the entropy power is $Q = (1/2\pi e)\exp(2\mathfrak{h}(Z))$, and $\mathfrak{h}(Z) = \mathrm{E}[-\log f(Z)]$ denotes the differential entropy of the source Z, where E[.] denotes the expectation operator. For the Laplacian distribution, the entropy power is $Q = (e/\pi)\sigma^2$ [29]. Inserting in (13) gives

$$\frac{1}{2}\log\frac{e}{\pi}\frac{\sigma^2}{D} \le \Re(D) \le \frac{1}{2}\log\frac{\sigma^2}{D}. \tag{14}$$

The bounds in (13) may be decreased if the outputs of a given source are split into a number of subsets having different variance and entropy (assuming we also know which subset each sample belongs to). This may be shown based on the concavity of the log and entropy functions, applying Jensen's inequality, $f(\mathrm{E}[Z]) \geqslant \mathrm{E}[f(Z)]$, to $\log\sigma^2$ of the upper bound and the entropy term $-f(Z)\log f(Z)$ of $\mathfrak{h}(Z)$ in the lower bound (13). As a result, for a given distortion level, the $\Re(D)$ bounds (13) over all clusters are reduced. Below we will describe the process of using clustering for DVC noise modeling.

A. Adaptive Noise Model Using Clustering of DCT Blocks

The decoder must estimate the statistics of the residual without access to the original frame X. Consistent with the remarks above, it was noted in [14] that the variance of the residual frame based on an estimated residual is higher than the expected variance over the sub-sets. This motivates reducing the codelength by clustering into sub-sets, which are processed using different parameter values. The techniques proposed in this paper are based on an initial work on the adaptive noise model using clustering of DCT blocks [14]. The adaptive noise model considers the (4×4 DCT) transformed residual of frequency bands in a block as components of a (feature) vector.

Let R_h be the residual frame in the transform domain using a frame interpolation scheme h. R_h is used to calculate the parameter of the Laplacian noise distribution $f_{X|Y_h}$. The value of the Laplacian parameter expresses the reliability of the corresponding estimated side information frame. R_h is initialized at the decoder based on the difference between matching blocks of the reference images [7]. Let R_{hk} denote block k out of the N 4×4 blocks in the residual frame R_h, $1 \le k \le N$. Each block R_{hk}, considered as a feature vector, contains 16 frequencies given by the transformed residual coefficients. Consider block k of band l and let R_{hk}^l and $\widehat{R}_{hk}^l$ ($1 \le l \le 16$) denote the initial coefficient of the residual and a refined coefficient based on the partially decoded information, respectively. The feature vector of each block $R_{hk} = (\widehat{R}_{hk}^1, \widehat{R}_{hk}^2, \ldots, \widehat{R}_{hk}^{l-1}, R_{hk}^l, R_{hk}^{l+1}, \ldots, R_{hk}^{16})$ belongs to the updated residual based on the successfully decoded bands (up to band $l-1$) before decoding band l. This feature vector is classified into one of M clusters, within which an estimate of the noise parameter is calculated. Thus, using clustering of DCT blocks, an adaptive noise model creates M noise parameters, α, one for each cluster.

B. Noise Model B

An extended noise model, which we denote Noise Model B, is obtained by adaptively combining the cluster level noise model in Section IV-A with the noise model in [7]. The clustering technique in [14] was updated at coefficient level and is here extended by updating at bitplane level. A noise residue refinement is exploited at bitplane level and integrated in the DVC scheme in [7]. The refinement is carried out once a bitplane is successfully decoded. The model consists of 4 steps as follows.

Step 1. Clustering of DCT Blocks: Our block clustering algorithm is operating on a set of N feature vectors R_{hk}. This set is separated into M subsets or clusters by using Fuzzy-C means clustering [30]. (The algorithm is configured with the fuzzification degree equal 2 and the predefined termination $\varepsilon = 0.0001$ as in [14].) For block k belonging to cluster j, let $R^l_{hkj} = R^l_{hk}$ denote the coefficients of feature vectors and α^l_{hj} denote the Laplacian noise distribution parameter of cluster j $(1 \le j \le M)$ containing N_j elements of band l, where $\sum_j N_j = N$. Figure 5 illustrates an example of clustering of DCT blocks for the *Soccer* sequence where OBMC was used to generate the SI frame (Fig. 5(b)). The residual frame in the transform domain R_h (Fig. 5(c)) is estimated at the decoder side before decoding the first (DC) band, $l = 1$. Thereafter the residual is classified into 3 clusters ($M = 3$) (Figs. 5(d)-5(f)).

Step 2. Noise Parameter Estimation: In band l, a noise parameter, α^l_{hj}, is obtained for each cluster j of the band based on the N_j observations within the cluster. We estimate this Laplacian parameter, α^l_{hj}, based on the variance σ^{l2}_{hj} by

$$\alpha^l_{hj} = \sqrt{2}/\sigma^l_{hj} \tag{15}$$

where $\sigma^l_{hj} = \sqrt{\mathrm{E}[|R^l_{hkj}|^2] - \mathrm{E}[|R^l_{hkj}|]^2}$. As a result, a noise parameter is estimated for each of the M clusters in a given band l.

Step 3. Updating Feature Vectors: The bands are decoded in a zig-zag order starting from DC and traversing the other (AC) coefficients, $l > 1$, following the order in [7]. Whenever a bitplane of band l is successfully decoded, the coefficients of the band are partially reconstructed and the set of feature vectors is now updated. Thereafter, the set of updated feature vectors is used to refine these vectors by Step 4 below. When all bitplanes are successfully decoded, band l is completely decoded. Subsequently, the set of feature vectors is updated as $R_{hk} = (\widehat{R}^1_{hk}, \widehat{R}^2_{hk}, \ldots, \widehat{R}^{l-1}_{hk}, \widehat{R}^l_{hk}, R^{l+1}_{hk}, \ldots, R^{16}_{hk})$ before decoding band $l + 1$. This set of updated feature vectors is further refined by Step 4 (below) and thereafter α^{l+1}_{hj} is updated for the next band $l + 1$ to be decoded. When all bands are successfully decoded, the process is completed.

Step 4. Refining Feature Vectors Using Neighbors: To take advantage of the correlation between the DCT coefficients of the residual of neighbor blocks within each band, a refinement of residuals is proposed. This technique uses neighboring residual coefficients along with the estimated noise parameters. Specifically, Noise Model B refines R^l_{hkj} based on α^l_{hj} and the 8-neighbor residual coefficients, indexed by s and denoted R^l_{hks}. Using the current coefficient R^l_{hk0} and the 8-neighbors, R^l_{hks} with $1 \le s \le 8$, a refined R^{*l}_{hkj} $(= R^{*l}_{hk}$ for k in cluster $j)$ is obtained by weighing the neighborhood coefficients as

$$R^{*l}_{hkj} = \sum_{s=0}^{8} \left(\frac{\exp(-\alpha^l_{hj}|R^l_{hkj} - R^l_{hks}|)}{\sum_{t=0}^{8} \exp(-\alpha^l_{hj}|R^l_{hkj} - R^l_{hkt}|)} \right) R^l_{hks}. \tag{16}$$

Also assuming a Laplacian distribution for the difference of a coefficient and its neighbors, the weights (16) may be seen as likelihood values and the denominator normalizes these.

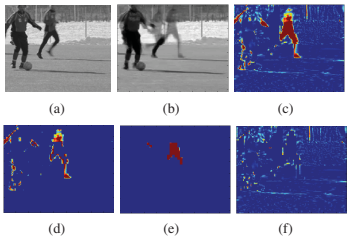


Fig. 5. Example of clustering *Soccer* frame no. 88 into three clusters. (a) Original frame. (b) SI frame. (c) Residual frame. (d) Cluster 1. (e) Cluster 2. (f) Cluster 3.

These refined residuals are used in the set of N refined feature vectors, $R^*_{hk} = (\widehat{R}^1_{hk}, \widehat{R}^2_{hk}, \ldots, \widehat{R}^{l-1}_{hk}, R^{*l}_{hk}, R^{l+1}_{hk}, \ldots, R^{16}_{hk})$ used for decoding band l. The set is reclassified again by going back to Step 1 and thereafter updating the noise parameter following Step 2 above. Consequently, refined noise parameters α^{*l}_{hj} are obtained using (15) based on the observations within the current band for each refined cluster j. The set of α^{*l}_{hj} parameters is denoted by α_1 and together with the set α_0 from [7], they constitute the set of estimates provided by Noise Model B. The resulting coding is referred to as Clustering TDWZ.

C. Noise Residual Learning

1) Noise Residual Learning Using Previously Decoded Residual Frames: This subsection extends Noise Model B above by using the previously WZ decoded residual frames to influence the noise distribution of the current frame. A window of previously decoded WZ frames are used to create decoded residual frames corresponding to the WZ decoded frames. The motivation is that the noise distributions based on previously decoded frames are available at the decoder and may be similar to the noise distribution of the current frame. To take advantage of both the previously decoded noise distributions and the estimated current noise distribution, the residuals based on previously decoded frames are used together with the current residual frame to form a larger set of data. This set is classified into clusters to estimate noise parameters for each cluster of the residual frame considered.

Let W be the window size specifying the number of previously decoded WZ frames for the learning process. Let $\widehat{R}_{h(2n-2W)}, \ldots, \widehat{R}_{h(2n-2)}$ denote residuals based on previously decoded frames and $R_{h(2n)}$ denote the current residual coefficient frame at time $2n$. Let $\widehat{R}_{h(2n-2W)k}, \ldots, \widehat{R}_{h(2n-2)k}, R_{h(2n)k}$ denote block k, $1 \le k \le N$, of N 4×4 blocks of $\widehat{R}_{h(2n-2W)}, \ldots, \widehat{R}_{h(2n-2)}, R_{h(2n)}$. For each of the residuals based on previously decoded frames, consider a set of N feature vectors $\widehat{R}_{h(2n-2\omega)k}$ with $1 \le \omega \le W$, where $\widehat{R}_{h(2n-2\omega)k} = (\widehat{R}^1_{h(2n-2\omega)k}, \widehat{R}^2_{h(2n-2\omega)k}, \ldots, \widehat{R}^{16}_{h(2n-2\omega)k})$ is given by the residuals of decoded bands. For the current residual frame $R_{h(2n)}$, $R_{h(2n)k} = (\widehat{R}^1_{h(2n)k}, \ldots, \widehat{R}^{l-1}_{h(2n)k}, R^l_{h(2n)k}, \ldots, R^{16}_{h(2n)k})$ is the updated residual based on the successfully decoded bands (up to band $l - 1$) before decoding band l.

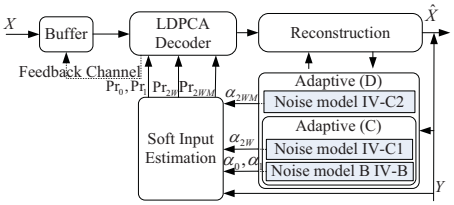


Fig. 6. TDWZ with adaptive noise model using clustering and noise residual learning.

Consider W sets, $S_{h\omega}$, of feature vectors where each set is created by combining N feature vectors $\widehat{R}_{h(2n-2\omega)k}$ of a previous frame with N feature vectors $R_{h(2n)k}$ of the current frame,

$$S_{h\omega} = \{R_{h(2n)}, \widehat{R}_{h(2n-2\omega)}\}. \tag{17}$$

Each set $S_{h\omega}$ is classified into M clusters by using Fuzzy C-means clustering [30]. Thereafter noise parameters $\alpha^l_{h\omega j}$ are obtained based on the observations for each cluster j of band l of set $S_{h\omega}$. As a result, there are W sets of noise parameters for decoding band l for each cluster j, $\alpha^l_{h\omega j}$, $1 \le \omega \le W$. The resulting adaptive noise model, denoted by Adaptive (C) and shown in Fig. 6, adaptively estimates the noise distribution by creating W different noise parameters $\alpha_{2\omega}$. Together with α_0, α_1 obtained from Noise Model B in Section IV-B, $\alpha_{2\omega}$ provide multiple inputs to the Soft Input Estimation block.

2) Adapting the Number of Clusters for Noise Modeling: This part extends the noise residual estimation by selecting the number of clusters, $m \le M$, giving the best decoding, i.e. optimizing the model order. The statistical characteristics of the noise distribution may change from region to region, and over time when decoding. One reason being, that the noise distribution may not be estimated properly in regions containing moving objects. It may improve the noise modeling, if the noise residual R_h is adaptively modeled using a variable number of noise distributions. A dynamic mechanism is carried out to determine the optimal number of candidate distributions within each frame once a bitplane is successfully decoded.

For each cluster j, m Laplacian distributions, for $1 \le m \le M$, are estimated. For each set in (17), we apply estimation by (15) to obtain the noise parameters $\alpha^l_{h\omega j}$, $1 \le \omega \le W$, based on the observations for each cluster j, $1 \le j \le m$, of band l of the set $S_{h\omega}$. This results in the set of distributions,

$$D_{hm} = \{\alpha^l_{h\omega mj}\},\ 1 \le \omega \le W,\ 1 \le j \le m \tag{18}$$

where $\alpha^l_{h\omega mj}$ is the noise parameter estimated for band l of set $S_{h\omega}$ of the distribution set D_{hm}.

The noise parameters $\alpha^l_{h\omega mj}$ of the $W \times M$ candidates are used as candidates for decoding band l for each cluster j. The resulting noise model, called Adaptive (D) (Fig. 6), adaptively estimates the noise distribution by creating $W \times M$ candidate noise parameters $\alpha_{2\omega m}$, along with $\alpha_{2\omega}, \alpha_0, \alpha_1$ obtained from Adaptive (C) in Section IV-C.1 as input to the Soft Input Estimation block.

3) Multiple Input LDPCA Decoding: In this subsection, multiple input LDPCA decoding is introduced using multiple soft input candidates when decoding. The multiple input LDPCA decoder tries to decode using each candidate soft side information and then selects the soft side information which converges first during decoding for each bitplane. (Convergence is confirmed using a CRC check.) This way, the decoder adaptively selects the best soft input for decoding. Thereafter, the selected noise parameter set for each bitplane is also used for the minimum mean squared error reconstruction process [8]. It can be noted that the computational cost of the LDPCA decoding is increased. In the worst case, the LDPCA decoding will try to decode using all inputs. In Fig. 6, the multiple input LDPCA decoder is used to decode with the candidates denoted $\Pr_0, \Pr_1, \Pr_{2\omega}$, and $\Pr_{2\omega m}$. These candidates are calculated by the Soft Input Estimation using the noise parameters from the Adaptive (D) model. In particular, $\Pr_0, \Pr_1$ are soft inputs calculated based on α_0, α_1, $\Pr_{2\omega}$ based on $\alpha_{2\omega}$, and $\Pr_{2\omega m}$ based on $\alpha_{2\omega m}$. The multiple input LDPCA decoding, as well as the learning technique, is carried out until all bitplanes (of the given quantization level) are successfully decoded. Applying multiple input LDPCA decoding based on the parameters of the Adaptive (D) noise model is referred to as Clustering(learning) TDWZ.

V. TDWZ Video With Side Information and Noise Learning

The quality of soft-input information plays a key role in terms of overall RD performance of TDWZ video coding. The quality of the reconstructed frame is highly dependent on the accuracy of the estimated noise distribution $f_{X|Y_h}$. The soft-input Pr is defined as the conditional probability of each bit b_i being equal to 0 or 1, and denoted $\Pr = \mathrm{P}(b_i|y_h, b^-; f_{X|Y_h})$, where y_h denotes the corresponding estimated side information value in the transform domain for bit b_i and b^- is the information from the previously decoded bitplanes. The probability is obtained by marginalizing the estimated conditional probability density function $f_{X|Y_h}$ for the coefficient, which b_i is part of. The essential aspects to improve the coding efficiency of TDWZ video are the quality of the soft-input information fed into the LDPCA decoder and the accuracy of the noise distribution for frame reconstruction.

A. Multihypothesis-Based Wyner–Ziv Decoding

To address these issues, multiple input LDPCA decoding (Section IV-C.3) is used. The Wyner-Ziv encoder is not changed. The basic idea is to generate H (> 1) different side information frames Y_h, $h \in [1, H]$, at the decoder for each Wyner-Ziv frame. Each side information frame is considered as an observation of the original Wyner-Ziv frame X with a different amount of noise. The architecture of the proposed Wyner-Ziv video decoder with an example of two side information generation schemes [28] ($H = 2$) is presented in Fig. 7. In principle, there can be any number of competitive side information generation schemes at the proposed Wyner-Ziv decoder. The two interpolation methods in Fig. 7 are the OBMC based frame interpolation described

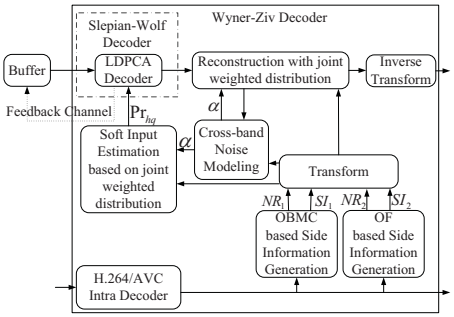


Fig. 7. Architecture of multihypothesis TDWZ video codec based on two frame interpolation schemes.

in [7] and the OF based frame interpolation method described in Section III. As shown in Fig. 7, the Side Information Generations generate the side information frames SI_1, SI_2 and the noise residual frames NR_1, NR_2, using OBMC [7] and OF techniques, respectively. SI_2 and NR_2 are generated by first applying OF based Side Information Generation and thereafter 4×4 DCT to I_0 (11) and r_0 (12), respectively. After transformation, each side information generation scheme not only creates an estimate of the Wyner-Ziv frame, Y_h, but also an estimated noise residue frame R_h. R_h is used to estimate the noise between the Wyner-Ziv frame X and its estimated side information frame Y_h. Here based on R_h and Y_h, the coefficient level noise model [7] is used. Each transform coefficient in a given band l is assigned an estimated Laplacian distribution parameter α_h^l.

Using Laplacian parameters based on different calculations of Y_h, multiple soft-inputs are calculated based on a weighted joint distribution. All the hypotheses of soft-input are fed into the multiple input LDPCA decoder (Section IV-C.3). Based on the estimated noise distribution $f_{X|Y_h}$ for each individual side information observation Y_h, a joint weighted distribution F_q is defined as

$$F_q = \sum_{h=1}^{H} u_{hq} f_{X|Y_h} \tag{19}$$

where q, $q \in [1, C]$, denotes the index of a candidate joint weighted distribution, C is the total number of candidate joint distributions, and u_{hq} denotes the qth predefined weight on side information h, $h \in [1, H]$, u_{hq} are predefined weights, $u_{hq} \geqslant 0$ and $\sum_{h=1}^{H} u_{hq} = 1$. (For the example shown in Fig. 7, $H = 2$, $C = 6$).

The frame interpolation schemes, using OBMC and OF, employed in this paper give different results on the different test sequences as shown in Table I. The OBMC and OF techniques may provide complementary results for each frame and thus, compensate each other's weaknesses frame by frame and even bitplane by bitplane. We consider a multi-hypothesis TDWZ video codec with two (or more) frame interpolation schemes based on either the OBMC or the OF scheme. Without loss of generality, assume that scheme h is now considered the basic scheme. The soft input calculation is only based on the joint weighted distribution within a specific unreliable region specified by the set Λ_h. Outside of the region Λ_h, the side information is given by the basic scheme h. The values of the Laplacian parameters may express the reliability of the corresponding side information frame. Therefore a set of Λ_h values for each single side information estimation Y_h in band l is determined by evaluating the individual Laplacian parameters and their corresponding mean value by

$$\Lambda_h = \{k | \alpha_h^l(k) < \bar{\alpha}_h^l\} \tag{20}$$

where $\alpha_h^l(k)$ is the Laplacian parameter of side information Y_h at the kth coefficient in band l and $\bar{\alpha}_h^l$ is the mean of all noise parameters in a given band l. Thus Λ_h (20) determines a map of coefficients whose noise parameters are potentially less reliable, as they are smaller than the mean value $\bar{\alpha}_h^l$. The unreliable region Ω, which will be processed differently, is defined as a union of the sets Λ_h,

$$\Omega = \bigcup_{h=1}^{H} \Lambda_h. \tag{21}$$

The multi-hypothesis soft-inputs using the Y_h as basic scheme are given by

$$\mathrm{Pr}_{hq} = \begin{cases} \mathrm{P}(b_i | y_h, b^-; f_{X|Y_h}) & \text{if } i \notin \Omega \\ \mathrm{P}(b_i | y_1, \ldots, y_H, b^-; F_q) & \text{if } i \in \Omega \end{cases} \tag{22}$$

where Pr_{hq} is the qth candidate soft-input fed into LDPCA decoder, b_i denotes the ith bit in the current bitplane, and $y_1, \ldots, y_H$ denote different side information values in the transform domain based on diverse side information generation schemes. Again the conditional probability of b_i is obtained by marginalizing the estimated noise distribution $f_{X|Y_h}$ ($i \notin \Omega$) or F_q ($i \in \Omega$). We use the cross-band noise model [7] to calculate $f_{X|Y_h}$ in (19) and (22). The resulting parameter set is denoted α_{hCB}.

In order to evaluate the quality of the side information, we calculate an Ideal Code Length (ICL) [7], which measures the number of bits required by applying ideal (arithmetic) coding to the given soft-input values if a (non-distributed) encoder would calculate the same soft-input values. $\mathrm{Pr}_{hq}(b_i)$ (22) is calculated by reading b_i as the bits after decoding. The code length, $\mathcal{L}$, for one bitplane is calculated as

$$\mathcal{L} = \sum_{i=1}^{N} -\log \mathrm{Pr}_{hq}(b_i). \tag{23}$$

The ICL is obtained as the sum over all bitplanes. This is equivalent to a log-likelihood measure of the coded coefficients.

All the soft-input hypotheses, Pr_{hq}, $q \in [1, C]$ which are calculated by (22) are fed into the multiple input LDPCA decoder as in Section IV-C.3. The first converging soft-input is chosen thus reducing the rate of LDPCA decoding. Subsequently, using the selected soft-input, the corresponding joint weighted distribution F_q, $q \in [1, C]$, in the unreliable region Ω is determined. Using the selected joint weighted distribution, F_q, the minimum mean-square error (mmse)

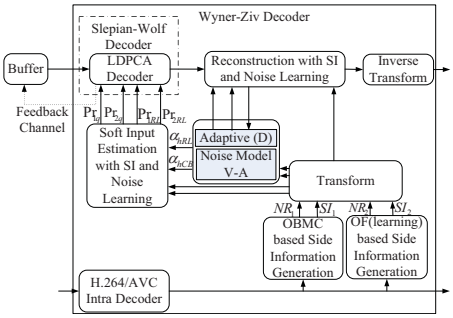


Fig. 8. TDWZ with side information and noise learning (SING 2SI).

reconstructed value, x', in the unreliable region Ω is obtained as a generalization of the mmse expression in [8]

$$x' = \mathrm{E}[x|x \in [L,U), y_1, \ldots, y_H] = \frac{\int_L^U x F_q(x)dx}{\int_L^U F_q(x)dx} = \frac{\sum_{h=1}^{H}\int_L^U x u_{hq} f_{X|Y_h}(x)dx}{\sum_{h=1}^{H}\int_L^U u_{hq} f_{X|Y_h}(x)dx} \quad (24)$$

where $[L, U)$ are decoded quantization intervals, F_q is the joint weighted distribution (19) selected by the LDPCA decoding. The reconstructed value outside the Ω region is calculated following the single side information reconstruction technique based on Y_h as in [8], i.e. for $H = 1$ in (24).

B. Side Information and Noise Learning Using Multiple-Hypothesis and Adaptive Noise Modeling

To take advantage of both side information learning (Section V-A) including optical flow (Section III) and noise learning using clustering (Section IV), a TDWZ scheme with **S**ide **I**nformation and **N**oise Learnin**G**, called **SING**(2SI), is proposed. The basic elements of the SING codec are depicted in Fig. 8. They consist of OBMC and OF(learning) based side information generations, a noise model using the residual learning as in Section IV-C, the soft-input estimation, the reconstruction using side information and noise learning (Section V-A) and multiple input LDPCA decoding. First, the Side Information Generations generate the noise residual frames NR_1, NR_2 and the side information frames, SI_1, SI_2, using OBMC [7] and OF(learning), where SI_2 and NR_2 are generated, as in Sec. V-A, by the OF(learning) based Side Information Generation using (11) and (12), respectively. These are transformed and input to the noise models. For each side information scheme h, noise parameters α_{hRL} are calculated using the Adaptive (D) model (Section IV-C) and parameters α_{hCB} are calculated using multi-hypothesis (Section V-A) combined with the cross-band estimate [7] for $f_{X|Y_h}$.

1) Soft Input Estimation With SI and Noise Learning: Based on the transformed side information frames and the noise parameters, the soft-inputs Pr_{1q}, Pr_{2q}, and Pr_{1RL}, Pr_{2RL} are calculated. Pr_{1q} and Pr_{2q} are calculated by (22) based on the cross-band noise and multi-hypothesis techniques with two OBMC and OF(learning) side information generations as described in Section V-A. Pr_{1q} are soft-inputs with the OBMC frame interpolation as basic scheme and Pr_{2q} are soft-inputs with the OF(learning) frame interpolation as basic scheme. Pr_{1RL}, Pr_{2RL} are obtained by applying the Adaptive (D) model in Section IV-C to each side information generation scheme, here OBMC and OF(learning). All soft-inputs are fed into the multiple input LDPCA decoder as described in Section IV-C.3. The soft-input which converges first is selected for LDPCA decoding.

2) Reconstruction With SI and Noise Learning: Decided by the selected candidate, the corresponding weighted joint distribution of multi-hypothesis or the corresponding input to the Adaptive (D) noise model is chosen for reconstruction. For instance, if Pr_{13} is the best candidate, the corresponding weighted joint distribution is F_3 and the multi-hypothesis based scheme is the OBMC based method ($h = 1$ in (22)). Consequently, the weighted joint distribution F_3, the multi-hypothesis OBMC based scheme, and the corresponding noise parameter α_{hCB} are used for reconstruction by (24). As another example, if Pr_{2RL} is chosen as the best candidate, α_{2RL} from the Adaptive (D) model, NR_2 and SI_2 from the OF(learning) side information generation are used for the mean squared error reconstruction [8] (24) in the reliable region. In the unreliable region Ω, the reconstruction is based on the multi-hypothesis reconstruction corresponding to the basic frame interpolation scheme in (24). As we do not have a winner F_q for the joint weighted distribution in this case, the F_q for reconstruction is determined by the corresponding soft-input that has the smallest ICL measured on the decoded data by (23) among the soft-input hypotheses. By this approach, the reconstruction takes advantage of different side information generation techniques in the unliable region to achieve a better quality of the reconstructed frames.

VI. Performance Evaluation

The rate-distortion (RD) performance of the proposed techniques are evaluated for the test sequences (149 frames of) *Foreman, Hall Monitor, Soccer,* and *Coastguard* with 15Hz frame rate and QCIF format. The GOP size is 2, where odd frames are coded as key frames using H.264/AVC Intra and even frames are coded using Wyner-Ziv coding. Eight RD points are considered corresponding to eight 4×4 quantization matrices [5]. H.264/AVC Intra is here given by the intra coding mode of the H.264/AVC reference codec JM 9.5 [31] in main profile. The parameters for H.264/AVC Intra are set as by DISCOVER [5] and QP values are set to those used for the key frames in the Wyner-Ziv video coding in the DISCOVER codec [5]. It can be noted that only luminance component of each frame is evaluated. In this paper, the number of candidate distributions in (19) is constrained to $C = 6$, which is an adequate number of candidates to improve performance. For the case $H = 2$, using side information frames generated by OBMC and OF(learning), and $C = 6$, the weighting parameters (19) used are $u_{1q} = \{1; 0.8; 0.6; 0.4; 0.2; 0\}$ and

TABLE II
BJØNTEGAARD RELATIVE BIT-RATE SAVINGS (%) OVER DISCOVER FOR WZ FRAMES (QCIF, 15 Hz, GOP2)

Sequence	Cross-band [7]	Clustering	Clustering(learning)	MH(2SI)	MH(learning 2SI)	MH(learning 3SI)	SING(2SI)	SING(3SI)
Foreman	14.0	17.7	21.6	27.0	27.2	32.6	35.1	40.1
Hall	8.3	14.3	21.0	13.3	12.2	13.3	21.6	19.5
Soccer	26.0	30.8	34.5	41.2	46.0	49.2	61.1	62.5
Coast	11.6	17.5	21.1	17.4	17.9	19.9	24.9	25.8
Average	15.0	20.1	24.6	24.7	25.8	28.7	35.7	37.0

TABLE III
BJØNTEGAARD PSNR IMPROVEMENT (dB) OVER DISCOVER FOR WZ FRAMES (QCIF, 15 Hz, GOP2)

Sequence	Cross-band [7]	Clustering	Clustering(learning)	MH(2SI)	MH(learning 2SI)	MH(learning 3SI)	SING(2SI)	SING(3SI)
Foreman	0.633	0.798	0.974	1.177	1.181	1.398	1.492	1.659
Hall	0.370	0.633	0.903	0.575	0.531	0.581	0.919	0.846
Soccer	1.305	1.521	1.677	1.921	2.088	2.216	2.649	2.690
Coast	0.352	0.530	0.637	0.526	0.540	0.600	0.741	0.762
Average	0.665	0.872	1.047	1.050	1.085	1.199	1.450	1.489

TABLE IV
BJØNTEGAARD RELATIVE BIT-RATE SAVINGS (%) OVER DISCOVER FOR ALL FRAMES (QCIF, 15 Hz, GOP2)

Sequence	Cross-band [7]	Clustering	Clustering(learning)	MH(2SI)	MH(learning 2SI)	MH(learning 3SI)	SING(2SI)	SING(3SI)
Foreman	6.0	7.5	9.0	11.0	11.0	13.0	13.8	15.6
Hall	2.6	3.9	5.4	3.8	3.6	3.8	5.5	4.8
Soccer	14.4	17.2	19.4	22.6	25.1	26.6	32.6	33.2
Coast	3.9	5.6	6.4	5.5	5.7	6.2	7.4	7.6
Average	6.7	8.6	10.0	10.7	11.3	12.4	14.8	15.3

TABLE V
BJØNTEGAARD PSNR IMPROVEMENT (dB) OVER DISCOVER FOR ALL FRAMES (QCIF, 15 Hz, GOP2)

Sequence	Cross-band [7]	Clustering	Clustering(learning)	MH(2SI)	MH(learning 2SI)	MH(learning 3SI)	SING(2SI)	SING(3SI)
Foreman	0.335	0.417	0.502	0.606	0.609	0.717	0.762	0.845
Hall	0.187	0.290	0.396	0.276	0.260	0.275	0.400	0.354
Soccer	0.723	0.852	0.950	1.087	1.186	1.255	1.501	1.525
Coast	0.186	0.265	0.306	0.261	0.268	0.296	0.354	0.363
Average	0.358	0.456	0.538	0.558	0.581	0.636	0.754	0.772

$u_{2q} = 1 - u_{1q},\ q \in [1, 6]$. For the case $H = 3$ and $C = 6$, the weighting parameters u_{hq} used are predefined as: $u_{1q} = \{1; 0; 1/2; 1/2; 0; 1/3\}$, $u_{2q} = \{0; 1; 1/2; 0; 1/2; 1/3\}$, and extrapolation [9] $u_{3q} = \{0; 0; 0; 1/2; 1/2; 1/3\}$. For $H = 3$, these parameters provide a uniform weighting of one, two, or three candidates. The proposed Clustering(learning) scheme (Section IV-C) uses a window size of $W = 6$ of previously decoded residual frames and a maximum number of clusters $M = 10$ (18), which is large enough to utilize the meaningful past information and adapt to an efficient number of noise distributions.

Tables II-V report RD performance of the combined schemes (Section V-B) SING(2SI) using OBMC and OF(learning) as well as SING(3SI), which additionally uses side information generation based on extrapolation [9]. Tables II-V present the relative average bitrate savings and equivalently the average PSNR improvements (using the Bjøntegaard metric [32] and fitting a curve through the 8 RD points measured) over the DISCOVER codec for WZ frames and overall frames. The results are also compared to the DVC scheme in [7] called Cross-band. The SING codecs are based on combining the clustering and multi-hypothesis techniques, which are also evaluated individually. The noise model in Section IV-B integrated in the DVC scheme in [7], is named Clustering. The noise model proposed in Section IV-C integrated in DVC scheme in [7] is named Clustering(learning). Both of these are based on the OBMC side information. The proposed multi-hypothesis TDWZ codecs combining OBMC with OF and OF(learning) techniques mentioned in Section V-A are called MH(2SI) and MH(learning 2SI), respectively. MH(learning 3SI) refers to the additional use of extrapolation, respectively. Compared to DISCOVER, the average bitrate saving for the combined scheme SING(3SI) model is overall (average Bjøntegaard) 37% and 15% better on WZ frames and all frames, respectively. The performance improvement is 62.5% and 33.2% (or equivalently the average improvement in PSNR is 2.69 dB and 1.53 dB) for WZ frames and overall frames, respectively, for the difficult *Soccer* sequence.

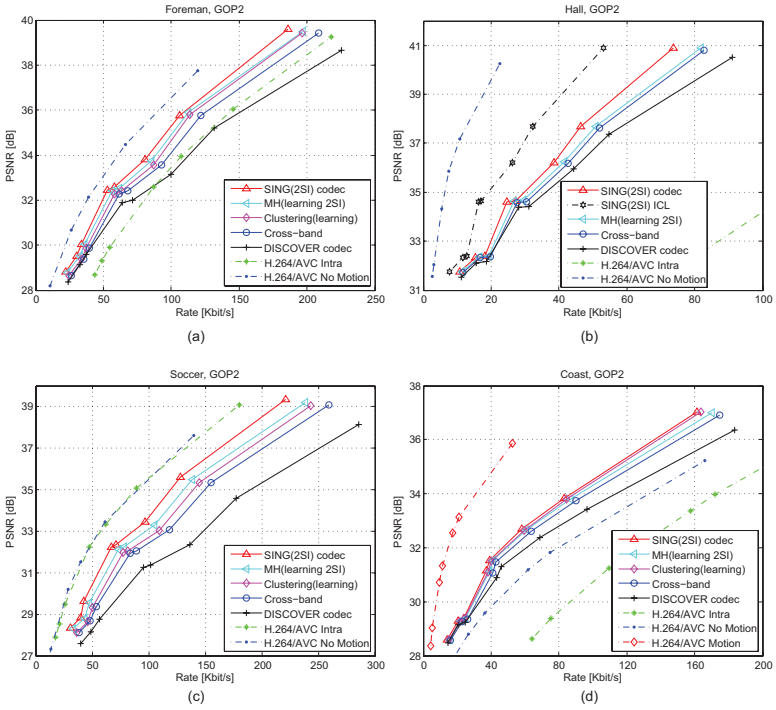


Fig. 9. PSNR versus rate for the proposed SING(2SI) codec for WZ frames (QCIF, 15 Hz, GOP2). (a) *Foreman*. (b) *Hall*. (c) *Soccer*. (d) *Coast*.

Compared to the DVC scheme in [7] denoted Cross-band, a bit-rate saving (Bjøntegaard) of 36.5% is observed for *Soccer* on the WZ frames. Looking at Table II, we see that both Clustering(learning) and MH(learning) introducing OF improve the average bit-rate savings to about 25% starting from the 15% savings of the baseline Cross-band codec [7]. Further, the Clustering and MH combine well in SING(2SI) for a 36% saving. Looking at the individual sequences, we see that using OF in MH improves performance most for high motion sequences *Foreman* and especially *Soccer*, whereas Clustering(learning) achieves better results on the low motion sequences as *Coast* and especially *Hall Monitor*. Our results may be compared with a few GOP2 results in [6], [13], [15], [33]. The TRACE method [33] reports 1.6% bit-rate saving for *Foreman* (at 30Hz) compared with [12]. The following comparisons are evaluated for QCIF and 15Hz frame rate at 400 Kb/s. Compared to DISCOVER, the results in [6] show an improvement of 0.4 dB for *Foreman* and 0.7 dB for *Soccer*. Improvements of 0.5 dB for *Foreman* and 0.1dB degradation for *Soccer* are reported [13]. More recently, the scheme in [15] shows an improvement 0.4 dB for *Foreman* and 0.5 dB for *Soccer*. At 400 Kb/s, improvements compared with DISCOVER of 1.0 dB for *Foreman* (Fig. 10(a)) and 1.4 dB for *Soccer* (Fig. 10(c)) are achieved by Clustering (learning). Specifically, the improvements of MH(learning 2SI) including OF(learning) are robust for the high motion sequences as *Soccer*. The proposed SING(2SI) gains considerable improvements on the more complex motion sequences such as *Soccer* with 61.1% and *Foreman* with 35.1% bitrate savings for WZ frames. The improvements are also robust ranging from the complex sequences, e.g. *Soccer*, to the simple motion sequences, e.g. *Hall Monitor*. As a special case, the performance of SING(3SI) for *Hall Monitor* is slightly worse than SING(2SI) as shown in Tables II-V. Looking at both rate and distortion results, the bit rate is, as expected, lower for SING(3SI) than for SING(2SI), but the problem is that the PSNR of SING(3SI) is also lower than that of SING(2SI). In general, the RD performances of all methods proposed are robustly better than using the noise model in [7], as well as DISCOVER. It may be noted that the

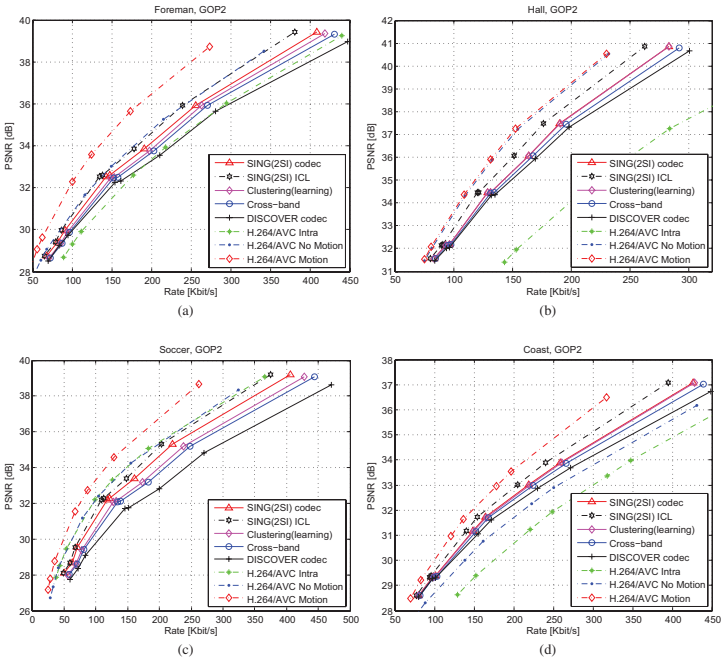


Fig. 10. PSNR versus rate for the proposed SING(2SI) codec for all frames (QCIF, 15 Hz, GOP2). (a) *Foreman*. (b) *Hall*. (c) *Soccer*. (d) *Coast*.

encoding and thereby also encoding complexity are the same in all cases.

The RD performance of the SING(2SI) codec and H.264/AVC coding is also depicted in Figs. 9-10 for WZ frames and all frames, respectively. The SING(2SI) codec gives a better RD performance than H.264/AVC Intra coding for *Foreman*, *Hall Monitor*, and *Coastguard*, and also better than H.264/AVC No Motion for *Coastguard*. The RD performance of the SING(2SI) codec clearly outperforms those of [7] and DISCOVER. For medium to high rates the improvement for *Soccer* is up to 4 dB for WZ frames. The Ideal Code Length (ICL) (23) measures the quality of the side information of the coded coefficients. The SING(2SI) ICL result (Fig. 10) actually matches those of H.264/AVC No Motion for *Foreman* and *Soccer*. For *Hall Monitor* SING(2SI) ICL is close to H.264/AVC Motion. This illustrates that if more efficient Slepian-Wolf coding is developed, the performance gap between practical Wyner-Ziv video coding and the conventional predictive video coding would be further reduced.

We have tested the proposed scheme SING(2SI) on four test sequences (299 frames, QCIF at 30 Hz of) *Foreman*, *Soccer*, *Hall Monitor*, and *Coastguard* using a GOP size 4. The two key frames are again coded using H.264/AVC Intra. Thereafter GOP4 follows the hierarchical decoding order in Section III-C.2, where the middle frame is first decoded based on the two decoded key frames (Section III-C.1) and then the two remaining frames are decoded based on the nearest decoded key frame and the decoded middle frame. RD points are calculated for the four 4×4 quantization matrices Q1, Q4, Q7, and Q8 [5]. The RD performance of the SING(2SI) codec in Fig. 11 is better than those obtained by the Cross-band codec [7] and DISCOVER. The SING(2SI) codec gives a better RD performance than H.264/AVC Intra and also better than H.264/AVC No Motion for *Foreman* and *Coastguard*. In particular, the SING(2SI) codec performance matches that of H.264/AVC No Motion for the high motion sequence *Soccer*. Compared to DISCOVER, the average Bjøntegaard bitrate saving is 37.5% and 23% (or equivalently the average PSNR improvement is 1.5 dB and 1.1 dB) for WZ frames and all frames, respectively. For the difficult sequence *Soccer*, the bitrate saving is 54.4% (or equivalently the improvement in PSNR is 2.2 dB) for WZ frames. The results may be compared with the GOP4 results in [6] at 400 Kb/s. Compared to DISCOVER, the results in [6] show an improvement of 1.0 dB for *Foreman* (QCIF, 15 Hz) and 0.9 dB for *Soccer* (QCIF, 15 Hz).

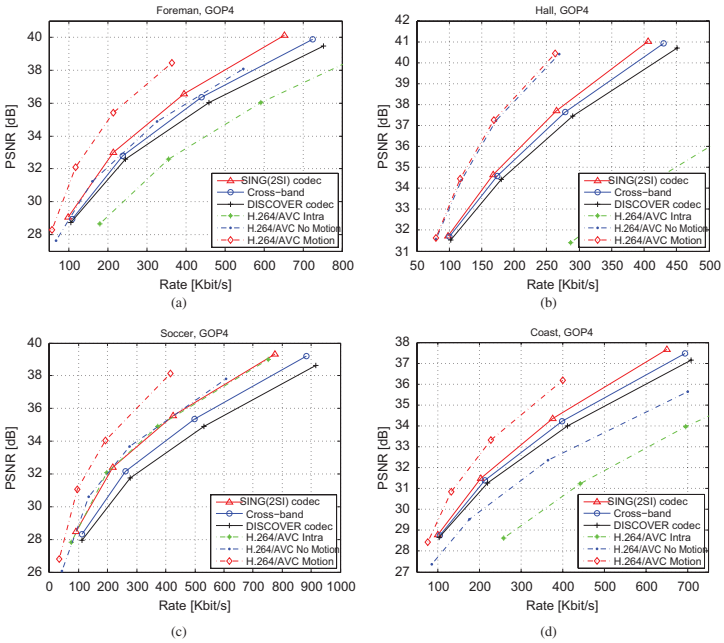


Fig. 11. PSNR versus rate for the proposed SING(2SI) codec for all frames (QCIF, 30 Hz, GOP4). (a) *Foreman*. (b) *Hall*. (c) *Soccer*. (d) *Coast*.

In comparison, an improvement of 1.6 dB for *Foreman* and 1.9 dB for *Soccer* are achieved by the SING(2SI) codec as seen in Figs. 11(a) and 11(c).

VII. Conclusion

In this paper, TDWZ video coding was improved using optical flow and clustering of DCT blocks. Optical flow was used for frame interpolation generating side information, which was adopted in a multihypothesis scheme to compensate weaknesses of block-based methods. Adaptive noise modeling using clustering was introduced additionally utilizing residues of previously decoded frames, and generating a number of noise residual distributions within a frame for adaptive optimization of the soft side information during decoding. Moreover, the adaptive noise model refined the residue to take advantage of correlation of DCT coefficients between neighboring blocks. Experimental results showed that the coding efficiency of the proposed SING scheme which combines all the techniques can significantly improve the RD performance of TDWZ video compared to DISCOVER as well as the cross-band TDWZ scheme in [7] without changing the encoder. For a GOP size of 2, the average bit-rate saving of the SING(3SI) codec is 37% (or equivalent the average improvement in PSNR is 1.5 dB) on WZ frames compared with the DISCOVER codec.

References

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

[2] R. Puri, A. Majumdar, and K. Ramchandran, "Prism: A video coding paradigm with motion estimation at the decoder," *IEEE Trans. Image Process.*, vol. 16, no. 10, pp. 1–13, Oct. 2007.

[3] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. 19, no. 4, pp. 471–480, Jul. 1973.

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

[5] *Discover Project*. (2007, Dec.) [Online]. Available: http://www.discoverdvc.org/

[6] R. Martins, C. Brites, J. Ascenso, and F. Pereira, "Refining side information for improved transform domain Wyner-Ziv video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 9, pp. 1327–1341, Sep. 2009.

[7] X. Huang and S. Forchhammer, "Cross-band noise model refinement for transform domain Wyner-Ziv video coding," *Signal Process., Image Commun.*, vol. 27, no. 1, pp. 16–30, 2012.

[8] D. Kubasov, J. Nayak, and C. Guillemot, "Optimal reconstruction in Wyner-Ziv video coding with multiple side information," in *Proc. IEEE Int. Workshop Multimedia Signal Process.*, Chania, Greece, Oct. 2007, pp. 183–186.

[9] X. Huang, J. Ascenso, C. Brites, F. Pereira, and S. Forchhammer, "Distributed video coding with multiple side information," in *Proc. Picture Coding Symp.*, Chicago, IL, May 2009, pp. 1–4.

[10] J. Skorupa, J. Slowack, S. Mys, N. Deligiannis, J. D. Cock, P. Lambert, C. Grecos, A. Munteanu, and R. V. de Walle, "Efficient low-delay distributed video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 4, pp. 530–544, Sep. 2011.

[11] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. Int. Joint Conf. Artif. Intell.*, Vancouver, BC, Canada, Aug. 1981, pp. 674–679.
[12] C. Brites and F. Pereira, "Correlation noise modeling for efficient pixel and transform domain Wyner-Ziv video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 9, pp. 1177–1190, Sep. 2008.
[13] G. R. Esmaili and P. C. Cosman, "Wyner-Ziv video coding with classified correlation noise estimation and key frame coding mode selection," *IEEE Trans. Image Process.*, vol. 20, no. 9, pp. 2463–2474, Sep. 2011.
[14] H. V. Luong, X. Huang, and S. Forchhammer, "Adaptive noise model for transform domain Wyner-Ziv video using clustering of DCT blocks," in *Proc. IEEE Int. Workshop Multimedia Signal*, Hangzhou, China, Oct. 2011, pp. 1–6.
[15] S. Wang, L. Cui, L. Stankovic, V. Stankovic, and S. Cheng, "Adaptive correlation estimation with particle filtering for distributed video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 5, pp. 649–658, May 2012.
[16] D. Varodayan, A. Aaron, and B. Girod, "Rate-adaptive codecs for distributed source coding," *EURASIP Signal Process.*, vol. 86, no. 11, pp. 3123–3130, Nov. 2006.
[17] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *Int. J. Comput. Vis.*, vol. 92, no. 1, pp. 1–31, 2011.
[18] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, nos. 1–3, pp. 185–203, Aug. 1981.
[19] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *Proc. Eur. Conf. Comput. Vis.*, Prague, Czech Republic, May 2004, pp. 25–36.
[20] A. Bruhn, J. Weickert, and C. Schnörr, "Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods," *Int. J. Comput. Vis.*, vol. 61, no. 3, pp. 211–231, 2005.
[21] J. Weickert and C. Schnörr, "A theoretical framework for convex regularizers in PDE-based computation of image motion," *Int. J. Comput. Vis.*, vol. 45, no. 3, pp. 245–264, 2001.
[22] H. Zimmer, A. Bruhn, and J. Weickert, "Optic flow in harmony," *Int. J. Comput. Vis.*, vol. 93, no. 3, pp. 368–388, 2011.
[23] L. L. Rakêt, L. Roholm, M. Nielsen, and F. Lauze, "TV-L^1 optical flow for vector valued images," in *Proc. Int. Conf. Energy Minim. Methods Comput. Vis. Pattern Recognit.*, Saint Petersburg, Russia, Jul. 2011, pp. 329–343.
[24] X. Bresson and T. Chan, "Fast dual minimization of the vectorial total variation norm and application to color image processing," *Inverse Probl. Imag.*, vol. 2, no. 4, pp. 455–484, 2008.
[25] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime TV-L^1 optical flow," in *Proc. Ann. Symp. German Assoc. Pattern Recognit.*, Heidelberg, Germany, Sep. 2007, pp. 214–223.
[26] A. Chambolle, "An algorithm for total variation minimization and applications," *J. Math. Imag. Vis.*, vol. 20, nos. 1–2, pp. 89–97, 2004.
[27] E. Herbst, S. Seitz, and S. Baker, "Occlusion reasoning for temporal interpolation using optical flow," Dept. Comput. Sci. Eng., Univ. Washington, Seattle, Tech. Rep. UW-CSE-09-08-01, 2009.
[28] X. Huang, L. L. Raket, H. V. Luong, M. Niesen, F. Lauze, and S. Forchhammer, "Multi-hypothesis transform domain wyner-ziv video coding including optical flow," in *Proc. IEEE Int. Workshop Multimedia Signal Process.*, Hangzhou, China, Oct. 2011, pp. 1–6.
[29] P. L. Dragotti and M. Gastpar, *Distributed Source Coding: Theory, Algorithms and Applications*. New York: Academic, 2009.
[30] R. L. Cannon, J. V. Dave, and J. C. Bezdek, "Efficient implementation of the fuzzy c-means clustering algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 2, pp. 248–255, Mar. 1986.
[31] *Joint Video Team (JVT) Reference Software* [Online]. Available: http://iphome.hhi.de/suehring/tml/index.htm
[32] G. Bjøntegaard, "Calculation of average psnr differences between RD curves," ITU, San Jose, CA, Tech. Rep. VCEG-M33, Apr. 2001.
[33] X. Fan, O. C. Au, and N. M. Cheung, "Transform-domain adaptive correlation estimation (trace) for Wyner-Ziv video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 11, pp. 1423–1436, Nov. 2010.

Huynh Van Luong received the M.Sc. degree in computer engineering from the University of Ulsan, Ulsan, Korea, in 2009. He is currently pursuing the Ph.D. degree with the Coding and Visual Communication Group, Technical University of Denmark, Lyngby, Denmark.

His current research interests include image and video processing and coding, distributed source coding, visual communications, and multimedia systems.

Lars Lau Rakêt was born in 1985. He received the M.Sc. degree in statistics from the University of Copenhagen, Copenhagen, Denmark, in 2010. He is currently pursuing the Ph.D. degree with the Image Group, Department of Computer Science, University of Copenhagen.

His current research interests include analysis and processing of image and video data, particularly motion estimation and statistical analysis of high-dimensional functional data.

Xin Huang received the B.A degree in telecommunication from Xidian University, Xi'an, China, in 2004, and the M.Sc. and Ph.D. degrees from the Technical University of Denmark, Lyngby, Denmark, in 2006 and 2009, respectively.

He was a Post-Doctoral Researcher with the Coding and Visual Communications Group, Technical University of Denmark, from 2009 to 2011. He is currently a Senior Engineer with Renesas Mobile, Copenhagen, Denmark. His current research interests include image and video coding, image and video processing, and error correction codes.

Søren Forchhammer (M'04) received the M.S. degree in engineering and the Ph.D. degree from the Technical University of Denmark (DTU), Lyngby, Denmark, in 1984 and 1988, respectively. He has been a Professor with DTU Fotonik, Technical University of Denmark, since 1988, where he is the Head of the Coding and Visual Communication Group. His current research interests include source coding, image and video coding, distributed source coding, distributed video coding, processing for image displays, 2-D information theory, and visual communication.

MULTIPLE LDPC DECODING USING BITPLANE CORRELATION FOR TRANSFORM DOMAIN WYNER-ZIV VIDEO CODING

Huynh Van Luong, Xin Huang, and Søren Forchhammer

DTU Fotonik, Technical University of Denmark, Building 343, Lyngby 2800, Denmark
Email: {hulu, xhua, sofo}@fotonik.dtu.dk

ABSTRACT

Distributed video coding (DVC) is an emerging video coding paradigm for systems which fully or partly exploit the source statistics at the decoder to reduce the computational burden at the encoder. This paper considers a Low Density Parity Check (LDPC) based Transform Domain Wyner-Ziv (TDWZ) video codec. To improve the LDPC coding performance in the context of TDWZ, this paper proposes a Wyner-Ziv video codec using bitplane correlation through multiple parallel LDPC decoding. The proposed scheme utilizes inter bitplane correlation to enhance the bitplane decoding performance. Experimental results show that the proposed scheme reduces the bit rate up to 3.9% and improves the rate-distortion (RD) performance of TDWZ.

Index Terms— Wyner-Ziv video coding, multiple decoders, bitplane correlation

1. INTRODUCTION

Distributed Video Coding [1][2] proposes to fully or partly exploit the video redundancy at the decoder, rather than at the encoder as in predictive video coding. According to the Slepian-Wolf theorem [3], it is possible to achieve the same rate by independently encoding but jointly decoding two statistically dependent signals as for typical joint encoding and decoding (with a vanishing error probability). The Wyner-Ziv theorem [4] extends the Slepian-Wolf theorem to the lossy case, becoming the theoretical basis for DVC where source data are lossy coded and decoded based on a correlated source at the decoder providing the so-called side information.

Transform Domain Wyner-Ziv (TDWZ) video coding is a popular approach to DVC. This approach was first proposed in [5], and thereafter improved by e.g. advanced side information generation schemes [6]-[9], finer noise models [7][10] and refinement schemes [11][12]. Despite the advances in practical TDWZ video coding, the RD performance of TDWZ video coding still remains to reach the performance of conventional video coding, such as H.264/AVC. The coding efficiency of error correcting codes, an LDPC Accumulate (LDPCA) codec [13] in this paper, plays a key role in TDWZ in terms of overall RD performance. To improve the RD performance, a Wyner-Ziv codec with multiple LDPCA decoders is proposed in this paper. The proposed scheme is inspired by the work in [14] using joint bitplane LDPC decoding. Different from [14], the proposed Wyner-Ziv codec utilizes multiple LDPCA decoders in parallel and takes inter bitplane correlation into account during decoding, thereby improving the overall RD performance of the TDWZ codec. The modifications involve the buffer part and the decoder, while the Wyner-Ziv encoder is not changed.

The rest of the paper is organized as follows. Section 2 presents the state-of-the-art TDWZ video codec adopted in this paper. Section 3 describes the proposed Wyner-Ziv codec with multiple LDPCA decoders. Section 4 analyzes the performance of our approach and compares with other existing methods.

2. STATE-OF-THE-ART TRANSFORM DOMAIN WYNER-ZIV VIDEO CODING

The architecture of a state-of-the-art TDWZ video codec is depicted in Fig. 1. It basically follows the same architecture as the one developed by the DISCOVER project [6]. However, a better side information generation scheme [8] and an improved noise model [10] are adopted to achieve a better RD performance.

At the encoder, periodically one frame out of N in the video sequence is named as key frame and intermediate frames are WZ frames. The key frames are intra coded by using a conventional video coding solution with low complexity such as H.264/AVC Intra, while the WZ frames in between are coded with a Wyner-Ziv approach. WZ frames are transformed using a 4x4 block size and the transformed coefficients within the same frequency band are grouped together and then quantized. DC coefficients and AC coefficients are uniformly scalar quantized and dead-zone quantized, respectively. Thereafter quantized coefficients are decomposed into bitplanes, each bitplane is fed to a rate-compatible LDPCA encoder [13] starting from the most significant bitplane (MSB) to least significant bitplane (LSB). For each encoded bitplane, the corresponding accumulated syndrome is stored in a buffer together with an 8-bit Cyclic Redundancy Check (CRC). The amount of bits to be transmitted depends on the requests made by the decoder through a feedback channel as shown in Fig. 1.

At the decoder, a side information frame is interpolated and the corresponding noise residue is generated by using previously decoded frames. Given the available side information, soft-input information (conditional bit probabilities Pr) within each bitplane is estimated using a noise model. Thereafter the LDPCA decoder starts to decode the various bitplanes, ordered from MSB to LSB, to correct the bit errors. For each bitplane, convergence is tested by the 8-bit CRC sum and the Hamming distance between the received syndrome and the decoded bitplanes [6]. After all the bitplanes are successfully decoded, the Wyner-Ziv frame can be decoded through combined de-quantization and reconstruction followed by an inverse transform.

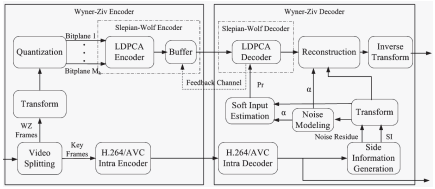


Fig. 1. Architecture of feedback channel based Transform Domain Wyner-Ziv video codec

For the LDPCA decoding, a Belief-Propagation (BP) algorithm is used to retrieve each transmitted bitplane. The BP algorithm is a soft-decoding approach, which is passing a Log-Likelihood Ratio (LLR) of *Pr* back and forth between source nodes and the syndrome nodes. Let $X=(b_{m-1},\ldots, b_1, b_0)$ denote a quantized DCT coefficient of a Wyner-Ziv frame, where b_{m-1} is an MSB bit and b_0 is an LSB bit and Y denotes a quantized DCT coefficient of the side information. The LLR of a bit b_i ($0\le i\le m$-1) of the i^{th} significant bitplane is described as:

$$L(b_i)=\log\left(\frac{\Pr(b_i=0\mid Y,b_{m-1},\ldots,b_{i+1})}{\Pr(b_i=1\mid Y,b_{m-1},\ldots,b_{i+1})}\right) \quad (1)$$

where $b_{m-1}\ldots b_{i+1}$ represent bits from previous successfully decoded bits of the transformed coefficient. The LDPCA decoder utilizes information from previous successfully decoded bitplanes for decoding future bitplanes. The BP algorithm performs an approximation of the Maximum-Likelihood decoding to determine an estimate of the transmitted bits.

3. WYNER-ZIV CODEC WITH MULTIPLE LDPCA DECODERS

In the TDWZ codec described in Section 2, the LDPCA decoder utilizes side information, modeled noise correlation and the information from previous decoded bitplanes to decode future bitplanes. However, the inter bitplane correlation is not fully explored during decoding, although a refinement scheme is employed in [12] to utilize the bitplane correlation to update soft-input for decoding further bitplanes. The limitation is that the soft-input of the LDPCA decoder is fixed until successful decoding. To overcome the above limitations and improve the performance of the LDPCA codec, a decoder may iteratively refine soft-input for each bitplane during the decoding process and take inter bitplane correlation into account. Thus, a Wyner-Ziv codec with multiple LDPCA decoders is proposed.

The multiple LDPCA decoders are running in parallel to keep refining soft-input at each iteration. Each LDPCA decoder is responsible for one bitplane. Different from single bitplane LDPCA decoding, where the decoder corrects errors one bitplane after another e.g., from MSB to LSB, the proposed Wyner-Ziv codec with multiple LDPCA decoders operates on all available bitplanes at once and exploits the correlation between bitplanes and passes syndrome information from one bitplane to another. Once a bitplane is successfully decoded, instantaneously, the responsible LDPCA decoder no longer requests syndrome bits from the buffer. Meanwhile, the rest of LDPCA decoders are reinitialized using the new soft-inputs, which were updated conditional on the successfully decoded bitplane.

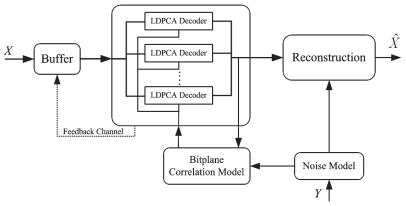


Fig. 2. Multiple LDPCA Decoders

We illustrate the proposed Wyner-Ziv codec with multiple LDPCA decoders in Fig. 2. It includes a critical part called the bitplane correlation model to reform soft-input based on feedback from the LDPCA decoders and the estimated noise distribution from the noise model. The bitplane correlation model collects all the information from multiple LDPCA decoders to recalculate soft-inputs. The new soft-input information of the source X is estimated and updated, where Y is combined with a given Laplacian parameter from the noise model.

The main difference between our proposed approach and [6] is that the LLR of a bit b_i ($0\le i\le m$-1) of the i^{th} significant bitplane is computed conditioned on the binary distributions (β_k, 1- β_k) of the remaining bits, b_k ($k\ne i$). This means that the LLR is calculated by using soft information. Let $\beta_k = Pr(b_k=0)$ denote a probability of bitplane k. Moreover, the decoding order of our approach does not consider the significance of bitplanes. The LLR described in formula (1) only uses the bits from previous successfully decoded bitplanes and decodes from MSB to LSB. Here the LLR expression is generalized for a bit b_i of bitplane i as:

$$L(b_i)=\log\left(\frac{\Pr(b_i=0\mid Y,\beta_{m-1},\ldots,\beta_{i+1},\beta_{i-1},\ldots,\beta_1,\beta_0)}{\Pr(b_i=1\mid Y,\beta_{m-1},\ldots,\beta_{i+1},\beta_{i-1},\ldots,\beta_1,\beta_0)}\right) \quad (2)$$

where β_k are soft-input values for the same coefficient as b_i.

To understand the method, we should take into account both bitplane (bit) and coefficient (symbol) levels to get soft side information updated via one BP algorithm used for LDPCA decoding which is propagated to bit level and thereafter symbol level. Similar to [14], the key idea is to use the BP mechanism during the decoding of a frame and to convert the LLR back and forth between symbol level and bit level. Distinctly, in the proposed method, the soft-input is only updated after the multiple LDPCA decoders of one coefficient band are completely processed (using a certain number of iterations) at bit level based on the given syndrome bits. Let $Pr^{(t-1)}(b_k)$ denote the probability of bit b_k at the iteration t-1 at bit level. The LLR of bit b_i, is updated at iteration t as an approximation of (2):

$$L^{(t)}(b_i)=\log\left(\frac{\sum_{X\in S}\Pr(X\mid Y,b_i=0)\prod_{k\ne i}\Pr^{(t-1)}(b_k)}{\sum_{X\in S}\Pr(X\mid Y,b_i=1)\prod_{k\ne i}\Pr^{(t-1)}(b_k)}\right) \quad (3)$$

where $X=(b_{m-1},\ldots, b_1, b_0)$ and S indicates the set of values $\{0,1,2,\ldots,2^m-1\}$ for the coefficient X which is coded by m bitplanes (for DC and the magnitude of AC coefficients). $Pr(X|Y,b_i)$ is calculated at symbol level by using the estimated noise distribution between the side information frame and the original Wyner-Ziv frame via a noise model as shown in Fig. 2 and selecting X with $b_i=0$ and $b_i=1$ in the numerator and denominator in (3), respectively.

The LLRs at iteration t noted by $L^{(t)}(b_i)$, are in turn input to multiple LDPCA decoders. After one LDPCA is processed, $L^{*(t)}(b_i)$

is temporarily achieved as output. The updated $Pr^{(t)}(b_i)$ values are obtained based on LLR definition:

$$L^{*(t)}(b_i) = \log\left(\frac{\Pr^{(t)}(b_i = 0)}{1 - \Pr^{(t)}(b_i = 0)}\right) \quad (4)$$

i.e. for the next iteration, we have:

$$\Pr^{(t)}(b_i = 0) = \frac{1}{2}\left(1 + \tanh\left(\frac{L^{*(t)}(b_i)}{2}\right)\right) \quad (5)$$

This $Pr^{(t)}(b_i)$ is used as a new probability of bit b_i to compute new LLRs, $L^{(t+1)}(b_i)$, for the next iteration of multiple LDPCA decoding based on (3).

Since all LDPCA decoders are running in parallel, once a bitplane is successfully decoded, instantaneously, the re-initialization procedure is performed. The new soft-inputs for the rest of the bitplanes are assigned conditional on the successfully decoded bitplane. The LDPCA decoder with the successfully decoded bitplane will no longer request syndromes from the buffer. Assume b_i is successfully decoded with value *0*, then $Pr^{(t)}(b_i=0)=1$ and the iteration count is reset as $t=0$. In addition, the remaining unfinished bitplanes are re-initialized by $Pr^{(0)}(b_j=0)=1/2$. The LDPCA decoders are iteratively operated up to a maximum numbers of iterations (T_{max}) with the given syndrome bits. If they are not successful after this number of iterations, the LDPCA decoders request more syndrome bits from the buffer via the feedback channel. Then a new process is started until all the bitplanes of the DCT coefficient are successfully decoded. Let N_{max} denote a maximum numbers of syndromes.

Overall, the multiple LDPCA decoding is handled as follows:

1. **Initiate parameters**. Iteration count $t=0$; Number of syndrome bits $n=0$; For all bits b_i, $Pr^{(0)}(b_i=0)=1/2$.
2. **Increase and check conditions**.
   a. **Syndrome bit condition**: Increase $n=n+1$. If $n \geq N_{max}$ then end, else go to Step 2.b.
   b. **Iteration count condition**: Increase $t=t+1$. If $t<T_{max}$ go to Step 3, else return to 2.a.
3. **Compute the LLRs.** At bit level, formula (3) is computed to get the LLRs, $L^{(t)}(b_i)$, by multiplying the soft side information, $Pr(X|Y,b_i)$ of symbol level, and the probabilities, $Pr^{(t-1)}(b_k)$, of bitplane level ($k \neq i$).
4. **Check if any LDPCA is successfully decoded?**
   a. **No: Compute probabilities of bitplanes**. $L^{(t)}(b_i)$ are forwarded to multiple LDPCA decoders where $L^{*(t)}(b_i)$ are received from LDPCA outputs. New probabilities of bitplanes, $Pr^{(t)}(b_i)$, are obtained by (5).
   b. **Yes: Re-initialize the process**. Assume LDPCA (b_i) is successfully decoded with value $b_i=0$, assign $Pr^{(t)}(b_i=0)=1$. Reset iteration count $t=0$ and the remaining unfinished LDPCA decoders by $Pr^{(0)}(b_j=0)=1/2$;
5. **Check all LDPCA decoders**. The process is ended if all bitplanes are successfully decoded, otherwise, go to step 2.b.

The above procedure is repeated for all bands of the DCT coefficients for which Wyner-Ziv bits are transmitted. Restarting the decoding of single LDPCA does increase complexity of the decoding.

4. PERFORMANCE EVALUATION

In this section, the RD performance of the proposed approach is presented and compared with the state-of-the-art TDWZ video codec described in Section 2 as well as relevant benchmarks. The test sequences are 149 frames of *Foreman*, *Hall Monitor*, *Soccer*, and *Coast-guard* with 15Hz frame rate and QCIF format. GOP (group of pictures) size is 2, where the first frame is coded as a key frame using H.246/AVC Intra and other frame is coded using Wyner-Ziv coding. Eight RD points (Q_j) are considered corresponding to eight 4x4 quantization matrices [6]. The values within these matrices determine the number of bitplanes associated to the DCT coefficient bands, therefore, the number of LDPCA decoding instances is known. The proposed model uses m (number of bitplanes of a given band) regular LDPC accumulate decoders [13] with a length of 1584 bits for each. At these settings, exactly 1584 transform coefficients per given band of a frame can be decoded at a time by m LDPCA each decoding one bitplane.

Table 1 shows rate and PSNR values of the proposed TDWZ codec with multiple LDPCA decoders (WZMD) as well as the savings in total rate, ΔR (in %), and WZ rate, ΔR_{WZ} (in %), compared with the state-of-the-art TDWZ codec [10]. The WZMD achieves a reduction of bit-rate for WZ frames up to 1.8% for *Foreman*; 2.59% for *Hall Monitor*; 2.26% for *Soccer*; 1.82% for *Coast-guard*. In terms of the overall bit-rate, it saves up to 0.82% for *Foreman* sequence; 0.59% for *Hall Monitor*; 1.46% for *Soccer*; 0.52% for *Coast-guard*. It can be noted that the same PSNR values were obtained for both WZMD and TDWZ [10].

In some cases, the required number of syndromes consumed for the LSB is (close to) N_{max}, even though there is still some correlation. This is due to a (relative) loss in the LDPCA decoder, which may be reduced by first coding the LSB independently and thereafter apply WZMD to the remaining bitplanes having decoded the LSB. This is called WZMD(LSB). As a result, the coding efficiency in terms of bit-rate is improved. Table 2 depicts the bit rate savings for WZMD and WZMD(LSB) compared with TDWZ [10]. The results shows that WZ rate savings up to 3.9% for *Foreman* and 3.77% for *Soccer*.

Table 1. Total rate and WZ rate savings (in %) for WZMD based TDWZ compared with TDWZ [10]

Q_j	Foreman				Hall				Soccer				Coast-guard			
	Rate [kbps]	PSNR [dB]	ΔR [%]	ΔR_{WZ} [%]	Rate [kbps]	PSNR [dB]	ΔR [%]	ΔR_{WZ} [%]	Rate [kbps]	PSNR [dB]	ΔR [%]	ΔR_{WZ} [%]	Rate [kbps]	PSNR [dB]	ΔR [%]	ΔR_{WZ} [%]
1	72,34	28,67	0,49	1,32	84,71	31,55	0,21	1,46	58,58	28,01	1,26	1,88	81,68	28,59	0,32	1,58
2	86,98	29,36	0,62	1,51	95,07	32,13	0,32	1,77	71,93	28,60	1,11	1,62	99,83	29,32	0,35	1,53
3	96,60	29,87	0,53	1,30	98,04	32,15	0,20	1,00	82,18	29,41	1,38	2,06	103,11	29,36	0,41	1,62
4	152,38	32,44	0,68	1,66	131,73	34,42	0,36	1,72	135,54	32,04	1,19	1,88	152,93	31,15	0,42	1,53
5	158,49	32,50	0,78	1,80	134,20	34,42	0,59	2,59	140,64	32,09	1,46	2,26	169,16	31,69	0,27	1,05
6	204,06	33,74	0,82	1,78	168,26	36,03	0,46	1,80	184,11	33,16	1,35	2,06	224,74	32,98	0,52	1,82
7	271,93	35,90	0,73	1,61	196,22	37,43	0,46	1,71	250,86	35,16	1,14	1,80	266,44	33,87	0,49	1,44
8	433,19	39,31	0,66	1,35	291,75	40,84	0,52	1,82	449,12	39,02	0,73	1,24	440,01	37,04	0,43	1,07

Table 2. Bit rate savings (in %) of WZMD and WZMD (LSB)

Q_j	Foreman				Soccer			
	WZMD		WZMD(LSB)		WZMD		WZMD(LSB)	
	ΔR [%]	ΔR_{WZ} [%]	ΔR [%]	ΔR_{WZ} [%]	ΔR [%]	ΔR_{WZ} [%]	ΔR [%]	ΔR_{WZ} [%]
1	0,49	1,32	1,44	3,90	1,26	1,88	2,51	3,77
2	0,62	1,51	1,48	3,60	1,11	1,62	1,95	2,84
3	0,53	1,30	0,99	2,41	1,38	2,06	1,89	2,82
4	0,68	1,66	0,68	1,66	1,19	1,88	1,38	2,18
5	0,78	1,80	0,78	1,80	1,46	2,26	1,62	2,51
6	0,82	1,78	1,05	2,26	1,35	2,06	1,41	2,15
7	0,73	1,61	0,86	1,89	1,14	1,80	1,29	2,03
8	0,66	1,35	0,79	1,62	0,73	1,24	0,80	1,36

The experimental results in Fig. 3 demonstrate that the proposed approach significantly improves RD performance compared with the DISCOVER codec, with PSNR gains up to about 0.7 dB for *Foreman* and 0.9 dB for *Soccer*. The performance of H.264/AVC (Intra) and the H.264/AVC (No Motion) codecs are also included. The WZMD is more efficient than H.264/AVC (Intra) for *Foreman*. H.264/AVC (No Motion) codec is more efficient than the TDWZ codecs for both sequences since it exploits co-located frame differences at the encoder.

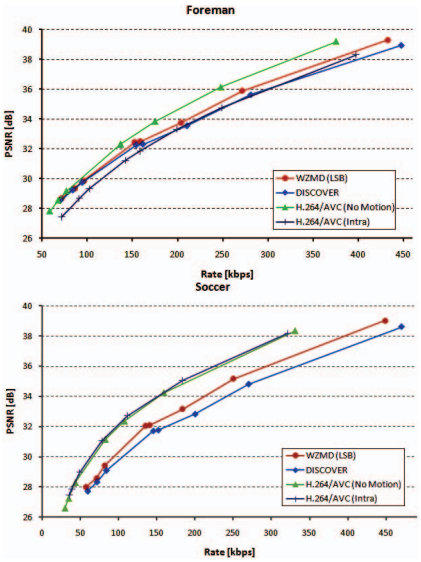


Fig. 3. RD performance comparison

5. CONCLUSION

This paper proposes a Wyner-Ziv video codec using multiple parallel LDPC decoding to utilize inter bitplane correlation. The technique takes bitplane correlation into account by iteratively refining the soft-input for each bitplane during decoding. Experimental results show that the proposed multiple LDPC decoding can improve the coding efficiency of TDWZ in terms of WZ rate savings up to 3.9% compared with the existing TDWZ [10] and provide better RD performance than DISCOVER codec.

6. REFERENCES

[1] A. Aaron, R. Zhang, and B. Girod, "Wyner-Ziv coding of motion video," *Proc. Asilomar Conference on Signals and Systems*, Pacific Grove, CA, Nov. 2002.

[2] R. Puri, A. Majumdar, and K. Ramchandran, "PRISM: a video coding paradigm with motion estimation at the decoder," *IEEE Trans. On Image Proc*., vol. 16 (10), pp. 1-13, Oct. 2007.

[3] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. on Inform. Theory*, vol. 19 (4), pp. 471–480, Jul. 1973.

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

[5] A. Aaron, S. Rane, E. Setton, and B. Girod, "Transform domain Wyner-Ziv codec for video," *Proc. SPIE VCIP*, San Jose, CA , USA, Jan. 2004.

[6] DISCOVER Project, www.discoverdvc.org, Dec. 2007.

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

[8] X. Huang and S. Forchhammer, "Improved side information generation for distributed video coding," *IEEE Int'l Workshop Multimedia Signal Proc.*, Cairns, Australia, Oct. 2008.

[9] X. Huang, J. Ascenso, C. Brites, F. Pereira and S. Forchhammer, "Distributed video coding with multiple side information," *Picture Coding Symposium*, Chicago, USA, May 2009.

[10] X. Huang and S. Forchhammer, "Improved virtual channel noise model for transform domain Wyner-Ziv video coding," *IEEE Int'l Conf. on Acoustics, Speech, and Signal Proc*., Taipei, Taiwan, ROC. April 2009.

[11] R. Martins, C. Brites, J. Ascenso, F. Pereira, "Refining Side Information for Improved Transform Domain Wyner-Ziv Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19 (9), pp. 1327-1341, Sep. 2009.

[12] X. Huang and S. Forchhammer, "Transform domain Wyner-Ziv video coding with refinement of noise residue and side information," *SPIE Visual Communications and Image Processing*, HuangShan, China, July 2010.

[13] D. Varodayan, A. Aaron, B. Girod, "Rate-adaptive codes for distributed source coding," *EURASIP Signal Processing*, vol. 23 (11), pp. 3123-3130, 2006.

[14] D. Varodayan, D. Chen, M. Flierl, B. Girod, "Wyner-Ziv coding of video with unsupervised motion vector learning," *EURASIP Signal Processing, Image Communication*, vol. 23 (5), pp. 369-378, 2008.

2011 18th IEEE International Conference on Image Processing

PARALLEL ITERATIVE DECODING OF TRANSFORM DOMAIN WYNER-ZIV VIDEO USING CROSS BITPLANE CORRELATION

Huynh Van Luong, Xin Huang, and Søren Forchhammer

DTU Fotonik, Technical University of Denmark, Building 343, Lyngby 2800, Denmark
Email: {hulu, xhua, sofo}@fotonik.dtu.dk

ABSTRACT

In recent years, Transform Domain Wyner-Ziv (TDWZ) video coding has been proposed as an efficient Distributed Video Coding (DVC) solution, which fully or partly exploits the source statistics at the decoder to reduce the computational burden at the encoder. In this paper, a parallel iterative LDPC decoding scheme is proposed to improve the coding efficiency of TDWZ video codecs. The proposed parallel iterative LDPC decoding scheme is able to utilize cross bitplane correlation during decoding, by iteratively refining the soft-input, updating a modeled noise distribution and thereafter enhancing the bitplane decoding performance. Experimental results show that the proposed scheme reduces the bit rate of Wyner-Ziv frames up to 5.6% and improves the rate-distortion (RD) performance of TDWZ.

Index Terms— Wyner-Ziv, Cross-bitplane correlation, noise distribution

1. INTRODUCTION

In conventional predictive video coding, the video redundancy is fully or partly exploited at the encoder side. However, in recent years, the conventional video coding architecture has been challenged by some emerging applications such as video surveillance, video sensor networks and wireless cameras etc. These applications require a relative low cost encoder. Distributed Video Coding [1][2] is proposed to match the low cost encoding requirement, be exploring the video statistics, partially or totally, at the decoder only. According to the Slepian-Wolf theorem [3], it is possible to achieve the same rate by independently encoding but jointly decoding two statistically dependent signals as for typical joint encoding and decoding (with a vanishing error probability). The Wyner-Ziv theorem [4] extends the Slepian-Wolf theorem to the lossy case, becoming the theoretical basis for DVC where source data are lossy coded and decoded based on a correlated source at the decoder providing the so-called side information.

Transform Domain Wyner-Ziv (TDWZ) video coding is one of the most efficient approaches to DVC. It was first proposed in [5], and thereafter improved by many other techniques, e.g. advanced side information generation schemes [6]-[9], finer noise models [7][10] and refinement schemes [11][12]. To further improve the coding efficiency of TDWZ video coding, a Wyner-Ziv codec with parallel iterative LDPC decoding is proposed in this paper. The proposed scheme is based on the previous work in [13], inspired by the work in [14] using joint bitplane LDPC decoding and the work in [12] with refinement of the modeled noise distribution. The main advantage of joint bitplane LDPCA decoding is to exploit correlation across bitplanes by exchanging soft information between bitplanes during the decoding. Different from [12][14], the proposed scheme utilizes multiple LDPCA decoders in parallel, taking inter bitplane correlation into account to iteratively refine the soft-input of bitplanes and update a modeled noise distribution during decoding, thereby improving the overall RD performance of the TDWZ codec. Compared with [13], the novelty is that the modeled noise distribution keeps updating based on the iteratively refined soft-input during parallel decoding. The rest of the paper is organized as follows. Section 2 presents the basic architecture of the TDWZ video codec adopted in this paper. Section 3 describes the proposed parallel iterative LDPC decoding scheme. The performance of the proposed approach is analyzed and compared with other existing methods in Section 4.

2. TRANSFORM DOMAIN WYNER-ZIV VIDEO CODING

The architecture of a TDWZ video codec [10] is depicted in Fig. 1. In this system, the frame sequence is split into key frames and so-called Wyner-Ziv frames. Key frames are intra coded using conventional video coding techniques such as H.264/AVC intra coding. The Wyner-Ziv frames are transformed, quantized and decomposed into bitplanes. Each bitplane is fed to a rate-compatible LDPC Accumulate (LDPCA) encoder [15] from most significant bitplane (MSB) to least significant bitplane (LSB). The corresponding error correcting information is stored in a buffer. The amount of information to be transmitted depends on the requests made by the decoder through a feedback channel. The Wyner-Ziv frame is predicted at the decoder side by using already decoded frames as references. The predicted frame, called Side Information (SI) frame, is an estimation of the original Wyner-Ziv frame available at the encoder. Given the available SI, soft-input information (conditional bit probabilities Pr) within each bitplane is estimated using a noise model [10]. Thereafter the LDPCA decoder starts to decode the various bitplanes, ordered from MSB to LSB, to correct the bit errors. After all the bitplanes are successfully decoded, the Wyner-Ziv frame can be decoded through combined de-quantization and reconstruction followed by an inverse transform.

In TDWZ video coding, coding efficiency of the LDPCA codec plays a key role in terms of overall RD performance. For LDPCA decoding, a Belief-Propagation (BP) algorithm is used to retrieve each transmitted bitplane. The BP algorithm is a soft-decoding approach, which is passing a Log-Likelihood Ratio (LLR) of Pr back and forth between source nodes and the syndrome nodes. Let $X=(b_{m-1},\ldots, b_1, b_0)$ denote a quantized DCT coefficient of a Wyner-Ziv frame, where b_{m-1} is an MSB bit and b_0 is an LSB bit and let Y denote a quantized DCT coefficient of the side information. The LDPCA corrects errors one bitplane after

another e.g., from MSB to LSB. The LLR of a bit b_i ($0 \le i \le m$-1) of the i^{th} significant bitplane is described as:

$$L(b_i) = \log\left(\frac{\Pr(b_i = 0 \mid Y, b_{m-1},...,b_{i+1})}{\Pr(b_i = 1 \mid Y, b_{m-1},...,b_{i+1})}\right) \quad (1)$$

where $b_{m-1}\ldots b_{i+1}$ represent bits from previous successfully decoded bits of the transformed coefficient. The LDPCA decoder utilizes information from previous successfully decoded bitplanes for decoding future bitplanes.

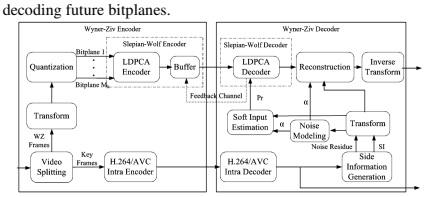


Fig. 1. Architecture of feedback channel based TDWZ video codec

3. WYNER-ZIV CODEC WITH PARALLEL ITERATIVE DECODING

In the TDWZ codec described in Section 2, the LDPCA decoder utilizes side information, modeled noise correlation and the information from previous decoded bitplanes to decode future bitplanes. One limitation is that the inter bitplane correlation is not fully explored during decoding. Although a refinement scheme is employed in [12] to utilize the bitplane correlation to update the noise distribution, thereby refining soft-input for decoding further bitplanes, the soft-input of the LDPCA decoder is fixed until successful decoding. To overcome the above limitations and improve the performance of the LDPCA codec, a novel Wyner-Ziv codec is proposed in this section to iteratively refine soft-input for each bitplane during the decoding process. The soft estimate of Wyner-Ziv coefficients is used to iteratively update the noise distribution and thereby refine the reliability of soft-input.

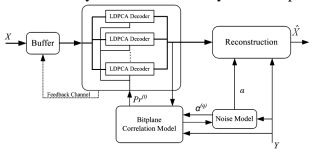


Fig. 2. Multiple LDPCA Decoders

The proposed Wyner-Ziv codec is depicted in Fig. 2. It mainly includes multiple LDPCA decoders and a bitplane correlation model. The bitplane correlation model is able to recalculate the soft-input based on the outputs of LDPCA decoders and update the estimated noise distribution from the noise model. The new soft-input information of the source X is estimated by conditioning on Y and using an iteratively refined Laplacian parameter from the noise model. The multiple LDPCA decoders are running in parallel to keep refining the soft-input. Each LDPCA decoder is responsible for one bitplane. Different from single bitplane LDPCA decoding, where the decoder corrects errors one bitplane after another e.g., from MSB to LSB [10] or from LSB to MSB [16], the multiple LDPCA decoders operates on all available bitplanes at once and exploits the correlation between bitplanes and passes information from one bitplane to another. In addition, the soft estimate of each Wyner-Ziv coefficient is iteratively generated to update the noise distribution within the bitplane correlation model. Therefore, the soft-input for decoding is regenerated in a way that exploits the noise correlation between Wyner-Ziv coefficients and the side information coefficients.

The proposed codec employs iterative refinement at both bitplane (bit) and coefficient (symbol) levels. The overall decoding procedure using multiple LDPCA decoders executes the BP algorithm to propagate LLRs back and forth between the syndrome nodes, bit nodes, and symbol nodes [14]. Let $\beta_k = Pr(b_k=0)$ define the probability distribution for bit b_k. At bit level, the main difference between our proposed approach and [6] is that the LLR for a bit b_i ($0 \le i \le m-1$) of the i^{th} significant bitplane is computed conditioned on the binary distributions (β_k, $1-\beta_k$) of the remaining bits, b_k ($k \ne i$). This means that the LLR is calculated by using soft information of the other bits. Moreover, the order of full decoding in our approach is not restricted to follow the order of significance of bitplanes. The LLR described in (1) only uses the bits from previous successfully decoded bitplanes and decodes from MSB to LSB. Here the LLR expression is generalized for a bit b_i of bitplane i as:

$$L(b_i) = \log\left(\frac{\Pr(b_i = 0 \mid Y, \beta_{m-1},...,\beta_{i+1},\beta_{i-1},...,\beta_1,\beta_0)}{\Pr(b_i = 1 \mid Y, \beta_{m-1},...,\beta_{i+1},\beta_{i-1},...,\beta_1,\beta_0)}\right) \quad (2)$$

where β_k ($k \ne i$) are soft-input values for the same coefficient as b_i.

In order to approximate the LLR expression (2), let $Pr^{(t-1)}(b_i)$ denote the a priori probability of b_i at iteration t-1 at bit level. Note that, at bit level, $Pr^{(t-1)}(X|Y) = Pr^{(q-1)}(X|Y)$, where q-1 indicates iteration q-1 at coefficient level. The denominator and numerator of (2) are substituted by applying the sum-product expressions [14][17] for specific values of $b_i = \{0,1\}$ and consequently, LLR can be computed via the sum-product algorithm [14][17] as :

$$L^{(t)}(b_i) = \log\left(\frac{\sum_{X \in S_0}\left(\Pr^{(q-1)}(X \mid Y)\prod_{k \ne i}\Pr^{(t-1)}(b_k)\right)}{\sum_{X \in S_1}\left(\Pr^{(q-1)}(X \mid Y)\prod_{k \ne i}\Pr^{(t-1)}(b_k)\right)}\right) \quad (3)$$

where $X=(b_{m-1},\ldots b_i,\ldots,\ b_1,\ b_0)$, S indicates the set of values $\{0,1,2,\ldots,2^m\text{-}1\}$ for the coefficient X which is coded by m bitplanes (for DC and the magnitude of AC coefficients) and $S_0=\{X \in S: b_i=0\}$, $S_1=\{X \in S: b_i=1\}$. $Pr^{(q-1)}(X|Y)$ is calculated at iteration q-1 at coefficient level by using the updated noise distribution between the side information coefficient and the original Wyner-Ziv coefficient via the noise model [10] as shown in Fig. 2.

Similar to bit level, we can rewrite the expression at coefficient level. Let us have an a priori belief of X conditioning on Y given by the probability distribution $Pr^{(q-1)}(X|Y)$ and variables ($\beta_{m-1},\ldots,\ \beta_1,\ \beta_0$), with likelihood $Pr^{(q-1)}(\beta_{m-1},\ldots,\ \beta_1,\ \beta_0 \mid \psi)$, where $\psi = Pr(X|Y)$, then the posterior probability $Pr^{\eta(q)}(X|Y)$ is approximated by:

$$\Pr^{\eta(q)}(X \mid Y) \propto \Pr^{(q-1)}(X \mid Y)\Pr^{(q-1)}(\beta_{m-1},...,\beta_1,\beta_0 \mid \psi) \quad (4)$$

Suppose that prior beliefs of (ψ, $\beta_{m-1},\ldots,\ \beta_1,\ \beta_0$) are independent, we get an approximation of (4):

$$\Pr^{\eta(q)}(X \mid Y) \propto \Pr^{(q-1)}(X \mid Y)\prod_k \Pr^{(q-1)}(b_k) \quad (5)$$

Thereafter $Pr^{\eta(q)}(X|Y)$ is normalized and used to update the noise residual coefficient $R^{(q)}$ at iteration q by:

$$R^{(q)} = \left| \sum_{X \in S} X \Pr^{\eta(q)}(X \mid Y) - Y \right| \quad (6)$$

A Laplacian distribution with parameter α is used to model the noise between X and Y. With the updated residue $R^{(q)}$ in (6), the

Laplacian parameter $\alpha^{(q)}$ is refined according to the noise model in [10]. The resulting soft estimate of Wyner-Ziv coefficient X is denoted:

$$\Pr^{(q)}(X \mid Y) = \Pr(X \mid Y, \alpha^{(q)}) \quad (7)$$

Since all LDPCA decoders are running in parallel, once a bitplane is successfully decoded, instantaneously, the re-initialization procedure is performed. The new soft-inputs for the rest of the bitplanes are assigned conditional on the successfully decoded bitplane. The LDPCA decoders with the successfully decoded bitplane will no longer request syndromes from the buffer. Assume b_i is successfully decoded with value *0*, then $Pr^{(t)}(b_i=0)=1$ and the iteration count is reset as $t=0$. In addition, the remaining unfinished bitplanes (b_j, $j \neq i$) are re-initialized by $Pr^{(0)}(b_j=0)=1/2$. The LDPCA decoders are iteratively operated up to a maximum numbers of iterations (T_{max}) with the given syndrome bits. If they are not successful after T_{max} iterations at bitlevel, the soft estimate of source X is iteratively updated as in (7). Furthermore, if they are not successful after a maximum number of iterations (Q_{max}) at coefficient level either, the LDPCA decoders request more syndromes (one for each of the bitplanes not fully decoded yet) from the buffer via the feedback channel. Thereafter a new process is started until all the bitplanes of the DCT coefficients of the band are successfully decoded.

In some cases, the required number of syndromes consumed for the LSB is (close to) a maximum number of syndromes denoted by N_{max}, even though there is some correlation. This is due to a (relative) loss in the LDPCA decoder, which may be reduced by first coding the LSB independently and thereafter apply the proposed codec to the remaining bitplanes after decoding the LSB. Thus, an entropy prediction mechanism is proposed to automatically predict these cases. A set of predefined thresholds is utilized to evaluate (up to 3) less significant bitplanes. The evaluation starts from LSB with its marginalized probabilities. For the LSB bitplanes considered, the entropy is estimated based on the updated LLRs from the output of the multiple LDPCA decoders after trying to decode by using the first syndrome, i.e. $n=1$. The predefined thresholds are experimentally determined to detect bitplanes for which the average estimated entropy of each bit is close to 1. If the estimated entropy of the LSB is larger than its corresponding threshold, the bitplane will be independently decoded. Then the second LSB will be evaluated based on the conditional probabilities and so on. As a result, the coding efficiency in terms of bit-rate is improved. If no LSB bitplanes are decoded first, the basic iterative multiple LDPCA decoding is handled as follows for each band, one at a time:

1. **Initiate parameters**. Number of syndromes n=0; Iteration count: $q=0$ at coefficient level, $t=1$ at bit level; For all bits b_i, $Pr^{(0)}(b_i=0)=1/2$.
2. **Increase and check conditions**.
   a. **Syndrome bit condition**: Increase $n=n+1$. If $n \geq N_{max}$ then end, else request a new syndrome for all bitplanes not decoded and continue to Step 2.b.
   b. **Iteration count condition at coefficient level**: Increase $q=q+1$. If $q \geq Q_{max}$ return to Step 2.a, else go to Step 3.
3. **Compute the LLRs.** For each bitplane, (3) is computed to get the LLRs, $L^{(t)}(b_i)$, which are forwarded as input to the multiple LDPCA unit for parallel decoding.
4. **Check for each bitplane if the LDPCA is successfully decoded?**
   a. **No: Compute probabilities of bitplanes**. New probabilities of bitplanes, $Pr^{(t)}(b_i)$, are obtained based on the updated LLRs output by the LDPCA.
   b. **Yes: Re-initialize the process**. Assume LDPCA (b_i) is successfully decoded with value $b_i=0$, assign $Pr^{(t)}(b_i=0)=1$. Reset iteration count $t=0$ and the remaining unfinished LDPCA decoders by $Pr^{(0)}(b_j=0)=1/2$.
5. **Iteration counts at bit level.** Increase $t=t+1$. If $t<T_{max}$ return to Step 3, else go to Step 6.
6. **Compute the soft estimate of source X at coefficient level.** The soft estimate, $Pr^{(q)}(X|Y)$, is updated by (7), where the noise $\alpha^{(q)}$ is computed with the updated residue based on (6).
7. **Check all LDPCA decoders**. The process is ended if all bitplanes are successfully decoded, otherwise, return to Step 2.b.

The above procedure is repeated for all bands of the DCT coefficients for which Wyner-Ziv bits are transmitted.

4. PERFORMANCE EVALUATION

In this section, the RD performance of the proposed approach is presented and compared with the TDWZ video codec described in Section 2 as well as relevant benchmarks. The test sequences are 149 frames of *Foreman*, *Hall Monitor*, *Soccer*, and *Coast-guard* with 15Hz frame rate and QCIF format. GOP (group of pictures) size is 2, where the odd frames are coded as key frames using H.246/AVC Intra and the even frames are coded using Wyner-Ziv coding. Eight RD points (Q_j) are considered corresponding to eight 4x4 quantization matrices [6]. The values within these matrices determine the number of bitplanes associated to the DCT coefficient bands, therefore, the number of LDPCA decoding instances is known. The proposed scheme uses m (number of bitplanes of a given band) regular LDPC accumulate decoders [15] with a length of 1584 bits for each. So 1584 transform coefficients per given band of a frame are decoded in parallel at a time by m LDPCA decoders each decoding one bitplane.

Table 1 shows rate and PSNR values of the proposed TDWZ codec with parallel iterative decoding as well as the savings in total rate, ΔR (in %), and WZ rate, ΔR_{WZ} (in %), compared with the TDWZ codec [10]. The proposed scheme achieves a reduction of bit-rate for WZ frames up to 3.53% for *Foreman*; 5.61% for *Hall Monitor*; 4.13% for *Soccer*; 3.75% for *Coast-guard*. It can be noted that the PSNR values are the same for both the proposed scheme and TDWZ in [10]. In addition, the relative average bitrate savings for the TDWZ [10], [13], and the proposed scheme over the DISCOVER codec for WZ frames are 11.97%, 13.78%, 15.44%, respectively (by average of the Bjøntegaard metric [18] for the 4 test sequences). Overall RD performance of the proposed scheme is depicted in Figs. 3-4. It can be seen that RD performance has been significantly improved compared with the DISCOVER codec. The performance of H.264/AVC Intra coding and No Motion Inter coding are also included. It can be noticed that the TDWZ video coding with the proposed scheme gives a better RD performance than H.264/AVC Intra coding for some sequences, e.g. *Hall Monitor* and *Foreman*, but remain worse than H.264/AVC no motion Inter coding for most of the test sequences. However, the gaps between no motion Inter coding and TDWZ are significantly reduced.

5. CONCLUSION

A Wyner-Ziv video codec with parallel iterative LDPC decoding is discussed in this paper. The technique takes bitplane correlation

into account by iteratively refining the soft-input for each bitplane and updating the noise distribution during decoding. Experimental results show that the proposed scheme can improve the coding efficiency of TDWZ in terms of WZ rate savings up to 5.6% compared with the available TDWZ video codec [10] and provide better RD performance than the DISCOVER codec.

6. REFERENCES

[1] A. Aaron, R. Zhang, and B. Girod, "Wyner-Ziv coding of motion video," *Proc. Asilomar Conference on Signals and Systems*, Pacific Grove, CA, Nov. 2002.

[2] R. Puri, A. Majumdar, and K. Ramchandran, "PRISM: a video coding paradigm with motion estimation at the decoder," *IEEE Trans. On Image Proc.*, vol. 16 (10), pp. 1-13, Oct. 2007.

[3] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. on Inform. Theory*, vol. 19 (4), pp. 471–480, Jul. 1973.

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

[5] A. Aaron, S. Rane, E. Setton, and B. Girod, "Transform domain Wyner-Ziv codec for video," *Proc. SPIE VCIP*, San Jose, CA , USA, Jan. 2004.

[6] DISCOVER Project, www.discoverdvc.org, Dec. 2007.

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

[8] X. Huang and S. Forchhammer, "Improved side information generation for distributed video coding," *IEEE Int'l Workshop Multimedia Signal Proc.*, Cairns, Australia, Oct. 2008.

[9] X. Huang, J. Ascenso, C. Brites, F. Pereira and S. Forchhammer, "Distributed video coding with multiple side information," *Picture Coding Symposium*, Chicago, USA, May 2009.

[10] X. Huang and S. Forchhammer, "Improved virtual channel noise model for transform domain Wyner-Ziv video coding," *IEEE Int'l Conf. on Acoustics, Speech, and Signal Proc.*, Taipei, Taiwan, ROC. April 2009.

[11] R. Martins, C. Brites, J. Ascenso, F. Pereira, "Refining Side Information for Improved Transform Domain Wyner-Ziv Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19 (9), pp. 1327-1341, Sep. 2009.

[12] X. Huang and S. Forchhammer, "Transform domain Wyner-Ziv video coding with refinement of noise residue and side information," *SPIE Visual Communications and Image Processing*, HuangShan, China, July 2010.

[13] H. Luong, X. Huang and S. Forchhammer, "Multiple LDPC decoding using bitplane correlation for Transform Domain Wyner-Ziv video coding", *IEEE Int'l Conf. on Acoustics, Speech, and Signal Proc.*, April 2011.

[14] D. Varodayan, D. Chen, M. Flierl, B. Girod, "Wyner-Ziv coding of video with unsupervised motion vector learning," *EURASIP Signal Processing, Image Communication*, vol. 23 (5), pp. 369-378, 2008.

[15] D. Varodayan, A. Aaron, B. Girod, "Rate-adaptive codes for distributed source coding," *EURASIP Signal Processing*, vol. 23 (11), pp. 3123-3130, 2006.

[16] Y. Vatis, S. Klomp, J. Ostermann, "Inverse Bit Plane Decoding Order for Turbo Code based Distributed Video Coding", *Int'l Conf. on Image Process.*, San Antonio, USA, September 18-21, 2007

[17] F.R. Kschischang, B.J. Frey, H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47 (2), pp. 498-519, 2001

[18] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," VCEG Contribution VCEG-M33, April 2001.

Table 1. Total rate and WZ rate savings (in %) for proposed scheme compared with TDWZ in [10]

Q_j	Foreman				Hall				Soccer				Coast-guard			
	Rate [kbps]	PSNR [dB]	ΔR [%]	ΔR_{WZ} [%]	Rate [kbps]	PSNR [dB]	ΔR [%]	ΔR_{WZ} [%]	Rate [kbps]	PSNR [dB]	ΔR [%]	ΔR_{WZ} [%]	Rate [kbps]	PSNR [dB]	ΔR [%]	ΔR_{WZ} [%]
1	72.01	28.67	0.95	2.57	84.21	31.55	0.79	5.61	57.69	28.01	2.75	4.13	81.32	28.59	0.75	3.75
2	86.25	29.36	1.45	3.53	94.55	32.13	0.87	4.80	70.81	28.60	2.65	3.86	99.57	29.32	0.61	2.65
3	95.88	29.87	1.28	3.10	97.44	32.15	0.81	3.95	81.39	29.41	2.32	3.48	102.86	29.36	0.65	2.57
4	151.90	32.44	1.00	2.42	131.26	34.42	0.72	3.41	134.70	32.04	1.81	2.85	152.42	31.15	0.75	2.76
5	157.88	32.50	1.17	2.69	133.99	34.42	0.75	3.28	139.99	32.09	1.91	2.95	168.23	31.69	0.82	3.15
6	202.65	33.74	1.51	3.26	167.55	36.03	0.89	3.43	182.68	33.16	2.12	3.23	223.18	32.98	1.21	4.22
7	271.13	35.90	1.03	2.25	195.38	37.43	0.89	3.30	250.07	35.16	1.45	2.29	265.04	33.87	1.02	2.95
8	432.61	39.31	0.79	1.62	289.88	40.84	1.16	4.06	447.12	39.02	1.18	1.99	437.98	37.04	0.89	2.22

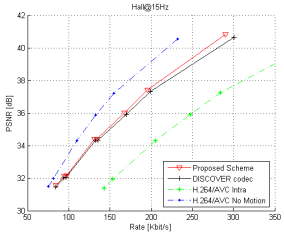


Fig. 3. RD performance comparison on Hall

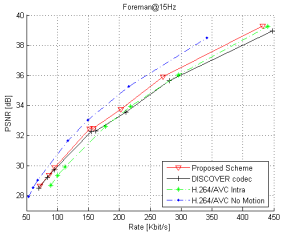


Fig. 4. RD performance comparison on Foreman

Adaptive Noise Model for Transform Domain Wyner-Ziv Video using Clustering of DCT Blocks

Huynh Van Luong, Xin Huang, Søren Forchhammer

DTU Fotonik, Technical University of Denmark
Building 343, Lyngby 2800, Denmark
{hulu,xhua,sofo}@fotonik.dtu.dk

***Abstract*—The noise model is one of the most important aspects influencing the coding performance of Distributed Video Coding. This paper proposes a novel noise model for Transform Domain Wyner-Ziv (TDWZ) video coding by using clustering of DCT blocks. The clustering algorithm takes advantage of the residual information of all frequency bands, iteratively classifies blocks into different categories and estimates the noise parameter in each category. The experimental results show that the coding performance of the proposed cluster level noise model is competitive with state-of-the-art coefficient level noise modelling. Furthermore, the proposed cluster level noise model is adaptively combined with a coefficient level noise model in this paper to robustly improve coding performance of TDWZ video codec up to 1.24 dB (by Bjøntegaard metric) compared to the DISCOVER TDWZ video codec.**

I. INTRODUCTION

In recent years, the conventional video coding architecture has been challenged by some emerging applications such as video surveillance, video sensor networks and wireless cameras etc. In contrast to conventional downstream applications, these applications are rather relying on an upstream model. In this case, many clients such as mobile communication devices with limited resources are transmitting data to a network. These applications require a relative low cost encoder with high coding efficiency. Distributed Video Coding (DVC) [1][2] is proposed to match the low cost encoding requirement, by exploring the video statistics, partially or totally, at the decoder only. DVC is an interesting instance of distributed source coding. According to the Slepian-Wolf theorem [3], it is possible to achieve the same rate by independently encoding but jointly decoding two statistically dependent signals as for typical joint encoding and decoding (with a vanishing error probability). The Wyner-Ziv theorem [4] extends the Slepian-Wolf theorem to the lossy case, becoming the theoretical basis for DVC where source data are lossy coded and decoded based on a correlated source at the decoder providing the so-called side information. Distributed video coding also provides a flexibility between encoder and decoder(s) which may be useful in a distributed environment.

Transform Domain Wyner-Ziv (TDWZ) [5] video coding is one popular approach to DVC. The decoder needs to have the correlation noise between corresponding source and the side information obtained through the frame interpolation at the decoder side. Therefore, the noise model is one of the most important aspects influencing the coding efficiency. The Laplacian distribution is commonly used for noise modelling [6]-[8]. The real noise reveals that it is not equally distributed across the frame even within one frequency band in the transform domain. In other words, the noise distribution is different for each object. For the adaptive noise modelling [6]-[8] uses both frame level and coefficient level estimates. The accuracy highly depends on the individual estimated noise residue. As a result, noise models may have less confidence when moving from the band level to the coefficient level especially when the estimated noise residue is not accurate enough. Inspired by the above, the noise distribution could be more accurately estimated based on classified blocks.

Related work on noise models is presented in [9]-[11]. These techniques consider different classes of correlation noise for each band of the side information. The technique in [9]-[10] estimates the correlation noise by differentiating blocks within a frame based on the accuracy of the side information. A residual energy between source and side information of a given block is used to classify blocks to classes by given thresholds. The Laplacian parameter is assigned through a lookup table once the block class is determined. The drawback of this work is that it only uses the residual energy information obtained between forward and backward interpolation based on the initial side information of a given block to classify blocks. The more reliable information of already decoded bands is not utilized at all. The thresholds defining the classes and the corresponding lookup table for decoder side use was obtained offline by using the training set. In [11], the reconstructed bands were used to influence the estimation for subsequent bands by classifying the reconstructed band into two categories. The bands were decoded conditioning on previous bands. The method in [11] has the disadvantage that it does not use the correlation of all bands but only 1-2 already decoded neighboring bands. Furthermore, two categories may not be enough to fully utilize the correlation.

In this paper, we propose a novel approach to adaptively estimate the Laplacian parameter by using clustering of DCT blocks. The clustering algorithm not only utilizes correlation over all frequency bands in a wise way but it also takes decoded bands into account. The intuition here is that the cross-band correlation and the successfully decoded

978-1-4577-1434-4/11/$26.00 ©2011 IEEE

information can significantly influence the reliability of block classification and consequently the accuracy of noise parameter estimation of subsequent bands. Furthermore, in order to take advantage of adaptive correlation noise modeling, the proposed noise model is combined with the noise model in [11] to adaptively optimize the soft side information for LDPCA decoding. The two noise models compete which could be implemented in a distributed environment. The rest of this paper is organized as follows. In Section II, the architecture of a state-of-the-art TDWZ video codec is presented. We describe our proposed noise model in Section III. In Section IV, the performance of the proposed method is evaluated and comparisons are presented as well.

II. State-of-the-Art Transform Domain Wyner-Ziv Video Coding

The architecture of a TDWZ video codec [11] is depicted in Fig. 1. In this system, the frame sequence is split into key frames and so-called Wyner-Ziv frames. Key frames are intra coded using conventional video coding techniques such as H.264/AVC intra coding. The Wyner-Ziv frames are transformed, quantized and decomposed into bitplanes. Each bitplane is fed to a rate-compatible LDPC Accumulate (LDPCA) encoder [12] from most significant bitplane (MSB) to least significant bitplane (LSB). Corresponding error correcting information is stored in a buffer. The amount of information to be transmitted depends on the requests made by the decoder through a feedback channel. The Wyner-Ziv frame is predicted at the decoder side by using already decoded frames as references [8]. The predicted frame, called the Side Information (SI) frame, is an estimation of the original Wyner-Ziv frame available at the encoder. Given the available SI, soft-input information (conditional bit probabilities *Pr*) within each bitplane is estimated using a noise model. Thereafter the LDPCA decoder starts to decode the various bitplanes, ordered from MSB to LSB, to correct the bit errors. After all the bitplanes are successfully decoded, the Wyner-Ziv frame can be decoded through combined de-quantization and reconstruction followed by an inverse transform.

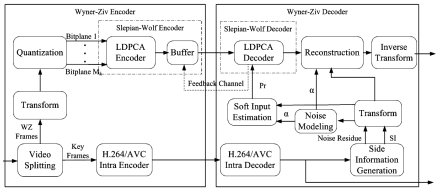


Fig. 1. Architecture of feedback channel based Transform Domain Wyner-Ziv video codec

III. Wyner-Ziv Coding with Adaptive Noise Model

As described in Section II, in order to take advantage of the side information for decoding, the Wyner-Ziv decoder needs an accurate noise model describing the distribution of the difference between the original Wyner-Ziv frame and the SI frame. An online noise model using clustering of DCT blocks is introduced below. Thereafter, this model is extended by adaptively selecting this model or the model in [11] for each bitplane.

A. Online noise model using clustering of DCT blocks

The motivation for online noise model parameter estimation using clustering of DCT blocks is that the coefficients in each cluster are closely correlated, assuming the clusters of the estimated residues are reliable enough. The Laplacian distribution estimated within each cluster is utilized to model the correlation noise in Wyner-Ziv video coding. The residual frame calculated as the difference between the motion compensated previous and the next key frames is first constructed. Often, the variance [13] of the correlation noise is estimated from the variance of the residual frame at different granularity levels: frame, block [11], and coefficient. In this paper, we first replace these levels with a cluster level obtained by clustering the DCT blocks. Assume that the residual frame is classified into non-overlapping sub-sets. The variance of the residual frame that is an approximated residual between the original Wyner-Ziv frame X and the side information frame Y is higher than the expected variance of the sub-sets (see Appendix B). This means that the estimation at cluster level should be more accurate than at frame level.

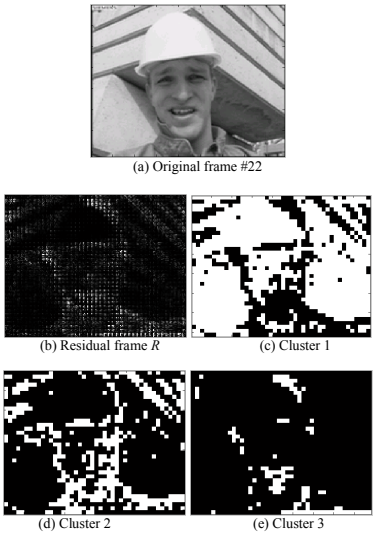
Let R be the residual frame in the transform domain. R is obtained by applying a 4-by-4-size block-based discrete cosine transform over the residual frame. R_k is used to indicate the k^{th} block of N 4-by-4-size blocks of R, $1 \leq k \leq N$. Each block R_k, considered as a feature vector, contains 16 frequencies given by the transformed residual coefficients. Consider the k^{th} block of band l and let R_k^l ($1 \leqslant l \leqslant 16$) and $\hat{R}_k^l$ denote the initial coefficient of the residual and a refined coefficient based on the partially decoded information, respectively.

The proposed method considers the transformed residual of frequency bands in a block as components of a vector and classifies each block into one of M categories by using the unsupervised clustering algorithm called Fuzzy-C means clustering [14]. Once a frequency band is successfully decoded, the information of vectors for all blocks is updated. Then the clustering algorithm is called again to classify for noise parameter estimation of the next band. This means that the proposed method not only correlates information of all bands in a precise way but takes decoded bands into account in the conditional decoding. In addition, the proposed noise model uses all bands of a block to create a feature vector for classification in order to take advantage of spatial correlation among blocks.

The proposed noise model consists of 3 steps as follows:

1) *Clustering of DCT blocks:* Our block clustering algorithm is working on a set of N feature vectors R_k, where $R_k = (R_k^1, R_k^2, ..., R_k^{16})$ is initialized based on the motion estimated residual before decoding the first band DC and $R_k = (\hat{R}_k^1, \hat{R}_k^2, ..., \hat{R}_k^{l-1}, R_k^l, R_k^{l+1}, ..., R_k^{16})$ is the updated residual based on successfully decoded bands (up to band *l-1*) before decoding band *l*. The set is separated into M clusters by using Fuzzy-C means clustering [14] (as applied, the algorithm is described in Appendix A). Let R_{kj}^l denote the coefficients of feature vectors and α_j^l denote the Laplacian noise distribution parameter of a cluster j ($1 \le j \le M$) containing N_j elements of band l, where $\sum_j N_j = N$.

Fig. 2 illustrates an example of clustering of DCT blocks. The original WZ frame (a) is frame #22 of *Foreman* sequence. The residual frame R (b) before decoding the first band DC in the transform domain is estimated at the decoder side. Then the residual is classified into 3 clusters ($M=3$) that are presented by binary masks: (c) Cluster 1, (d) Cluster 2, and (e) Cluster 3, respectively.



(a) Original frame #22

(b) Residual frame R (c) Cluster 1

(d) Cluster 2 (e) Cluster 3

Fig. 2. (a) Original frame, (b) Residual frame R in the transfrom domain, (c) Cluster 1, (d) Cluster 2, and (e) Cluster 3, respectively, of *Foreman* frame #22 before decoding the first band DC with M=3

2) *Noise parameter estimation:* In one band l, noise parameters α_j^l are obtained based on the observation for each cluster j of the band. This means α_j^l is estimated for the Laplacian distribution of a cluster j containing N_j elements of given band l by utilizing the variance σ_j^{l2} as:

$\alpha_j^l = \sqrt{2}/\sigma_j^l$, where $\sigma_j^l = \sqrt{E[|R_{kj}^l|^2] - E^2[|R_{kj}^l|]}$ where $E[.]$ is the expectation operator. As a result, there are M noise parameters for the M clusters in a given band l.

3) *Updating:* The bands are decoded in a zig zag order starting from DC and traversing AC coefficients following the order in [11]. Whenever a band l is successfully decoded, the coefficients of the band are reconstructed. This means that the set of feature vectors is now updated as $R_k = (\hat{R}_k^1, \hat{R}_k^2, ..., \hat{R}_k^{l-1}, \hat{R}_k^l, R_k^{l+1}, ..., R_k^{16})$ before decoding band $l+1$. The set of updated feature vectors is used to reclassify by going back to Step 1 and then update α_j^{l+1} for the next band $l+1$ to be decoded as Step 2 above. When all bands are successfully decoded, the algorithm is completed.

The proposed noise model is working on cluster level. This can achieve competitive results compared to the coefficient level noise model in [11] (as shown in Table I in Section IV). By experimental observations, performance of the proposed noise model may be worse than the one in [11] for some individual frames. The reason is that the estimation is more accurate at coefficient level in the case that the estimated noise residue highly depends on coefficient values. In order to take fully advantage of both of the noise models, an adaptive noise model, which adaptively combines cluster level and coefficient level noise models, is proposed in the next subsection.

B. TDWZ with adaptive noise model

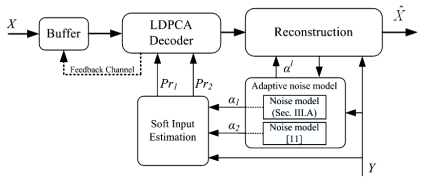


Fig. 3. TDWZ with adaptive noise model

The proposed method adaptively estimates the noise distribution by combining both the above noise model (Sec. III.A) and the noise model [11] as illustrated in Fig. 3. This combination creates two different noise parameters α_1 and α_2 as input to the Soft Input Estimation block. The LDPCA module tries to decode both options of soft side information Pr_1 and Pr_2. The LDPCA then selects the soft side information that converges first during decoding for

each bitplane. In other words, the decoder adaptively optimizes the consumed bits for decoding. In addition, the chosen noise model for one specific bitplane is also used for the minimum mean squared error reconstruction process [15]. The adaptive noise model takes advantage of both the noise models (Sec. III.A) and [11] where the optimized selection is used for both LDPCA decoding and reconstruction. As a consequence, the cluster level and coefficient level noise model can compensate each other's weaknesses and achieve gains both in bit-rate and PSNR performance.

IV. Performance Evaluation

In this section, the Rate Distortion (RD) performance of the proposed noise model is evaluated and compared to the method in [11] and the noise model in [9]. The test sequences are 149 frames of *Foreman*, *Hall Monitor*, *Soccer*, and *Coast-guard* with 15Hz frame rate and QCIF format. GOP (group of pictures) size is 2, where odd frames are coded as key frames using H.246/AVC Intra and even frames are coded using Wyner-Ziv coding. Eight RD points are considered corresponding to eight 4x4 quantization matrices [6]. The number of clusters (*M*) used in both the proposed method and [9] is 8. In this paper, the Fuzzy-C means clustering as described in Appendix A is configured with the fuzzification degree *m=2* and the predefined termination threshold ε =0.0001.

Table I and Table II show the relative average bitrate savings and equivalently the average PSNR improvement (using the Bjøntegaard metric [16] and fitting a curve through the 8 RD points measured) over the DISCOVER codec for WZ frames and overall frames, respectively. The improvements are reported for the TDWZ (described in Section II) with the noise model in [9] (TDWZ offline), the coefficient level noise model in [11], and the proposed noise model. The average bitrate saving for the proposed noise model is up to 24.57% and 13.74% (or equivalently the average improvement in PSNR is up to 1.24 dB and 0.69 dB) for WZ frames and overall frames, respectively, for the difficult *Soccer* sequence. In general, the performance of the proposed noise model in Sec. III.A is competitive with the noise model in [11] and robustly better than using the noise model in [9]. For both the relative bitrate saving and PSNR improvement, the adaptive noise model introduced in Section III.B is robustly better than the others.

The overall RD performance of TDWZ with different noise models is illustrated in Fig. 4. The performance of H.264/AVC Intra coding and No Motion Inter coding are also included. It can be noticed that the TDWZ video coding with the proposed noise model gives a better RD performance than H.264/AVC Intra coding for *Foreman*, *Hall Monitor*, and *Coast-guard*, but it is still not as good as H.264/AVC no motion Inter coding for most of the test sequences. The RD performance of TDWZ with the proposed noise model clearly outperforms those of [9] (TDWZ offline), [11], and DISCOVER.

V. Conclusion

This paper proposes an adaptive noise model for Wyner-Ziv video codec using clustering of DCT blocks. The technique not only utilizes correlation over all frequency bands but takes advantage of decoded bands to influence the decoding of subsequent bands. Experimental results show that the coding efficiency of the proposed noise model using clustering of DCT blocks is competitive with the coefficient level noise model. Moreover, the proposed adaptive noise model can significantly improve the RD performance of TDWZ compared to the existing TDWZ noise models [9],[11]. The average bitrate savings of TDWZ using the adaptive noise model is up to 24.57 % (equivalent average improvement in PSNR is up to 1.24 dB) over the DISCOVER codec.

Acknowledgment

The work presented is funded by the Danish Research Council (FTP Nr. 274-09-0249).

Appendix A.

The fuzzy C-means (FCM) clustering algorithm [14]: Consider a given finite set R, with elements $R_k \in \Re^{16}$ i.e. the set of 16-dimensional real numbers called the feature space, i.e. $R = \{R_1, R_2, ..., R_N\}$ with feature vectors $R_k = (R_k^1, R_k^2, ..., R_k^{16})$. Let $V = (V_1, V_2, ..., V_M)$ be the cluster centres, $V_i \in \Re^{16}$. A feature vector R_k belongs to a specific cluster V_i that is given by the membership value u_{ik} which can be represented by a matrix $U \in \Re_{MN}$, where $\Re_{MN}$ is the set of real $M \times N$ matrices.

The FCM algorithm iteratively optimizes the standard FCM objective function defined as:

$$J_m(U,V) = \sum_{k=1}^{N}\sum_{i=1}^{M}(u_{ik})^m (d_{ik})^2$$

where $d_{ik}^2 = \|R_k - V_i\|^2$ represents the squared Euclidean distance between the feature vector R_k and centre V_i, $m \geq 1$ is the degree of fuzzification. The optimization is initiated using the constraint $\sum_{i=1}^{M} u_{ik} = 1$.

Local minimization of the objective function $J_m(U,V)$ is accomplished by iteratively adjusting the values of u_{ik} and V_i according to the following equations:

$$u_{ik} = \frac{1}{\sum_{j=1}^{M}\left(\frac{d_{ik}}{d_{jk}}\right)^{2/(m-1)}}$$

$$V_i = \frac{\sum_{k=1}^{N} (u_{ik})^m R_k}{\sum_{k=1}^{N} (u_{ik})^m}$$

As J_m is iteratively minimized, V_i becomes more stable. Iteration of feature vector groupings is terminated at iteration t when the termination measurement $\max_{1\le i\le M}\left\{\left\|V_i^{(t)} - V_i^{(t-1)}\right\|\right\} < \varepsilon$ is satisfied, where $V_i^{(t)}$ is an updated centre, $V_i^{(t-1)}$ is the previous centre, and ε is the predefined termination threshold. Finally, all feature vectors are classified into clusters by assigning a feature vector R_k to the cluster V_j for $u_{jk} = \max_{1\le i\le M}\{u_{ik}\}$. The FCM algorithm converges to a minimum or a saddle point [17].

APPENDIX B.

The cluster-based variance: The proposed method is motivated and supported by the following lemma evaluating the cluster-based variance.

Lemma: Let R be a data set where R is classified into non-overlapping sub-sets. The variance σ^2 of a set R is higher than the expected variance of the sub-sets.

Proof: Assume $R=\{R_k\}, 1\le k\le N$ is separated into M clusters, for instance, cluster j ($1\le j\le M$) includes N_j elements that are denoted by $R_{j(i)}$ ($1\le i\le N_j$), where $\sum_j N_j = N$.

σ^2 and σ_j^2 are the variances of R and a set j including N_j elements $R_{j(i)}$ given j, respectively. What we need to prove is:

$$\sigma^2 \ge \frac{1}{N}\sum_j \sum_i (R_{j(i)} - E_j[R_{j(i)}])^2 \qquad (*)$$

where $E_j[.]$ is the expectation operator of elements given j, this means the elements $R_{j(i)}$ are included in a set j.

Equation (*) is equivalent to $\sigma^2 \ge \frac{1}{N}\sum_j N_j \sigma_j^2$

$$\Leftrightarrow N(E[R^2] - E^2[R]) \ge \sum_j N_j (E_j[R_{j(i)}^2] - E_j^2[R_{j(i)}]) \quad (**)$$

where $NE[R^2] = \sum_j N_j E_j[R_{j(i)}^2]$ because

$$NE[R^2] = \sum_k R_k^2 \text{ and } \sum_j N_j E_j[R_{j(i)}^2] = \sum_j \sum_i R_{j(i)}^2$$

Equation (**) is equivalent to $\sum_j N_j E_j^2[R_{j(i)}] \ge NE^2[R]$

$$\Leftrightarrow N\sum_j N_j \left(\frac{\sum_i R_{j(i)}}{N_j}\right)^2 \ge \left(\sum_k R_k\right)^2$$

$$\Leftrightarrow \left(\sum_j \left(\sqrt{N_j}\right)^2\right)\left(\sum_j \left(\frac{\sum_i R_{j(i)}}{\sqrt{N_j}}\right)^2\right) \ge \left(\sum_j \left(\sum_i R_{j(i)}\right)\right)^2$$

which is true due to the Cauchy–Schwarz inequality for any real number $N_j > 0$ and $R_{j(i)}$. The two sides are equal if and only if the ratios $\frac{\sum_i R_{j(i)}}{N_j}$ are equal.

REFERENCES

[1] A. Aaron, R. Zhang, and B. Girod, "Wyner-Ziv coding of motion video," *Proc. Asilomar Conference on Signals and Systems*, Pacific Grove, CA, Nov. 2002.

[2] R. Puri, A. Majumdar, and K. Ramchandran, "PRISM: a video coding paradigm with motion estimation at the decoder," *IEEE Trans. On Image Proc.*, vol. 16 (10), pp. 1-13, Oct. 2007.

[3] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. on Inform. Theory*, vol. 19 (4), pp. 471–480, Jul. 1973.

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

[5] A. Aaron, S. Rane, E. Setton, and B. Girod, "Transform domain Wyner-Ziv codec for video," *Proc. SPIE VCIP*, San Jose, CA , USA, Jan. 2004.

[6] DISCOVER Project, www.discoverdvc.org, Dec. 2007.

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

[8] X. Huang and S. Forchhammer, "Improved side information generation for distributed video coding," *IEEE Int'l Workshop Multimedia Signal Proc.*, Cairns, Australia, Oct. 2008.

[9] G. R. Esmaili, P. Cosman "Correlation noise classification based on matching success for transform domain Wyner-Ziv video coding," *IEEE Int'l Conf. on Acoustics, Speech, and Signal Proc.*, Taipei, Taiwan, ROC. April 2009.

[10] G. R. Esmaili, P. Cosman "Wyner-Ziv Video Coding with Classified Correlation Noise Estimation and Key Frame Coding Mode Selection," *IEEE Trans. on Image Processing*, vol. PP (99), 2011.

[11] X. Huang and S. Forchhammer, "Improved virtual channel noise model for transform domain Wyner-Ziv video coding," *IEEE Int'l Conf. on Acoustics, Speech, and Signal Proc.*, Taipei, Taiwan, ROC. April 2009.

[12] D. Varodayan, A. Aaron, B. Girod, "Rate-adaptive codes for distributed source coding," *EURASIP Signal Processing*, vol. 23 (11), pp. 3123-3130, 2006.

[13] C. Brites and F. Pereira, "Correlation noise modeling for efficient pixel and transform domain Wyner-Ziv video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, pp. 1177 – 1190, September 2008.

[14] R. L. Cannon, J. V. Dave, and J. C. Bezdek, "Efficient Implementation of the Fuzzy c-Means Clustering Algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8 (3), pp. 248–255, 1986.

[15] D. Kubasov, J. Nayak, and C. Guillemot, "Optimal reconstruction in Wyner-Ziv video coding with multiple side information," *in Proc. IEEE Int'l Workshop Multimedia Signal Process.*, pp. 183–186, Chania, Greece, Oct. 2007.

[16] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," VCEG Contribution VCEG-M33, April 2001.

[17] J. C. Bezdek, R. J. Hathaway, M. J. Sabin, and W. T. Tucker, "Convergence theory for fuzzy c-means: Counterexamples and repairs," *IEEE Trans. Syst., Man, Cybern.*, vol. 17, pp. 873–877, Sept. 1987.

TABLE I
RELATIVE BIT-RATE SAVING (%) AND PSNR DIFFERENCE (DB) OF TDWZ [9], [11], AND THE PROPOSED OVER DISCOVER FOR WZ FRAMES

Sequence	Relative bit-rate saving (%)				PSNR difference (dB)			
	TDWZ offline	TDWZ [11]	Proposed (Sec. III.A)	Proposed (Sec. III.B)	TDWZ offline	TDWZ [11]	Proposed (Sec. III.A)	Proposed (Sec. III.B)
Foreman	2.20	11.23	10.05	14.70	0.06	0.50	0.45	0.66
Hall	3.56	5.96	6.86	11.50	0.18	0.27	0.32	0.52
Soccer	-8.05	20.88	15.70	24.57	-0.60	1.06	0.82	1.24
Coast	8.67	9.80	11.64	15.36	0.25	0.29	0.35	0.47

TABLE II
RELATIVE BIT-RATE SAVING (%) AND PSNR DIFFERENCE (DB) OF TDWZ [9], [11], AND THE PROPOSED OVER DISCOVER FOR ALL FRAMES

Sequence	Relative bit-rate saving (%)				PSNR difference (dB)			
	TDWZ offline	TDWZ [11]	Proposed (Sec. III.A)	Proposed (Sec. III.B)	TDWZ offline	TDWZ [11]	Proposed (Sec. III.A)	Proposed (Sec. III.B)
Foreman	0.73	4.79	4.35	6.19	0.03	0.26	0.24	0.34
Hall	1.84	1.98	2.42	3.39	0.13	0.15	0.18	0.25
Soccer	-6.24	11.49	8.56	13.74	-0.41	0.58	0.44	0.69
Coast	3.16	3.36	4.03	4.94	0.15	0.16	0.19	0.23

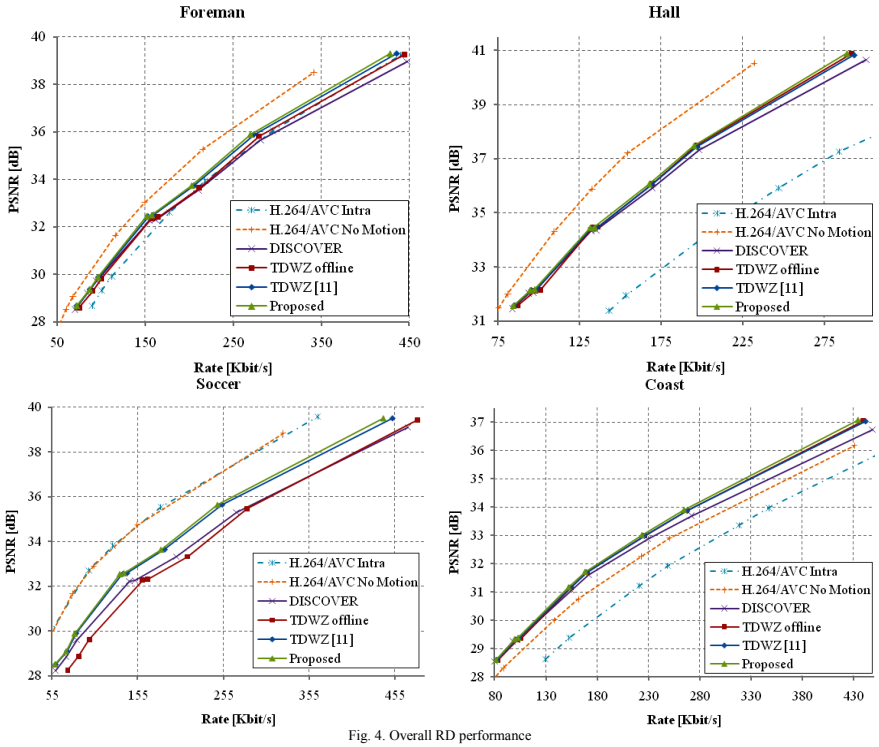


Fig. 4. Overall RD performance

Multi-hypothesis Transform Domain Wyner-Ziv Video Coding including Optical Flow

Xin Huang[1#], Lars Lau Rakêt [2*], Huynh Van Luong[#], Mads Nielsen[*], François Lauze[*], Søren Forchhammer[#]

[#] *DTU Fotonik, Technical University of Denmark,*
Building 343, Lyngby 2800, Denmark
[1]xhua@fotonik.dtu.dk

[*] *Department of Computer Science, University of Copenhagen,*
Universitetsparken 1, Copenhagen 2100, Denmark
[2]larslau@diku.dk

***Abstract*—Transform Domain Wyner-Ziv (TDWZ) video coding is an efficient Distributed Video coding solution providing new features such as low complexity encoding, by mainly exploiting the source statistics at the decoder based on the availability of decoder side information. The accuracy of the decoder side information has a major impact on the performance of TDWZ. In this paper, a novel multi-hypothesis based TDWZ video coding is presented to exploit the redundancy between multiple side information and the source information. The decoder used optical flow for side information calculation. Compared with the best available single estimation mode TDWZ, the proposed multi-hypothesis based TDWZ achieves robustly better Rate-Distortion (RD) performance and the overall improvement is up to 0.6 dB at high bitrate and up to 2 dB compared with the DISCOVER TDWZ video codec.**

I. INTRODUCTION

Distributed Video Coding (DVC) [1] provides a video coding paradigm which fully or partly exploits the temporal redundancy of video at the decoder, instead of at the encoder as in predictive video coding, thereby shifting computational requirements from encoder to decoder. This may be of interest when communicating video from mobile devices. Further the use of distributed source coding also provides flexibility on the decoder side. DVC is based on two major information theoretic results: the Slepian-Wolf [2] and Wyner-Ziv [3] theorems. The Slepian-Wolf theorem proves that, two statistically dependent discrete random sequences (X,Y) which are independently and identically distributed (i.i.d.) may be independently encoded but jointly decoded, at the same rate as for joint encoding and decoding. The Wyner-Ziv theorem extends the Slepian-Wolf theorem to lossy source coding of X based on side information Y at the decoder. This suggests that a novel video coding system, which encodes individual frames independently, but decodes them jointly, may achieve low complexity encoding with similar coding efficiency as conventional hybrid predictive video coding, notably if X and Y are jointly Gaussian and a mean-square error distortion measure is considered.

Transform Domain Wyner-Ziv (TDWZ) video coding [4] is one efficient approach to DVC. Its coding efficiency is highly dependent on the accuracy of side information at the decoder. Most TDWZ video codecs are based on a single side information estimation mode. For example, in [5]-[9], there is one soft-input estimate available at the Wyner-Ziv decoder, obtained from side information frame generation and noise modeling. Although the quality of side information frames and the accuracy of the noise model have been improved in [6][8][9], the coding efficiency of the single estimation mode TDWZ trails that of conventional video coding solutions, such as H.264/AVC, most notable in high motion sequences. Related work about multiple side information based TDWZ have been proposed in [10][11]. In [10], two different frame interpolation methods are employed, but the Wyner-Ziv decoder is only dealing with the average of two estimates for decoding and reconstruction. In [11], a set of weighted multiple soft-inputs are firstly developed in TDWZ, which is based on one frame interpolation and one frame extrapolation technique. However, the contribution brought by frame extrapolation is rather limited to the multiple soft inputs and the reconstruction in [11] was only based on one side information. Calculations of multiple side information and multi-hypothesis may also be considered in a distributed computing environment.

In order to enhance performance and reduce the RD gap between TDWZ and conventional video coding, which is especially pronounced in high motion sequences, a novel multi-hypothesis based TDWZ decoder is introduced in this paper. The multiple side information is generated by both block based and optical flow based side information generation techniques. The intuition is that optical flow based frame interpolation can generate different side information and compensate the estimation weakness in block based methods. An additional contribution of this paper is that the multiple soft-inputs for decoding and reconstruction are based on a weighted joint distribution in contrast to [11]. In this way, the proposed multi-hypothesis based TDWZ decoder will not only reduce the required bitrate for decoding but also improve the quality of reconstructed frames.

The rest of this paper is organized as follows: Section II briefly describes the state-of-the-art TDWZ video coding with single side information estimation. In Section III, the proposed Wyner-Ziv decoder is introduced. Finally, the performance results are presented in Section IV.

978-1-4577-1434-4/11/$26.00 ©2011 IEEE

II. TRANSFOM DOMAIN WYNER-ZIV VIDEO CODING WITH SINGLE SIDE INFORMATION

In a TDWZ video codec with single side information estimation mode, the frame sequence is split into key frames and so-called Wyner-Ziv frames. Key frames are intra coded using conventional video coding techniques such as H.264/AVC intra coding. The Wyner-Ziv frames X are transformed, quantized and decomposed into bit planes. Each bit plane is fed to a rate-compatible LDPC Accumulate (LDPCA) encoder [12] from most significant bit plane (MSB) to least significant bit plane (LSB). Corresponding error correcting information is stored in a buffer. The amount of bits to be transmitted depends on the requests made by the decoder through a feedback channel. The original Wyner-Ziv frame available at the encoder is estimated at the decoder side by using already decoded frames as references. An adaptive weighted Overlapped Block Motion Compensation (OBMC) based frame interpolation scheme [6] is employed to generate an estimated side information frame Y. With the obtained estimation, soft-input information within each bit plane is estimated using a cross-band based noise model [13]. Thereafter the LDPCA decoder starts to decode the various bit planes, ordered from MSB to LSB, to correct the bit errors. After all the bit planes are successfully decoded, the Wyner-Ziv frame can be decoded through combined de-quantization and reconstruction [10] followed by an inverse transform.

In single side information estimation mode TDWZ video coding, the certainty of soft-input information plays a key role in terms of overall RD performance. The soft-input Pr is defined as a conditional probability of each bit b_i being equal to 0 or 1, i.e. $Pr = P(b_i \mid y, f_{X|Y}, b^-)$, where y denotes the corresponding estimated side information value in transform domain for bit b_i, $f_{X|Y}$ is an estimated probability density function obtained from the adopted noise model and b^- is the information from the previously decoded bit planes. The quality of the reconstructed frame is highly dependent on the accuracy of the estimated noise distribution $f_{X|Y}$ [10]. The TDWZ video codec with single side information estimation mode presented in this section is considered as the best available TDWZ codec [13]. As seen in Section IV, this gives better performance than the DISCOVER video codec due to better frame interpolation [6] and noise model [13]. For a detailed description, we refer to [13].

III. MULTI-HYPOTHESIS BASED WYNER-ZIV DECODING

As described in Section II, the essential aspects to improve the coding efficiency of TDWZ are the certainty of the soft-input information fed into the LDPCA decoder and the accuracy of the noise distribution for frame reconstruction. To address these issues, a multi-hypothesis based Wyner-Ziv decoding is proposed. The Wyner-Ziv encoder is not changed, as the basic idea is to generate M (>1) different side information frames Y_k, $k \in (1,M]$, at the decoder for each Wyner-Ziv frame. Each side information frame is considered as an observation of the original Wyner-Ziv frame X with a different amount of noise. Processing multiple side information frames may reduce the bitrate required and improve the quality of reconstructed frame.

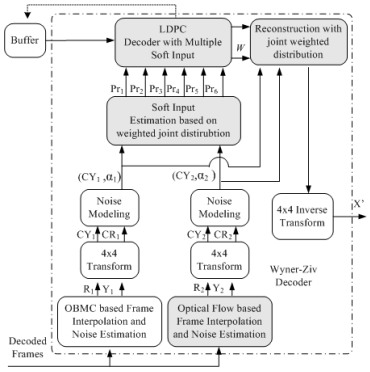


Fig. 1. Architecture of multi-hypothesis TDWZ video codec based on two frame interpolation schemes

The architecture of the proposed Wyner-Ziv video decoder with an example of two side information generation schemes (M=2) is presented in Fig. 1. In principle, there can be any number of competitive side information generation schemes at the proposed Wyner-Ziv decoder. The two different interpolation methods shown in Fig. 1 are the OBMC based frame interpolation introduced in [6] and a novel optical flow based frame interpolation method described in Section III-A. Each side information generation scheme not only creates an estimation of Wyner-Ziv frame, Y_k, but also an estimated noise residue frame R_k. R_k is used to express the correlation noise between the Wyner-Ziv frame X and its estimated side information frame Y_k. (We refer to Section III-A for more details.)

R_k is used to calculate the parameter of the noise distribution $f_{X|Y_k}$ outlined. The estimated noise residue R_k and side information frame Y_k undergo the same 4x4 block DCT. Taking its corresponding transform coefficients as inputs for a coefficient level noise model [13], the noise distribution $f_{X|Y_k}$ between the estimated side information frame Y_k and Wyner-Ziv frame X in transform domain is modeled by Laplace distributions. Each transform coefficient in a given band b_l is assigned with an estimated Laplace distribution parameter $\alpha_l^k(m,n)$, where (m,n) are the coordinates of the corresponding DCT coefficient. The value of the Laplacian parameter expresses the reliability of the corresponding estimated side information frame.

With Laplacian parameters based on different calculations of Y_k, multiple soft-inputs are calculated based on a proposed weighted joint distribution (described in Section III-B). All the hypotheses of soft-input are fed into a modified LDPCA decoder. The soft-input that converges first is chosen as the best candidate soft input for LDPCA decoding. Subsequently, the corresponding weighted joint distribution is given to the

reconstruction module for minimum mean-square error reconstruction [10]. More details about the soft-input calculation and the reconstruction based on the joint weighted distribution are described in Section III-B.

A. Optical Flow based Frame Interpolation

The goal of optical flow estimation is to determine the apparent motion in a given video sequence, and the optical flow between two video frames I_0 and I_1 is the displacement field v between these, i.e. in a suitable sense

$$I_1(\mathbf{z}+v(\mathbf{z}))=I_0(\mathbf{z})\,. \tag{1}$$

where $\mathbf{z}\in\Re^2$ denotes the image coordinates.

In recent years a large number of techniques have been proposed to determine optical flow, and the accuracy of optical flow algorithms has improved tremendously [14]. We use an optical flow algorithm that determines the optical flow by minimizing the energy functional *OF* given by

$$OF(v)=\lambda\int\|I_1(\mathbf{z}+v(\mathbf{z}))-I_0(\mathbf{z})\|_2\,\mathrm{d}\mathbf{z}+\int\|\nabla v(\mathbf{z})\|_2\,\mathrm{d}\mathbf{z}\,. \tag{2}$$

This functional consists of a robust L^1 norm of the data fidelity term (1) and a total variation (TV) regularization of the flow v. The parameter λ determines the tradeoff between data fidelity and regularity of the estimated optical flow, and has been set to 10 in all experiments. The minimizing flow is calculated using a highly efficient method, relying on tools from non-smooth convex analysis in a coarse-to-fine pyramid setup, which was introduced in [15], and further improved in [16]. Here we use the very efficient GPU implementation described in [17], which allows for the inclusion of higher order terms in the frames I_0 and I_1, e.g. gradient information either alone or in conjunction to the luminance information. This can improve flow quality under difficult lighting conditions [18], but since the lighting is fairly constant in the test sequences, gradient information has not been included in the evaluation in Section IV. In general this method for calculating optical flow should be more robust under high motion than block based methods, as the estimation procedure is done in a spatial continuous setting and considers the frames on a number of scales, to align structures of different size. Furthermore it should handle luminance patterns undergoing strong deformation better, as neighborhood relations are less rigid when using TV regularization compared to the constraints imposed by blocks. On the other hand block based methods will typically perform better in low motion sequences. This is because the optical flow algorithm is fine-tuned for complex motion, so it will be more disposed to falsely interpreting small changes in the video frames as motion.

For the interpolation, the forward flow v_f is calculated between two consecutive decoded key frames and the backward flow v_b is calculated between the same frames in reverse temporal order. For each of the flows an estimate of the intermediate frame can be produced as the one transporting the brightness patterns half way along the flow lines between the two key frames. This has the unfortunate consequence that the interpolated frames will be blank in regions undergoing disocclusion, since the optical flow will map away from these. To avoid this problem, one can utilize the principle that first following a forward flow line, and then taking the backward flow should bring one back to the starting point

$$v_b(\mathbf{z}+v_f(\mathbf{z}))+v_f(\mathbf{z})=0\,. \tag{3}$$

This identity only holds as long as we stay away from occlusion, but ignoring this, one can translate the coordinate system by v_b for the forward flow and v_f for the backward flow, which gives that the intermediate frame interpolations Y_f and Y_b can be calculated as

$$Y_f(\mathbf{z})=I_0(\mathbf{z}+1/2\,v_b(\mathbf{z}))\quad\text{and}\quad Y_b(\mathbf{z})=I_1(\mathbf{z}+1/2\,v_f(\mathbf{z})) \tag{4}$$

in every pixel point $\mathbf{z}$, where the key frame evaluation points are rounded to nearest pixel. If a region of points in I_0 is undergoing disocclusion along the forward flow lines to Y_f, the equation (3) will not hold, but using (4) will then just assign the disoccluded region in Y_f with similar (in terms of value) nearby values in I_0, because when looking along the backward flow lines, the disocclusion will be an occlusion, and this occluded region will be mapped to a nearby region in I_0 that has similar values, since we are minimizing the data fidelity term (1). This means that we automatically have an implicit inpainting of holes in this scheme. Finally, the interpolated frame Y will be the average of Y_f and Y_b

$$Y(\mathbf{z})=\frac{1}{2}\left(Y_f(\mathbf{z})+Y_b(\mathbf{z})\right) \tag{5}$$

and the noise residue frame R is calculated as the difference between the two estimated frames

$$R(\mathbf{z})=Y_f(\mathbf{z})-Y_b(\mathbf{z})\,. \tag{6}$$

Given the forward and backward flows, this interpolation scheme is quite simple, but the process is independent of the flow calculations, so one could improve results by using a more sophisticated optical flow algorithm without altering its internal mechanics, e.g. a flow calculated using the excellent framework proposed by Xu et al. [19] would likely produce significantly better interpolations than the method used here [14].

A number of alternative flow based interpolation methods have been compared to the one presented here. Interpolation based on direct interpolation along the flow lines and subsequent calculation of Y as a locally weighted average of the two intermediate frames, with weights determined from information about occluded/disoccluded areas, proved to give results that were marginally worse than (5). The more elaborate scheme presented in [20], where motion and interpolation are estimated simultaneously has been implemented, but while it provides interpolated frames that are visually more pleasing than (5), because the forward and backward flow will converge to a common interpolation, they are quantitatively inferior in terms of difference to the real frames.

B. Multi-hypothesis Soft-input and Reconstruction based on Joint Weighted Distribution

With the obtained noise distribution $f_{X|Y_k}$ for each individual side information observation Y_k, a joint weighted distribution F is defined as:

$$F_j = \sum_{k=1}^{M} w_{jk} f_{X|Y_k} \tag{6}$$

where j, $j \in [1,N]$ denotes the index of candidate joint weighted distribution, N is the total number of joint distributions employed candidates, and w_{jk} denotes the jth predefined weighting parameter on side information k, $k \in (1,M]$ and $\sum_{k=1}^{M} w_{jk} = 1$. M is the total number of distinct side information frames available at the decoder. (As the example shown in Fig.2, M=2, N=6).

As the OBMC based frame interpolation scheme [6] gives better results on the different test sequences compared to the other side information techniques employed in this paper (shown in Table I), the soft input calculation is only based on the joint weighted distribution within a specific unreliable region specified by the set *map*. Outside of the *map* region, the side information is given by the OBMC based scheme. The values of the Laplacian parameters should express the reliability of the corresponding side information frame, thus an unreliable set S_k of each single side information estimation Y_k in band b_l can be determined by evaluating the individual Laplacian parameters and their corresponding mean value as:

$$S_k = \{(m,n) \mid \alpha_k^{b_l}(m,n) < E(\alpha_k^{b_l})\} \tag{7}$$

where $\alpha_k^{b_l}(m,n)$ is the estimated Laplacian parameter of side information Y_k at position (m,n) in band b_l and E is the expectation operator. The overall unreliable region *map* is defined as a union of the sets S_k:

$$map = \bigcup_{k=1}^{M} S_k \tag{8}$$

The multi-hypothesis soft-input are calculated as:

$$Pr_j = \begin{cases} P(b_i \mid y_1, f_{X|Y_1}, b^-), & if \quad i \notin map \\ P(b_i \mid y_1 .. y_M, F_j, b^-), & if \quad i \in map \end{cases} \tag{9}$$

where Pr_j is the jth candidate soft-input fed into LDPCA decoder, b_i denotes the ith bit in one bit plane, i is the one-dimensional presentation of the coordinate (m,n), and $y_1 .. y_M$ denote different side information values in transform domain based on diverse side information generation schemes. Particularly, y_1 and $f_{X|Y_1}$ denote the corresponding side information value for bit b_i and the estimated noise distribution based on the OBMC based frame interpolation scheme.

All the hypotheses of soft-input, $Pr_j, j \in [1,N]$, are fed into a modified LDPCA decoder. The first converging soft-input is chosen thus reducing the rate of LDPCA decoding. Subsequently, with the information of chosen soft-input, the corresponding joint weighted distribution $F_j, j \in [1,N]$, in the unreliable region *map* is determined. Given this information to a proposed joint weighted distribution based on reconstruction module, the minimum mean-square error reconstructed value, x', in the unreliable region *map* is obtained as:

$$x' = E[x \mid x \in [L,U), y_1 \ldots y_M] = \frac{\int_L^U x F_j(x) dx}{\int_L^U F_j(x) dx} = \frac{\sum_{k=1}^{M} \int_L^U x w_{jk} f_{X|Y_k}(x) dx}{\sum_{k=1}^{M} \int_L^U w_{jk} f_{X|Y_k}(x) dx} \tag{10}$$

where $[L,U)$ are decoded quantization intervals. F_j denotes the most accurate joint weighted distribution available at decoder. w_{jk} (where $\sum_{k=1}^{M} w_{jk} = 1$) are predefined weighting parameters corresponding to F_j for side information k, $k \in (1,M]$. The reconstructed value outside the *map* region is calculated following the single side information reconstruction technique as in [11].

IV. Experimental Results

In order to make a fair evaluation of the proposed Wyner-Ziv video coding, the test conditions adopted in this paper are the commonly used DISCOVER project [7] test conditions. The test sequences are *Foreman*, *Soccer*, *Coastguard* and *Hall* at 15 frames per second, QCIF, GOP size 2. Key frames are coded with H.264/AVC intra and QPs are chosen as in [7] so that the average quality of Wyner-Ziv frames is similar to the average quality of the key frames.

First of all, in order to evaluate the performance of the optical flow based frame interpolation scheme described in Section III-B, the quality of interpolated frames is measured by average Peak Signal-to-Noise Ratio (PSNR) over the set of test sequences and compared with block based frame interpolation [6] and extrapolation [11] techniques in Table I. It can be seen that the OBMC based frame interpolation method gives better performance overall. However, optical flow based frame interpolation outperforms OBMC on the high motion sequences, e.g. *Soccer*, and on the individual interpolated frames with high motion, as shown in Fig. 2. On the other hand, block based methods typically perform better in low motion sequences. Taking these diverse side information generation schemes as input, the proposed Wyner-Ziv video decoder is able to combine the different estimated side information adaptively.

The RD performance of the proposed TDWZ video coding is evaluated. Only the luminance component is coded, allowing for comparison with the DISCOVER codec [7] and

TABLE I
THE AVERAGE PSNR RESULTS FOR DIFFERENT SIDE INFORMATION GENERATION METHODS. KEY FRAMES ARE INTRA CODED WITH FIXED QP

	OBMC based Interpolation [7]	**Optical Flow based Interpolation**	**Block based Extrapolation [12]**
Foreman, QP=25	29.26	29.28	25.20
Soccer, QP=25	21.30	22.43	19.26
Coast, QP=26	31.83	30.92	28.55
Hall, QP=24	36.46	32.28	33.24

the best available single side information mode TDWZ codec [13]. The performances of benchmark codecs, H.264/AVC Intra and Inter no motion, are also included. The proposed Wyner-Ziv decoder is implemented in two different versions, by employing two frame interpolation schemes as shown in Fig. 1 (i.e. M=2) and three frame generation schemes (two frame interpolation shown in Fig. 1 plus the frame extrapolation technique employed in [12], i.e. M=3), respectively. In order to make the comparison fair, the number of candidate hypothesis is constrained to N=6 (also allowing for fair comparison with the similar complexity as in [11]). For the case M=2 and N=6, the weighting parameters used are, w_{1j}={1;0.8;0.6;0.4;0.2;0} and w_{2j}=1-w_{1j}, $j \in [1,6]$. For the case M=3 and N=6, the weighting parameters are empirically predefined as: w_{1j}={1;0;1/2;1/2;0;1/3}, w_{2j}={0;1;1/2;0;1/2;1/3}, and w_{3j}={0;0;0;1/2;1/2;1/3}.

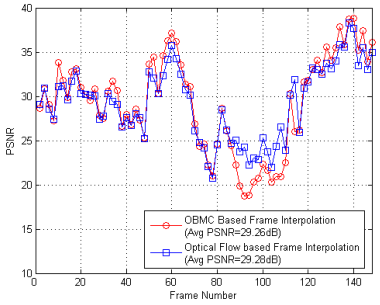


Fig. 2. PSNR for interpolated frames of Foreman, (Key frames QP=25) using OBMC and optical flow

Fig. 2 shows that the frame by frame interpolation performance may differ even though the overall performance is similar.

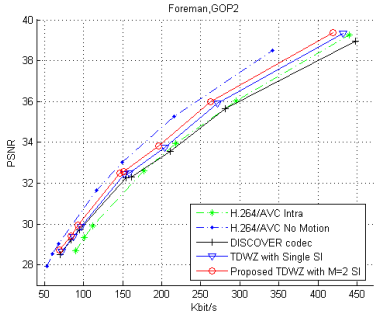


Fig. 3. Overall RD performance comparison for Foreman

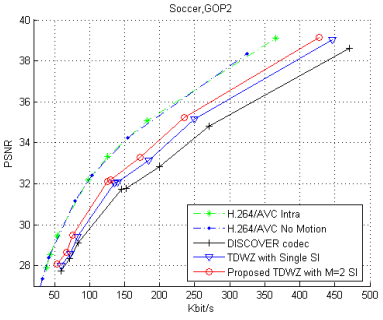


Fig. 4. Overall RD performance comparison for Soccer

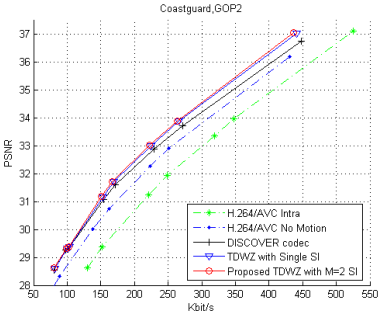


Fig. 5. Overall RD performance comparison for Coastguard

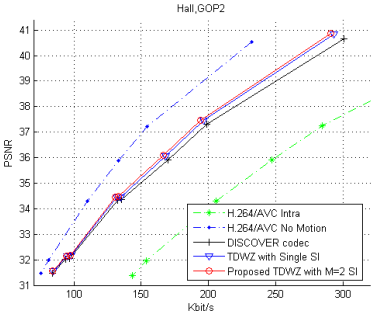


Fig. 6. Overall RD performance comparison for Hall

As shown in Figs. 3-6, the performance of the best available TDWZ with single side information [13] employed in this paper is significantly better than the DISCOVER video codec. With the proposed Wyner-Ziv video codec (with M=2 and N=6 mode), the overall RD performance of TDWZ can be improved by up to 0.6 dB at high bitrate for the sequence

Soccer. The accumulated improvement compared to the DISCOVER codec is up to 2 dB at high bitrate. Compared with H.264/AVC Intra coding, the proposed TDWZ codec gives a better RD performance for relative low motion sequences, *Foreman*, *Coastguard* and *Hall*. For the high motion sequence *Soccer*, the performance gap compared to H.264/AVC Intra coding has been substantially reduced but not eliminated yet. It is worth to note that, the proposed TDWZ significantly outperforms H.264/AVC Inter no motion coding for *Coastguard* but still is not competitive for the other test sequences.

RD improvements are also measured by average Bjøntegaard bitrate savings [21] over the DISCOVER codec and reported in Table II. It shows that the proposed Wyner-Ziv video coding scheme (either M=2 or M=3 mode) outperforms DISCOVER codec and the best available single side information mode TDWZ [13]. By adding one more frame extrapolation scheme (M=3 mode) in the proposed Wyner-Ziv decoder, the performance can be further improved. Compared to the related previous work in [11] with the same N, the proposed scheme is also better. The proposed Wyner-Ziv decoder provides larger gains for high motion sequences like *Foreman* and *Soccer* with average rate gain up to 44.5% for Wyner-Ziv frames (24.2% for overall performance) compared to DISCOVER codec. Although the OBMC based frame interpolation is quite efficient (see Table. 1) in single side information mode TDWZ for low motion sequences, e.g. *Hall*, the other side information generation methods can still contribute to RD performance improvement in the proposed multi-hypothesis Wyner-Ziv decoder.

TABLE II
BJØNTEGAARD AVERAGE BITRATE SAVINGS IN PERCENTAGE COMPARED WITH DISCOVER TDWZ VIDEO CODEC

	RD improvement measured in average bitrate savings (in %) over DISCOVER codec for WZ frames and for overall performance in parentheses			
	Best Available TDWZ with Single SI [13]	***Related Previous Work in [11]***	***TDWZ with M=2 SI***	***TDWZ with M=3 SI***
Foreman	13.3 (5.71)	20.3 (8.50)	27.0 (10.9)	31.4 (12.6)
Soccer	23.1 (12.8)	29.5 (16.5)	41.2 (22.6)	44.5 (24.2)
Coast	11.5 (3.91)	16.4 (5.22)	17.4 (5.51)	19.1 (5.96)
Hall	8.77 (2.72)	14.1 (3.88)	13.3 (3.78)	14.3 (3.94)

V. CONCLUSION

A novel multi-hypothesis TDWZ video coding including optical flow frame interpolation is proposed in this paper. The multiple side information is generated by both block based and optical flow based side information generation techniques. Multi-hypothesis soft-input is utilized for both decoding and reconstruction based on weighted joint distributions. In this way, the proposed scheme is not only able to reduce the required bitrate for decoding but also improve the quality of reconstructed Wyner-Ziv frame. Compared with the best available single side estimation mode TDWZ video coding, the overall RD performance can be improved up to 0.6 dB and up to 2 dB compared with DISCOVER TDWZ video codec.

ACKNOWLEDGMENT

The work presented is funded by the Danish Research Council (FTP Nr. 274-09-0249).

REFERENCES

[1] 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 Communication*, pp. 339-352, 2008.

[2] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. on Inform. Theory*, vol. 19 (4), pp. 471–480, Jul. 1973.

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

[4] A. Aaron, S. Rane, E. Setton, and B. Girod, "Transform domain Wyner-Ziv codec for video," *Proc. SPIE VCIP*, San Jose, CA , USA, Jan. 2004.

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

[6] X. Huang and S. Forchhammer, "Improved side information generation for distributed video coding," *IEEE Int'l Workshop Multimedia Signal Proc.*, Cairns, Australia, Oct. 2008.

[7] DISCOVER Project, www.discoverdvc.org, Dec. 2007.

[8] X. Huang and S. Forchhammer, "Improved virtual channel noise model for transform domain Wyner-Ziv video coding," *IEEE Int'l Conf. on Acoustics, Speech, and Signal Proc.*, Taipei, Taiwan, ROC. April 2009.

[9] R. Martins, C. Brites, J. Ascenso, F. Pereira, "Refining Side Information for Improved Transform Domain Wyner-Ziv Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19 (9), pp. 1327-1341, Sep. 2009.

[10] D. Kubasov, J. Nayak, and C. Guillemot, "Optimal reconstruction in Wyner-Ziv video coding with multiple side information," *in Proc. IEEE Int'l Workshop Multimedia Signal Process.*, pp. 183–186, Chania, Greece, Oct. 2007.

[11] X. Huang, J. Ascenso, C. Brites, F. Pereira and S. Forchhammer, "Distributed video coding with multiple side information," *Picture Coding Symposium*, Chicago, USA, May 2009.

[12] D. Varodayan, A. Aaron, B. Girod, "Rate-adaptive codes for distributed source coding," *EURASIP Signal Processing*, vol. 23, pp. 3123-3130, 2006.

[13] X. Huang and S. Forchhammer, "Cross-band noise model refinement for transform domain wyner-ziv video coding," *Signal Processing: Image Communication*, 2011, accepted.

[14] S. Baker, D Scharstein, J.P. Lewis, S. Roth, M.J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *International Journal of Computer Vision*, vol 31, pp. 1-31, 2011.

[15] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime TV-L^1 optical flow," *DAGM-Symposium*, 2007.

[16] A. Wedel, C. Zach, T. Pock, H. Bischof, and D. Cremers, "An improved algorithm for TV-L^1 optical flow," *Dagstuhl Motion Workshop*, 2008.

[17] L. Rakêt, L. Roholm, M. Nielsen and F. Lauze "TV-L^1 optical flow for vector valued images," *In Energy Minimization Methods in Computer Vision and Pattern Recognition,* 2011 (to appear).

[18] T. Brox, A. Bruhn, N. Papenberg and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," *Proceedings of ECCV*, vol. 4, pp. 25-36, 2004.

[19] L. Xu, J. Jia, Y. Matsushita, "A unified framework for large- and small-displacement optical flow estimation" Technical report, *The Chinese University of Hong Kong*, 2010.

[20] S. Keller, F. Lauze, and M. Nielsen, "Temporal super resolution using variational methods." *Hugh-Quality Visiual Experience: Creation, Processing and Interactivity of High-Resolution and High-Dimensional Video Signals*, eds. M. Mrak, M. Grgic, and M. Kunt, 2010.

[21] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," *VCEG Contribution VCEG-M33*, April 2001.

Noise Residual Learning for Noise Modeling in Distributed Video Coding

Huynh Van Luong and Søren Forchhammer
DTU Fotonik, Technical University of Denmark, 2800 Lyngby, Denmark
E-mail: {hulu,sofo}@fotonik.dtu.dk

Abstract—**Distributed video coding (DVC) is a coding paradigm which exploits the source statistics at the decoder side to reduce the complexity at the encoder. The noise model is one of the inherently difficult challenges in DVC. This paper considers Transform Domain Wyner-Ziv (TDWZ) coding and proposes noise residual learning techniques that take residues from previously decoded frames into account to estimate the decoding residue more precisely. Moreover, the techniques calculate a number of candidate noise residual distributions within a frame to adaptively optimize the soft side information during decoding. A residual refinement step is also introduced to take advantage of correlation of DCT coefficients. Experimental results show that the proposed techniques robustly improve the coding efficiency of TDWZ DVC and for GOP=2 bit-rate savings up to 35% on WZ frames are achieved compared with DISCOVER.**

Keywords- Distributed Video Coding; noise residual learning; adaptive noise model

I. INTRODUCTION

Distributed video coding is an interesting instance of distributed source coding where the video redundancy is partly or fully exploited at the decoder side. In recent years, conventional video coding has been challenged by some emerging applications such as video surveillance, video sensor networks etc. that require a relatively low cost encoder with high coding efficiency. DVC [1][2] has been proposed as a solution. DVC is based on two information theory theorems, the Slepian-Wolf theorem [3] and the Wyner-Ziv theorem [4], where source data are independently lossy coded but jointly decoded based on a correlated source at the decoder.

Transform Domain Wyner-Ziv (TDWZ) [1] video coding is one popular DVC scheme where the noise estimation is one of the most important aspects influencing the coding performance. The decoder needs to estimate the correlation between the corresponding source and the side information which can be obtained through frame interpolation at the decoder side. The accuracy of the correlation has a significant impact on the compression performance of DVC. Several correlation noise models [5]-[7] have been proposed, where the Laplacian distribution is commonly used for the DCT coefficients. The noise model uses different granularity levels, e.g. frame level, band level, and coefficient level. More recently, an adaptive noise model using clustering of DCT blocks has been presented [8]. The technique not only utilizes the correlation over all frequency bands but takes the decoded bands into account to influence the decoding of subsequent bands. Our goal is to improve coding efficiency by improving the adaptive noise modeling by introducing better learning of correlations using both spatial and temporal correlation.

Estimating the correlation noise has been enhanced by correlation of coefficients in each residual frame [5]-[6] or noise residual refinement [7] in the transform domain. The noise residue refinement updates the estimated noise residue for noise modeling and side information quality during decoding. In order to improve the noise estimation, this paper proposes a refinement technique that utilizes the correlation of neighbor coefficients to refine the coefficient considered, and thereafter, updates the noise parameters. To utilize the temporal redundancy, the paper uses residuals of already decoded frames to influence the noise distribution of the current frame. Finally, adaptive optimization of the number of clusters in the noise model is addressed to adaptively get the best soft side information during decoding. The rest of this paper is organized as follows. In Section II, the architecture of a TDWZ video codec is presented. The new learning techniques proposed are described in Section III. Section IV analyzes and compares the performance of our approach to other existing methods.

II. TRANSFORM DOMAIN WYNER-ZIV VIDEO CODING

The architecture of a state-of-the-art TDWZ video codec [7] is depicted in Fig. 1. In this system, the frame sequence is split into key frames and so-called Wyner-Ziv frames. Key frames are intra coded using conventional video coding techniques such as H.264/AVC intra coding. The Wyner-Ziv frames are transformed (4x4 DCT), quantized and decomposed into bitplanes. Each bitplane is fed to a rate-compatible LDPC Accumulate (LDPCA) encoder [9] from most significant bitplane (MSB) to least significant bitplane (LSB). The corresponding error correcting information is stored in a buffer. The amount of information to be transmitted depends on the requests made by the decoder through a feedback channel.

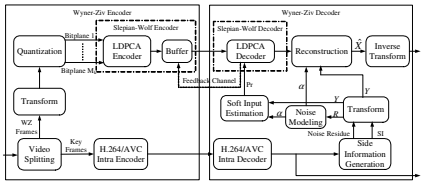


Figure 1. Architecture of feedback channel based Transform Domain Wyner-Ziv video codec

The Wyner-Ziv frame is predicted at the decoder side by using already decoded frames as references. The predicted frame, called the Side Information (SI) frame, is an estimation of the original Wyner-Ziv frame only available at the encoder. Given the available SI, soft-input information (conditional probabilities *Pr* at bit-level) within each bitplane is estimated using a noise model. Thereafter the LDPCA decoder starts to decode the various bitplanes, ordered from MSB to LSB, to

correct the bit errors. After all the bitplanes are successfully decoded, the Wyner-Ziv frame can be decoded through combined de-quantization and reconstruction followed by an inverse transform.

III. Noise Residual Learning for Adaptive Noise Model

We consider the difference between the original Wyner-Ziv frame X and the side information frame Y. The residual difference between the transformed coefficients of the WZ frame and the interpolated frame will be modeled by a Laplacian distribution with probability density function $f(r) = (\alpha/2)\exp(-\alpha \mid r-\mu \mid)$.

We consider the rate distortion bounds that depend on the variance of a source. It is stated in [10] that for a quadratic distortion D and memoryless source with variance σ^2 and entropy power Q, the upper and lower rate distortion $\Re(D)$ bounds are:

$$\frac{1}{2}\log\frac{Q}{D} \le \Re(D) \le \frac{1}{2}\log\frac{\sigma^2}{D} \tag{1}$$

where the entropy power is $Q = (1/2\pi e)\exp(2H)$ and H denotes the entropy of the source. For the Laplacian distribution, H may be calculated specifying a lower bound and thus:

$$\frac{1}{2}\log\frac{e}{\pi}\frac{\sigma^2}{D} \le \Re(D) \le \frac{1}{2}\log\frac{\sigma^2}{D} \tag{2}$$

The bounds could be reduced if we split the outputs of a given source into a number of clusters having different variance. This may e.g. be shown based on the convexity of the log-function and Jensen's inequality. We will consider using clustering for DVC noise modeling below.

A. Adaptive Noise Model using Clustering of DCT Blocks

The decoder must estimate the statistics of the residual without access to the original frame X. Consistent with the remarks above, it was noted in [8] that the variance of the residual frame based on an estimated residual is higher than the expected variance of the sub-sets. This motivates reducing the codelength by clustering into several clusters. Therefore, the techniques proposed in this paper are based on the adaptive noise model using clustering of DCT blocks [8]. Basically, the adaptive noise model method considers the (4x4 DCT) transformed residual of frequency bands in a block as components of a vector.

Let R be the residual frame in the transform domain. It is initialized at the decoder based on the difference between matching blocks of the reference images. R_k is used to indicate block k out of N 4x4 blocks in R, $1 \le k \le N$. Each block R_k, considered as a feature vector, contains 16 frequencies given by the transformed residual coefficients. Consider block k of band l and let R_k^l ($1 \leqslant l \leqslant 16$) and $\hat{R}_k^l$ denote the initial coefficient of the residual and a refined coefficient based on the partially decoded information, respectively. The feature vector of each block $R_k = (\hat{R}_k^1, \hat{R}_k^2, \ldots, \hat{R}_k^{l-1}, R_k^l, R_k^{l+1}, \ldots, R_k^{16})$ is classified into one of M categories, which in turn provides an estimate of the noise parameter. Furthermore, the proposed cluster level noise model is adaptively combined with a band level noise model. The clustering technique in [8] was updated at coefficient level and is here extended by updating at bitplane level. The noise residue refinement is exploited by an adaptive noise model [8] which is applied on a bitplane level noise residue refinement and adopted and integrated in the DVC scheme in [7]. The refinement is carried out once a bitplane is successfully decoded. Using this as the noise model in Fig. 1 is referred to as Model A.

B. Noise Model using Neighbors in Residual Refinement

To take advantage of the correlation of the residual of DCT coefficients between neighbors within each band, refinement of residuals is proposed. This residual refinement technique uses neighbor residual coefficients along with the estimated noise parameters to refine the residual of the coefficient considered, and thereafter, updates the noise parameters. Let R_{kj}^l denote the coefficients of feature vectors and α_j^l denote the Laplacian noise distribution parameter of a cluster j within band l. Assume that at the time band l needs to be decoded, α_j^l were obtained by clustering DCT blocks and estimating the noise parameter for each cluster j as in the online noise model using clustering of DCT blocks [8].

Specifically, the extension (Model B) refines R_{kj}^l based on α_j^l and the 8-neighbor residual coefficients, indexed by s and denoted R_{kjs}^l. Using the current coefficient R_{kj0}^l and the 8-neighbors, R_{kjs}^l with $1 \le s \le 8$, a refined $R^{*l}{}_{kj}$ is obtained by:

$$R^{*l}{}_{kj} = \sum_{s=0}^{8}\left(\frac{\exp(-\alpha_j^l \mid R_{kj}^l - R_{kjs}^l \mid)}{\sum_{s=0}^{8}\exp(-\alpha_j^l \mid R_{kj}^l - R_{kjs}^l \mid)}\right)R_{kjs}^l \tag{3}$$

These refined residuals are used in the set of N refined feature vectors, $R_k^* = (\hat{R}_k^1, \hat{R}_k^2, \ldots, \hat{R}_k^{l-1}, R^{*l}{}_k, R_k^{l+1}, \ldots, R_k^{16})$ used for decoding band l. The set is classified again into M clusters by using Fuzzy-C means clustering [8]. Consequently, refined noise parameters $\alpha^{*l}{}_j$ are obtained based on the observations within the current band for each refined cluster j. The bands are decoded starting from DC and proceeding with the AC coefficients in zig-zag order. Whenever a band l is successfully decoded, the coefficients of the band are reconstructed. This means that the set of feature vectors is now updated as $R_k = (\hat{R}_k^1, \hat{R}_k^2, \ldots, \hat{R}_k^{l-1}, \hat{R}_k^l, R_k^{l+1}, \ldots, R_k^{16})$ before decoding band $l+1$. The process is continued until all bands are successfully decoded.

C. Noise Residual Learning using Previously Decoded Residual Frames

This subsection extends the residual learning technique by using the previously decoded residual frames to influence the noise distribution of the current frame. The previously WZ decoded frames within a window are used to create decoded residual frames corresponding to the WZ decoded frames. The motivation is that the noise distributions based on previously decoded frames are available at the decoder and may be similar to the noise distribution of the current frame. To take advantage of both the previous decoded noise distributions and the estimated current noise distribution, the residuals based on previously decoded frames are used together with the current residual frame to form a set of data. Then the set is classified

into clusters to estimate noise parameters for each cluster of the residual frame considered.

Let W be the window size specifying the number of previously decoded WZ frames for the learning process. We consider coding even frames using WZ coding, i.e. GOP size 2. Let $\hat{R}_{(2u-2W)},...,\hat{R}_{(2u-2)}$ denote residual based on previously decoded frames and $R_{(2u)}$ denote the current residual coefficient frame at $2u$. Let $\hat{R}_{(2u-2W)k},...,\hat{R}_{(2u-2)k},R_{(2u)k}$ denote the block k, $1 \le k \le N$, of N 4x4 blocks of $\hat{R}_{(2u-2W)},...,\hat{R}_{(2u-2)},R_{(2u)}$. For each of the residuals based on previously decoded frames, consider a set of N feature vectors $\hat{R}_{(2u-2\omega)k}$ with $1 \le \omega \le W$, where $\hat{R}_{(2u-2\omega)k} = (\hat{R}^1_{(2u-2\omega)k},\hat{R}^2_{(2u-2\omega)k},...,\hat{R}^{16}_{(2u-2\omega)k})$ holds the residuals of decoded bands. For the current residual frame $R_{(2u)}$, $R_{(2u)k} = (\hat{R}^1_{(2u)k},\hat{R}^2_{(2u)k},...,\hat{R}^{l-1}_{(2u)k},R^l_{(2u)k},R^{l+1}_{(2u)k},...,R^{16}_{(2u)k})$ is the updated residual based on successfully decoded bands (up to band *l-1*) before decoding band *l*.

Consider W sets, S_ω, of feature vectors where each set is created by combining N feature vectors $\hat{R}_{(2u-2\omega)k}$ of a previous frame with N feature vectors $R_{(2u)k}$ of the current frame,

$$S_\omega = \{R_{(2u)k},\hat{R}_{(2u-2\omega)k}\} \quad (4)$$

Each set S_ω is classified into M clusters by using Fuzzy C-means clustering [8]. Thereafter noise parameters $\alpha^j_{\omega j}$ are obtained based on the observations for each cluster j of band l of set S_ω. As a result, there are W sets of noise parameters for decoding band l for each cluster j, $\{\alpha^j_{\omega j}\}$, $1 \le \omega \le W$. Let α_2 denote the parameter determined by the noise model in Section III.B. The adaptive noise model, denoted by Adaptive (C) and shown in Fig. 2, adaptively estimates the noise distribution by creating W different noise parameters $\alpha_{1\omega}$ as well as α_2 as input to the Soft Input Estimation block. The LDPCA module tries to decode using Pr_2 based on α_2 as well as each side information $Pr_{1\omega}$ based on $\alpha_{1\omega}$. The LDPCA then selects the soft side information that converges first during decoding for each bitplane. The chosen soft side information for one specific bitplane is also used for the minimum mean squared error reconstruction process [11].

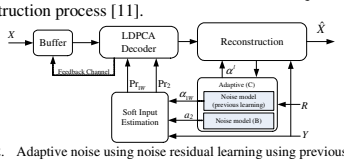


Figure 2. Adaptive noise using noise residual learning using previous frames

D. Adapting the Number of Clusters for Noise Modeling

This part will extend the noise residual estimation by selecting the number of clusters, $m \le M$, giving the best decoding, i.e. optimizing the model order. The statistical characteristics of the noise distribution may change from region to region, and over time when decoding. One reason being, that the noise distribution may not be estimated properly in regions containing moving objects. It may improve the noise modeling, if the noise residual R is adaptively modeled using a variable number of noise distributions. A dynamic mechanism is carried out to determine the optimal number of distributions within each frame once a bitplane is successfully decoded.

For each cluster j, m Laplacian distributions, D_m for $1 \le m \le M$, are estimated. As in (4), the noise parameters $\alpha^j_{\omega j}$ which are obtained based on the observation for each cluster j, $1 \le j \le m$, of band l of set S_ω are estimated for each D_m.

$$D_m = \{\alpha^j_{\omega mj}\}, 1 \le \omega \le W, 1 \le j \le m \quad (5)$$

where $\alpha^j_{\omega mj}$ is a noise parameter estimated for band l of set S_ω of distribution set D_m.

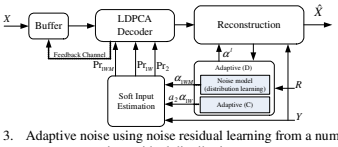


Figure 3. Adaptive noise using noise residual learning from a number of noise residual distributions

The noise parameters $\alpha^j_{\omega mj}$ of $W \times M$ candidates are considered for decoding band l for each cluster j. Let $\alpha_2, \alpha_{1\omega}$ denote noise parameters that are created by the noise model in Section III.C. The resulting noise model, called Adaptive (D) and shown in Fig. 3, adaptively estimates the noise distribution by creating $W \times M$ candidate noise parameters $\alpha_{1\omega m}$ and $\alpha_2, \alpha_{1\omega}$ as input to the Soft Input Estimation block. The LDPCA module tries to decode using each set of soft side information $Pr_{1\omega m}$ and $Pr_{1\omega}$, Pr_2. The LDPCA then selects the soft side information that converges first during decoding for each bitplane. This way, the decoder adaptively optimizes the number of bits required for decoding. In addition, the chosen noise parameter for each bitplane is also used for the minimum mean squared error reconstruction process [11]. It can be noted that the learning technique is carried out each time one bitplane is successfully decoded.

IV. PERFORMANCE EVALUATION

In this section, the Rate Distortion (RD) performance of the three proposed noise models are evaluated and compared to the DVC scheme in [7] and the noise model in Section III.A and adopted in the scheme in [7], named TDWZ(A). The three noise models proposed in Sections III.B, III.C, III.D are integrated in DVC scheme [7] in Fig. 1 and named TDWZ(B), TDWZ(C) (Fig. 2), TDWZ(D) (Fig. 3), respectively. The test sequences are 149 frames of *Foreman*, *Hall Monitor*, *Soccer*, and *Coast-guard* with 15Hz frame rate and QCIF format. GOP (group of pictures) size is 2, where odd frames are coded as key frames using H.246/AVC Intra and even frames are coded using Wyner-Ziv coding. Eight RD points are considered corresponding to eight 4x4 quantization matrices [5]. In this paper, the proposed TDWZ(C) learning from the previous decoded residual frames uses the window size *W=6*. The proposed TDWZ(D) adapting the number of noise residual distributions uses a maximum number of clusters *M=10*.

Tables I and II show the relative average bitrate savings and equivalently the average PSNR improvements (using the Bjøntegaard metric [12] and fitting a curve through the 8 RD

points measured) over the DISCOVER codec for WZ frames and overall frames, respectively. The improvements are reported for the TDWZ(A), TDWZ(B), TDWZ(C), TDWZ(D), respectively. Compared to DISCOVER, the average bitrate saving for the proposed noise model is up to 35% and 19% (or equivalently the average improvement in PSNR is up to 1.67 dB and 0.95 dB) for WZ frames and overall frames, respectively, for the difficult *Soccer* sequence. Compared to the DVC scheme in [7] denoted as TDWZ[7], the most improvements are observed for *Hall* and *Coast-guard*. Improvement of 13% is observed for *Hall* on the WZ frames. In general, the RD performances of the proposed noise models in Sec. III.A-D are robustly better than using the noise model in [7].

The overall RD performance of TDWZ(D) with the proposed noise model is illustrated in Fig. 4. The TDWZ(D) gives a better RD performance than H.264/AVC Intra coding for *Foreman*, *Hall Monitor*, and *Coast-guard*, and even better than H.264/AVC No motion for *Coast-guard*. The RD performance of TDWZ(D) clearly outperforms those of [7] and DISCOVER.

V. Conclusion

This paper proposes an adaptive noise model for Wyner-Ziv video coding using residual learning techniques. The technique utilizes residues of previously decoded frames and generates a number of noise residual distributions within a frame for adaptive optimization of the soft side information during decoding. Moreover, the technique refines the residue to take advantage of correlation of DCT coefficients and neighboring blocks. Experimental results show that the coding efficiency of the proposed noise model can significantly improve the RD performance of TDWZ compared to the TDWZ noise model [7]. The average bitrate savings of TDWZ using the adaptive noise model are up to 35 % (or equivalent the average improvement in PSNR is up to 1.67 dB) over the DISCOVER codec.

References

[1] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," Proc. of the IEEE, vol. 93(1), pp. 71–83, 2005.

[2] R. Puri, A. Majumdar, and K. Ramchandran, "PRISM: a video coding paradigm with motion estimation at the decoder," IEEE Trans. On Image Proc., vol. 16 (10), pp. 1-13, Oct. 2007.

[3] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," IEEE Trans. on Inform. Theory, vol. 19 (4), pp. 471–480, Jul. 1973.

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

[5] DISCOVER Project, www.discoverdvc.org, Dec. 2007.

[6] C. Brites and F. Pereira, "Correlation noise modeling for efficient pixel and transform domain Wyner-Ziv video coding," IEEE Trans. Circuits Syst. Video Technol., vol. 18, pp. 1177−1190, Sep. 2008.

[7] X. Huang and S. Forchhammer, "Cross-band noise model refinement for transform domain wyner-ziv video coding," Signal Processing: Image Communication, vol. 27(1), pp. 16-30, 2011.

[8] H.V. Luong, X. Huang, and S. Forchhammer, "Adaptive Noise Model for Transform Domain Wyner-Ziv Video using Clustering of DCT Blocks," IEEE Int'l Workshop Multimedia Signal Proc., Hangzhou, China, Oct. 2011.

[9] D. Varodayan, A. Aaron, and B. Girod, "Rate-adaptive codes for distributed source coding," EURASIP Signal Processing, vol. 23 (11), pp. 3123-3130, 2006.

[10] P.L. Dragotti and M. Gastpar, "Distributed Source Coding: Theory, Algorithms and Applications," Academic Press, Elsevier, 2009.

[11] D. Kubasov, J. Nayak, and C. Guillemot, "Optimal reconstruction in Wyner-Ziv video coding with multiple side information," in Proc. IEEE Int'l Workshop Multimedia Signal Process., pp. 183–186, Chania, Greece, Oct. 2007.

[12] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," VCEG Contribution VCEG-M33, April 2001.

TABLE I. Bjøntegaard Relative Bit-Rate Saving (%) and PSNR Difference (dB) Comparisons over DISCOVER for WZ Frames

Sequence	Bit-Rate Saving (%)					PSNR Difference (dB)				
	TDWZ[7]	*TDWZ(A)*	*TDWZ(B)*	*TDWZ(C)*	*TDWZ(D)*	*TDWZ[7]*	*TDWZ(A)*	*TDWZ(B)*	*TDWZ(C)*	*TDWZ(D)*
Foreman	14.029	16.498	17.670	18.888	**21.643**	0.633	0.745	0.798	0.854	**0.974**
Hall	8.305	12.276	14.256	15.274	**21.022**	0.370	0.550	0.633	0.673	**0.903**
Soccer	26.006	29.720	30.846	31.907	**34.516**	1.305	1.472	1.521	1.572	**1.677**
Coast	11.635	16.277	17.471	18.216	**21.131**	0.352	0.495	0.530	0.552	**0.637**
Average	14.994	18.693	20.061	21.071	**24.578**	0.665	0.816	0.870	0.913	**1.047**

TABLE II. Bjøntegaard Relative Bit-Rate Saving (%) and PSNR Difference (dB) Comparisons over DISCOVER for all Frames

Sequence	Bit-Rate Saving (%)					PSNR Difference (dB)				
	TDWZ[7]	*TDWZ(A)*	*TDWZ(B)*	*TDWZ(C)*	*TDWZ(D)*	*TDWZ[7]*	*TDWZ(A)*	*TDWZ(B)*	*TDWZ(C)*	*TDWZ(D)*
Foreman	6.011	7.004	7.466	7.946	**8.967**	0.335	0.391	0.417	0.444	**0.502**
Hall	2.556	3.571	3.947	4.164	**5.393**	0.187	0.262	0.290	0.306	**0.396**
Soccer	14.419	16.549	17.176	17.775	**19.370**	0.723	0.823	0.852	0.881	**0.950**
Coast	3.937	5.280	5.560	5.748	**6.420**	0.186	0.251	0.265	0.274	**0.306**
Average	6.731	8.101	8.537	8.908	**10.037**	0.358	0.432	0.456	0.476	**0.538**

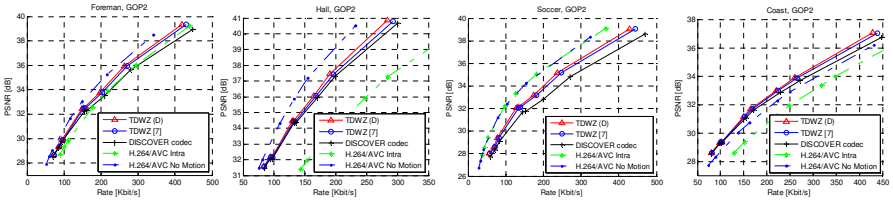


Figure 4. Overall RD performance

Exploiting the Error-Correcting Capabilities of Low Density Parity Check Codes in Distributed Video Coding using Optical Flow

Lars Lau Rakêt[a], Jacob Søgaard[b], Matteo Salmistraro[b], Huynh van Luong[b], Søren Forchhammer[b]

[a]Department of Computer Science, University of Copenhagen, Universitetsparken 1, 2100 Copenhagen, Denmark;
[b]Department of Photonics Engineering, Technical University of Denmark, Ørsteds Plads, 2800 Kgs. Lyngby, Denmark

ABSTRACT

We consider Distributed Video Coding (DVC) in presence of communication errors. First, we present DVC side information generation based on a new method of optical flow driven frame interpolation, where a highly optimized TV-L^1 algorithm is used for the flow calculations and combine three flows. Thereafter methods for exploiting the error-correcting capabilities of the LDPCA code in DVC are investigated. The proposed frame interpolation includes a symmetric flow constraint to the standard forward-backward frame interpolation scheme, which improves quality and handling of large motion. The three flows are combined in one solution. The proposed frame interpolation method consistently outperforms an overlapped block motion compensation scheme and a previous TV-L^1 optical flow frame interpolation method with an average PSNR improvement of 1.3 dB and 2.3 dB respectively. For a GOP size of 2, an average bitrate saving of more than 40% is achieved compared to DISCOVER on Wyner-Ziv frames. In addition we also exploit and investigate the internal error-correcting capabilities of the LDPCA code in order to make it more robust to errors. We investigate how to achieve this goal by only modifying the decoding. One of approaches is to use bit flipping; alternatively one can modify the parity check matrix of the LDPCA. Different schemes known from LDPC codes are considered and evaluated in the LDPCA setting. Results show that the performance depend heavily on the type of channel used and on the quality of the Side Information.

Keywords: Distributed Video Coding, LDPC, Error-Resilience, Side Information Generation, Frame Interpolation

1. INTRODUCTION

The distributed video coding paradigm contrasts ordinary hybrid video coding, by fully or partly exploiting the temporal redundancy of video data at the decoder side. This also means that one has to rethink the components one would normally use. In particular one does not have to worry about coding motion vectors, which makes it possible to consider alternative motion estimation strategies. In addition the use of alternative decoders may give rise to other opportunities. The contribution of this paper is two-fold. First we propose a novel side information generation scheme, which significantly increases the bitrate saving. Secondly we investigate methods for exploiting the error-correcting capabilities of the LDPCA[1] (low-density parity-check accumulate) code in DVC, in the case of transmission errors.

A novel DVC side information generation scheme is proposed. In this new setup three different motion estimates are used to generate a single side information frame. The motion is estimated using standard forward and backward schemes, and in addition we include a symmetric estimate, that has recently been showed to give superior quality for frame interpolation.[2] Together these three estimates are used for generating side information

Further author information:
Søren Forchhammer[b]: E-mail: sofo@fotonik.dtu.dk, Telephone: +45 45253622

Applications of Digital Image Processing XXXV, edited by Andrew G. Tescher, Proc. of SPIE
Vol. 8499, 84990N · © 2012 SPIE · CCC code: 0277-786/12/$18 · doi: 10.1117/12.929435

for Wyner-Ziv frames, and we demonstrate that results from this procedure outperforms overlapped block motion compensation and optical flow methods[3, 4], resulting in a significant bitrate saving.

Various techniques for error correction, that has been developed for fixed rate LDPC codes, has been implemented and compared, using transmission modeled by a Binary Symmetric Channel (BSC) and Gaussian channel. We have restricted ourselves to methods which do not require alterations of the encoder but only of the decoder. While previous works[5] addressed the problem using rate-adaptive Turbo codes, this is the first study on using LDPCA codes in DVC to also combat transmission errors, to the best of our knowledge.

The rest of the paper is organized as follows: In the next section we will briefly describe the DVC setup used. In Section 3 we will consider our optical flow driven side information generation. Section 4 describes the error-correcting techniques that has been implemented. Results are given in Section 5, and finally conclusions are drawn in the last section.

2. DISTRIBUTED VIDEO CODING

An efficient approach to DVC is Transform Domain Wyner-Ziv (TDWZ) video coding with a feedback channel, which was first proposed by Girod et al.[6] The decoder controls the rate by requests over a feedback channel. The DISCOVER codec[7] improved the performance of the initial TDWZ architecture and it constitutes a well known benchmark . More recently various improvements have been reported. TDWZ video coding with a cross-band noise model was proposed[3] to further improve the coding efficiency by utilizing the cross-band correlation, without changing the encoder.

The architecture of a TDWZ video codec[7] is depicted in Fig. 1. In this system, the sequence of frames is split into key frames and so-called Wyner-Ziv frames. Key frames are intra coded using conventional video coding techniques such as H.264/AVC intra coding. The Wyner-Ziv frames are transformed (4×4 DCT), quantized and decomposed into bitplanes. Each bitplane is fed to a rate-compatible LDPC Accumulate (LDPCA) encoder[1] from most significant bitplane to least significant bitplane. The corresponding error correcting information is stored in a buffer and requested by the decoder through a feedback channel.

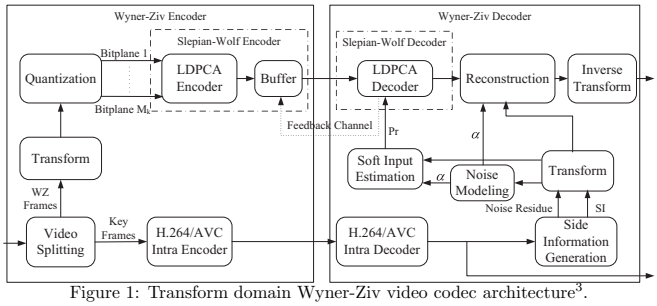


Figure 1: Transform domain Wyner-Ziv video codec architecture[3].

The Wyner-Ziv frame is predicted at the decoder side by using already decoded frames as references. The predicted frame, called the Side Information (SI) frame, is an estimate of the original Wyner-Ziv frame. Given the available SI, soft-input information (conditional probabilities for each bit) within each bitplane is estimated using a noise model. Thereafter the LDPCA decoder starts to decode the bitplanes selected by the quantizer, ordered from most to least significant bitplane, to correct the bit errors. The decoder requests bits from the buffer until the bitplane is decoded. Thereafter CRC bits are sent for confirmation. After all the bitplanes are successfully decoded, the Wyner-Ziv frame can be decoded through combined de-quantization and reconstruction followed by an inverse transform.

In DVC (Fig. 1) there are three different channels, namely the transmission channel, the virtual channel and the feedback channel. Through the transmission channel the parity bits are sent from the encoder to the decoder. The feedback channel is used by the decoder in order to request more bits to the encoder. Finally the virtual channel is used to model and calculate the relation between side information and the actual encoded frame. While the two previous channels are real communication channels, the latter is only a theoretical construction.

3. OPTICAL FLOW DRIVEN SIDE INFORMATION GENERATION

The problem of frame interpolation find uses in a number of fields, e.g. video post processing, restoration of historic material, and, the application we will consider here, video coding. For the two former applications, the goal is often to satisfy a viewer, in which case the main concern often is that the results look good,[8] rather than having good performance in terms of a specific error measure. In distributed video coding, however, it is used to generate side information for decoding and performance in terms of specific error measures are more important than crisp results. In ordinary video coding applications discrete methods like block matching has been used very successfully, and variational motion estimation methods have not gained much ground. One reason for this is that optical flow fields are dense, and thus problematic to code. In distributed video coding, however the source statistics are exploited at the decoder side, eliminating the problem of coding the flow field motion vectors. Such a setup makes it possible to exploit the highly accurate motion estimates of modern optical flow methods[4,26]. We shall extend own previous work on optical flow in DVC, by including a symmetric flow.

3.1 TV-L^1 Optical Flow

Optical flow estimation concerns the determination of apparent (projected) motion. Given a sequence of temporally indexed images I_t, we want to estimate the optical flow $\boldsymbol{v}$ such that the motion matches the image sequence while still maintaining sufficient regularity. Here we will consider a Total Variation (TV)-L^1 energy for the optical flow estimation, which is given by

$$E(\boldsymbol{v}) = \int \|I_{t+1}(\boldsymbol{x} + \boldsymbol{v}(\boldsymbol{x})) - I_t(\boldsymbol{x})\| \,\mathrm{d}\boldsymbol{x} + \int \|D\boldsymbol{v}(\boldsymbol{x})\| \,\mathrm{d}\boldsymbol{x}, \tag{1}$$

where the first term is a L^1 norm of the difference between I_t and the motion-compensated version of I_{t+1}, and the second term is a total variation regularization, which is to be understood as the integral over the Frobenius norm of the derivative of $\boldsymbol{v}$.[9] The total variation regularization will smooth the estimated motion while still allowing for sharp motion boundaries. In order to efficiently minimize E we introduce two relaxations. First we linearize the data fidelity term $I_{t+1}(\boldsymbol{x} + \boldsymbol{v}) - I_t(\boldsymbol{x}) \approx \rho(\boldsymbol{v})(\boldsymbol{x})$, where ρ is the first order Taylor approximation

$$\rho(\boldsymbol{v})(\boldsymbol{x}) = I_{t+1}(\boldsymbol{x} + \boldsymbol{v}_0) - I_t(\boldsymbol{x}) + (\boldsymbol{v}(\boldsymbol{x}) - \boldsymbol{v}_0)^\top \nabla I_1(\boldsymbol{x} + \boldsymbol{v}_0) \tag{2}$$

with $\boldsymbol{v}_0$ being the current estimate of $\boldsymbol{v}$ around $\boldsymbol{x}$. We further relax E by introducing an auxiliary variable $\boldsymbol{u}$ that splits data fidelity and regularization in two quadratically coupled energies:

$$E_1(\boldsymbol{v}) = \int \lambda \|\rho(\boldsymbol{v})(\boldsymbol{x})\| + \frac{1}{2\theta}\|\boldsymbol{v}(\boldsymbol{x}) - \boldsymbol{u}(\boldsymbol{x})\|^2 \,\mathrm{d}\boldsymbol{x}, \tag{3}$$

$$E_2(\boldsymbol{u}) = \int \frac{1}{2\theta}\|\boldsymbol{v}(\boldsymbol{x}) - \boldsymbol{u}(\boldsymbol{x})\|^2 + \|D\boldsymbol{u}(\boldsymbol{x})\| \,\mathrm{d}\boldsymbol{x}, \tag{4}$$

The above type of relaxation was first proposed by Zach et al.[10], and has since been used in a large number of optical flow algorithms.[11,12] Its most important advantage is that the two problems can easily be solved pointwise which makes the solution very easy to implement on massively parallel processors like graphics processing units (GPUs). The minimizing solutions (3) and (4) will not be replicated here, but we note that the minimizer of (3) can be found by the method of Zach et al.[10] in the case of grayscale images and in the general case of vector valued images the minimizer is explicitly presented in the work of Rakêt et al.[12]. The regularization energy (4) is minimized by the projection method of Chambolle[9,13]. We have also applied this to DVC[4,26], but here we select parameters differently.

In order to improve interpolation quality we use a specialized coarse-to-fine pyramidal implementation of the above algorithm (for more details on standard implementations we refer to the works of Rakêt et al.[2,12]). We have 70 pyramid levels with a scaling factor of 0.95, where each pyramid level is smoothed with a Gaussian with standard deviation $\frac{\sqrt{2}}{4}$ before downscaling to the coarser level. On each level we do 30 warps of first solving (3) and then solving (4) using 10 iterations of the algorithm of Bresson[9], with $\lambda = 3$ and $\theta = 0.2$, where in order to improve interpolation quality, ρ has been weighted by the gradient magnitude $\|\nabla I_1(\boldsymbol{x} + \boldsymbol{v}_0) + 0.01\|$ (slightly shifted to avoid division by 0) in the minimization of (3)[14]. Additional improvement of interpolation quality was found by applying a 3×3 median filter of the flow after upscaling to the next pyramid level[11].

3.2 Frame Interpolation algorithm and results

We are interested in interpolating an in-between frame $I_{1/2}$ using only the two surrounding frames I_0 and I_1. We first note that the optical flow algorithm presented in the previous section is asymmetric, since the (forward) flow estimated from I_0 to I_1 is not the same as the (backward) flow from I_1 to I_0. In addition the forward flow will have a coordinate system corresponding to the pixels in I_0 and the backward flow follows the coordinate system given by the pixels in I_1, so in order to use these flows to interpolate at pixel positions in $I_{1/2}$ we need to temporally warp the flows[15–17] to match the intermediate frame. This is done by assuming that the intermediate frame follows the estimated motion linearly, and then defining the warped forward flow as the flow from $I_{1/2}$ to I_1, which is approximated by

$$\boldsymbol{v}_f^{1/2}(\text{round}(\boldsymbol{x} + 1/2\boldsymbol{v}_f(\boldsymbol{x}))) = 1/2\boldsymbol{v}_f(\boldsymbol{x}), \tag{5}$$

where the round function rounds to nearest pixel. The warped backward flow is estimated similarly. This simple warping procedure does however contain some problems, first multiple flow vectors may hit the same pixel $\text{round}(\boldsymbol{x} + 1/2\boldsymbol{v}_f(\boldsymbol{x}))$ (typically occlusion), which can be dealt with by choosing the vector with best data fidelity. A more serious problem is the problem of dis-occlusion which causes holes in the warped flow. We will correct this by filling holes using an outside-in strategy, however ideally one would reason about depth and occlusion in the interpolation procedure, which should give slightly better results[16].

With the warped flows, the straightforward approach for interpolation is to interpolate along the flow vectors,

$$I_{1/2}(\boldsymbol{x}) = \frac{1}{2}(I_1(\boldsymbol{x} + \boldsymbol{v}_f^{1/2}(\boldsymbol{x})) + I_0(\boldsymbol{x} + \boldsymbol{v}_b^{1/2}(\boldsymbol{x}))), \tag{6}$$

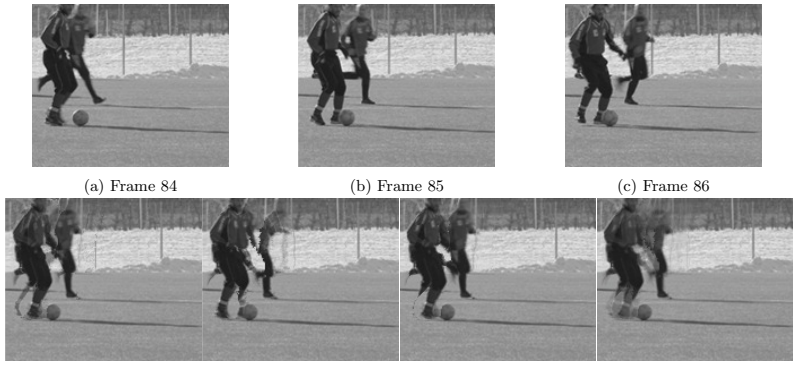
however, since we have discarded occlusion information by filling holes and clearing collisions, the warped forward flow should have been symmetrized, so it can be thought of as a minimizer of $I_1(\boldsymbol{x} + \boldsymbol{v}_f^{1/2}(\boldsymbol{x})) + I_1(\boldsymbol{x} - \boldsymbol{v}_f^{1/2}(\boldsymbol{x}))$, and vice versa for the backward flow. Even though the two computed flows are symmetric around $I_{1/2}$, they will be different since they originated from asymmetric flows. We propose to include a truly symmetric flow estimate which is calculated directly using the pixel positions of the unknown frame $I_{1/2}$, to complement the two asymmetric flows. This flow $\boldsymbol{v}_s$ is calculated using the reparametrization of (3) first suggested by Alvarez et al.[18], and recently analyzed in a frame interpolation setup by Rakêt et al.[2] i.e. replacing the data fidelity term in (3) by

$$I_1(\boldsymbol{x} + \boldsymbol{v}_s(\boldsymbol{x})) + I_1(\boldsymbol{x} - \boldsymbol{v}_s(\boldsymbol{x})) \approx I_1(\boldsymbol{x} + \boldsymbol{v}_0) + I_1(\boldsymbol{x} - \boldsymbol{v}_0) + (\boldsymbol{v}_s(\boldsymbol{x}) - \boldsymbol{v}_0)^\top (\nabla I_1(\boldsymbol{x} + \boldsymbol{v}_0) + \nabla I_0(\boldsymbol{x} - \boldsymbol{v}_0)). \tag{7}$$

We see that the linearized data fidelity term fits in the setup of Zach et al.[10], and so can be minimized by the formula giving the minimizer of (2). The result will however be different in a number of ways. The motion vectors are now only half size, which makes the method more robust against large deviations. Furthermore the sum of the two gradient terms will make the algorithm more robust to noise, and finally we do not have to do a temporal warping of the flow, in order to use it for interpolation. All in all this produces a more robust flow for interpolation, and combining the symmetric flow with the warped forward and backward flows, we propose to do the interpolation as follows

$$\begin{aligned} I_{1/2}(\boldsymbol{x}) = \frac{1}{6}(&I_1(\boldsymbol{x} + \boldsymbol{v}_f^{1/2}(\boldsymbol{x})) + I_1(\boldsymbol{x} - \boldsymbol{v}_b^{1/2}(\boldsymbol{x})) + I_1(\boldsymbol{x} + \boldsymbol{v}_s(\boldsymbol{x})) \\ &+ I_0(\boldsymbol{x} - \boldsymbol{v}_f^{1/2}(\boldsymbol{x})) + I_0(\boldsymbol{x} + \boldsymbol{v}_b^{1/2}(\boldsymbol{x})) + I_0(\boldsymbol{x} - \boldsymbol{v}_s(\boldsymbol{x}))), \end{aligned} \tag{8}$$

i.e. the interpolation is the average of the two surrounded frames warped to the center using the three different flows. Figure 2 shows the results of the three different types of interpolation, along with the estimate (8). The noise residual frames (in pixel domain) used in the DVC setup are calculated by subtracting the average of the three warped versions of I_0 from the three warped versions of I_1.



(a) Frame 84 (b) Frame 85 (c) Frame 86

(d) Forward interp. PSNR 20.40 (e) Backward interp. PSNR 20.33 (f) Symmetric interp. PSNR 21.45 (g) Average (8) PSNR 21.17

Figure 2: (a)–(c) Frames 84, 85 and 86 of the *Soccer* sequence. (d)–(f) The forward, backward and symmetric parts of (8). (e) The average interpolation (8).

We will evaluate (8) which we will denote 3OF on the test sequences (QCIF, 15 fps) *Coastguard* QP=26, *Foreman* QP=25, *Hall* QP=24 and *Soccer* QP=25, where we interpolate every other frame and compare to the overlapped block motion compensation (OBMC) method of Huang et al.[3] and the TV-L^1 optical flow (OF) method presented by Huang et al.[4]. The results can be found in Table 1 where we see that the proposed method outperforms OBMC and OF on all sequence, with an average increase in PSNR of 1.16 dB over OBMC and 2.14 dB over OF.

Sequence	OBMC[3]	OF[4]	3OF
Coastguard	31.83	30.92	32.59
Foreman	29.26	29.28	30.08
Hall	36.46	32.28	36.91
Soccer	21.30	22.43	23.90

Table 1: Average PSNR across the 74 interpolated frames for the four test sequences.

The SI generated based on the frame interpolation (8) is then used inside the TDWZ decoder together with the OBMC method, for more details please refer to Section 5.

4. ERROR CORRECTION

We now consider the problem of having noise on the transmission of the syndrome bits. We assume that the feedback channel, the transmission channel of the H.264 frames and the transmission of the Cyclic Redundancy Check (CRC) are error free. In LDPCA-based decoders, since the syndromes are error-free they are used to check the results. We relax this condition in order to allow the decoder to accept a result even if the syndrome condition is not satisfied.

4.1 Expanded Code

The most straightforward method to handle errors on the transmission channel is to consider the syndrome bits belonging to the code with parity check matrix H as the last parity bits of another larger code $[H_{m\times n}|I_{m\times m}]$ where $I_{m\times m}$ is the identity matrix. This was proposed by Tan and Li[19] among others. Thus instead of having to fulfill the syndrome conditions $HY = S$, where Y is the side information and S is the syndromes, the new code should fulfill:

$$[H_{m\times n}|I_{m\times m}] \begin{bmatrix} Y_n \\ S_m \end{bmatrix} = 0. \tag{9}$$

This means that instead of only considering Y as a noisy version of the original bitplane X, the received syndromes $\hat{S}$ are also considered as a noisy version of the original syndromes S.

It is well known[20] that there are three major features of the parity check matrix that influence the performance of the message passing algorithm for a LDPC code. The three features are:

1. The weight of each column should be big enough
2. The weight of each row should be small enough
3. The graph of the code should contain no cycles of length four

In a typical DVC setup with a regular LDPCA code the first feature is satisfied for all rates for the original parity check matrix, but when concatenated with the identity matrix a problem arises since each new column only has a weight of one. The second feature is easily satisfied for high rates, but is harder to satisfy for low rates, since the number of rows drops. The third feature is again easily satisfied for high rates but in typical LDPCA codes it is not ensured, even for high rates. For low rates it may be impossible to satisfy. The concatenation with the identity matrix does not change the second and third features. An alternative to item 3 could be that the girth of the corresponding Tanner graph should be big enough. It should be noted that even though these features are well known to influence the performance of a LDPC code we do not have theoretical grounds allowing us to predict the behavior of the modified LDPCA code.

The next two sections will present methods inspired by traditional LDPC codes assuming that the errors on the transmission channel can be considered as a Binary Symmetric Channel (BSC). In Section 4.4, the noise on the transmission channel will be assumed to be Gaussian distributed.

4.2 Bit Flip

Bit flipping methods[21] for LDPC codes are fairly good approximations to the more advanced belief propagation. More advanced variations of this method such as weighted bit flip decoding[22], reliability ratio based weighted bit flip decoding[23] and gradient descent bit flip decoding[24] have also been developed in the recent years.

The main idea behind the methods is that if there is a low enough number of parity checks which fail it might be due to transmission errors. Thus in this case all the syndromes involved in these failed parity checks could be flipped and if the decoding is successful with these new syndromes, it is assumed that the flipping was correct. If the correctness of the decoding is checked by a CRC then it should be noted that each time a sequence of syndromes are flipped the strength of the CRC is in a sense weakened since there is a new risk of decoding into a wrong code word which also satisfies the CRC. Before starting the explanation of the developed methods, it should also be noted that since there are two errors on the syndromes for each error on the accumulated syndromes (unless the errors on the accumulated syndromes are right next to each other) the expected number of errors on the syndromes are approximated by multiplying the expected number of errors on the accumulated syndromes by two.

The first method is the simplest version of this way of thought and it is called "Simple Bit Flip". Suppose we have received m bits, and P_e is the error probability on the transmission channel and let τ be a small natural number. After running the belief propagation algorithm, if the decoding is not successful, we define with PCF the number of failed parity checks, if $PCF < 2mP_e + \tau$ we flip syndromes involved in failed parity checks and

we rerun the belief propagation, after that we again check the syndrome condition and the CRC check. If both are satisfied we accept the word, otherwise we increase the rate.

The second method is inspired by the gradient descent method of Wadayama et al.[24] which outperforms traditional weighted bit flip. In this method the maximal number of expected errors on the syndrome is calculated using the binomial distribution, and after the belief propagation an error function value for each bit is calculated: $E(y_i) = \lambda y_i \hat{x}_i + \sum_{k \in C} PC_k$ where λ is a weight parameter, y_i is the bit belonging to the SI in bipolar coordinates, $\hat{x}_i$ is the corresponding decoded bit in bipolar coordinates, C is the map of connected parity checks to the current node, PC_k is the value of the parity check in bipolar coordinates. The first term in the error function corresponds to the correlation between the SI word and a codeword while the second term is the sum of the bipolar syndromes. At a given rate, after the first belief propagation, if $PCF \leq m$ where m is the highest number of expected errors with certainty η, we calculate the error term for each bit and with this the reliability of the syndromes. The syndromes having lowest reliability are flipped and the belief propagation is executed again.

4.3 Increased Column Weight (ICW)

In order to improve the aforementioned features various methods have been proposed.[20,25] We have developed an alternative approach in order to increase the weight of columns with column weight one and disregard cycles of four (since they are present in the original LDPCA code anyway). Our method is outlined in Algorithm 1. It should be noted that the algorithm is only designed for LDPC codes where all columns have a weight above one except for the concatenated identity matrix.

Algorithm 1 Increase Column Weight

Let $H_{m \times n}$ be the input parity check matrix and initialize $NM = n + m$ and an all-zero output matrix $H'_{4m \times n+2m}$.
for $i = 1$ **to** $i = m$ **do**
  Let the set O_i denote all the positions of 1's in row i.
  if any bit in row i is part of a cycle of length four **then**
    Choose a random element $o_i \in O_i$ which is part of such a cycle.
  else
    Choose a random element $o_i \in O_i$.
  end if
  Let $N = n + i$ and $K = 4(i - 1)$.
  Set the elements indicated by N and $NM + 1$ in row number $K + 1$ of the output matrix to 1.
  Set the elements indicated by o_i, $NM + 1$ and $NM + 2$ in row number $K + 2$ of the output matrix to 1.
  Set the elements indicated by o_i, N and $NM + 2$ in row number $K + 3$ of the output matrix to 1.
  Set the elements indicated by $o_i \setminus o_i$ and $NM + 2$ in row number $K + 4$ of the output matrix to 1.
  Set $N = N + 2$.
end for

4.4 Modifications in Case of Gaussian Errors in Accumulated Syndrome Bits

We assume the noise on the transmission channel to be Gaussian distributed. The error function flip method uses the error probabilities calculated from the soft values of the syndromes. One can calculate the probability of error of the syndromes $P(S_i)$ from the error probability on the accumulated syndromes $P(A_i)$.

The Error Function Flip method is altered to handle soft errors by changing the error function $E(y_i) = -\lambda P(1 - \hat{x}_i|y_i) - \sum_{k \in C} P(S_i)$, where $P(1 - \hat{x}_i|y_i)$ is the probability of the decoded bit to be wrong given the soft value of the received bit and P_e^k is the error probability of a connected syndrome.

The Log-Likelihood Ratio (LLR) values for the syndromes, in this work when using soft errors, are initialized by comparing the magnitude of the current LLR-value (of the accumulated syndrome) and the magnitude of the previous LLR-value (previous syndrome), and then choosing the lower of the two as the magnitude for the current LLR-value of the syndrome. In this way the uncertainty for a syndrome bit is propagated to the next bit to accommodate for the relationship between accumulated syndromes and not accumulated syndromes.

5. RESULTS

5.1 Performance Evaluation for DVC using Optical Flow

This section considers the TDWZ video codec[26] obtained by including the proposed 3OF (Section 3) in our TDWZ codec, which uses a cross-band[3] noise model with clustering[27] techniques in the noise model.

5.1.1 Transform Domain Wyner-Ziv Video using Optical Flow and Clustering

The TDWZ video depicted in Fig. 3 consists of OBMC and the proposed Optical Flow based side information generations (3OF), a noise model (Clustering) using clustering[27], and a cross-band noise model (Cross Band)[3]. The proposed optical flow (3OF) replaces the optical flow of our previous TDWZ codec[26]. The cross band noise model[3] was introduced utilizing cross band correlation based on the previously decoded neighboring bands. The decoder cross band noise model includes a classification module, a bitplane level noise residue refinement, and a modified maximum likelihood estimator to calculate noise parameter. The clustering noise model[27] was utilized to take correlation of DCT coefficients and residues from previously decoded frames into account to estimate the decoding residue more precisely. This noise model estimates the correlation noise by clustering of DCT blocks and using the correlation of neighbor coefficients to refine the Laplacian parameter. Furthermore, the noise model also generates a number of noise residual distributions based on previously decoded frames for adapting of soft side information during decoding.

The architecture of the TDWZ decoder[26] including the proposed 3OF is presented in Fig. 3. The side information generations generate the noise residual frames NR_1, NR_2 and the side information frames, SI_1, SI_2. SI_1 and NR_1 are generated by using OBMC[3] and SI_2 and NR_2 are generated by the proposed 3OF. These are transformed and input to the noise models. For each side information scheme, noise parameters α_{CB} using multiple hypotheses[4] combined with the cross-band[3] and α_{CL} are calculated using the clustering model[27]. Based on the transformed side information frames and the noise parameters, the soft-inputs Pr_{1CB}, Pr_{2CB}, and Pr_{1CL}, Pr_{2CL} are calculated, where Pr_{1CB} and Pr_{2CB} are calculated based on the cross-band noise and multi-hypothesis techniques.[4] Pr_{1CL}, Pr_{2CL} are obtained by applying the clustering model to each side information generation scheme, here OBMC and the proposed 3OF. All soft-inputs are fed into the multiple input LDPCA decoder and the soft-input which converges first is selected for LDPCA decoding. The corresponding selected noise parameter is chosen for reconstruction.

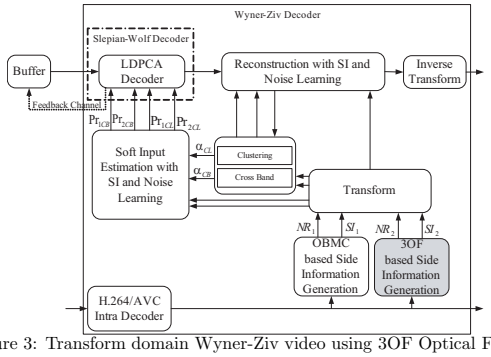


Figure 3: Transform domain Wyner-Ziv video using 3OF Optical Flow.

5.1.2 Performance Evaluation

The rate-distortion (RD) performance of the proposed techniques are evaluated for the test sequences (149 frames of) *Foreman, Hall Monitor, Soccer,* and *Coastguard* with 15Hz frame rate and QCIF format. The GOP size is 2, where odd frames are coded as key frames using H.264/AVC Intra and even frames are coded using Wyner-Ziv

coding. Eight RD points are considered corresponding to eight 4×4 quantization matrices[7]. The parameters for H.264/AVC Intra are set as by DISCOVER[7] and QP values are set to those used for the key frames in the Wyner-Ziv video coding in the DISCOVER codec[7]. It can be noted that only the luminance component of each frame is evaluated.

Table 2: Bjøntegaard Relative Bit-rate Savings (%) over DISCOVER for WZ Frames

Sequence	Cross-band	Clustering	Multi-hypothesis	TDWZ (3OF)
Foreman	14.0	21.6	27.0	36.0
Hall	8.3	21.0	13.3	26.0
Soccer	26.0	34.5	41.2	63.2
Coast	11.6	21.1	17.4	35.6
Average	15.0	24.6	24.7	40.2

Table 3: Bjøntegaard PSNR Improvement (dB) over DISCOVER for WZ Frames

Sequence	Cross-band	Clustering	Multi-hypothesis	TDWZ (3OF)
Foreman	0.633	0.974	1.177	1.530
Hall	0.370	0.903	0.575	1.095
Soccer	1.305	1.677	1.921	2.782
Coast	0.352	0.637	0.526	1.031
Average	0.665	1.047	1.050	1.610

Tables 2 and 3 report RD performance of the proposed scheme in Section 5.1.1, named TDWZ(3OF). Tables 2 and 3 present the relative average bitrate savings and equivalently the average PSNR improvements (using the Bjøntegaard difference metric[28] and fitting a curve through the 8 RD points measured) over the DISCOVER codec for WZ frames. The results are also compared to the DVC scheme called Cross-band[3]. The TDWZ(3OF) codec based on combining the clustering[27] and multi-hypothesis[4] techniques, which are also individually compared (Clustering[27] and Multi-hypothesis[4]). Compared to DISCOVER, the average bitrate saving for the proposed scheme TDWZ(3OF) is overall (average Bjøntegaard) 40.2% and 16.2% better on WZ frames and all frames, respectively. The performance improvement is 63.2% and 33.6% (or equivalently the average improvement in PSNR is 2.78 dB and 1.56 dB) for WZ frames and overall frames, respectively, for the difficult *Soccer* sequence.

The RD performance of the TDWZ(3OF) codec and H.264/AVC coding is also depicted in Fig. 4 for all frames. The TDWZ(3OF) codec gives a better RD performance than H.264/AVC Intra coding for *Foreman*, *Hall Monitor*, and *Coastguard*, and also better than H.264/AVC No Motion for *Coastguard*. The RD performance of the TDWZ(3OF) codec clearly outperforms those of Cross-band[3] and DISCOVER.

5.2 Error Prone Transmission Channel

In the following sections, results for transmission channels with noise will be presented.

5.2.1 Binary Symmetric Channels

In this section it is assumed that the bit X_i forming the bitplane has equal probability of being 0 or 1 and that the transmission channel and Side Information channel are BSC's. We will refer to the error probability of the SI channel by *crossover probability*. The effect of different parameters will be investigated and the performance of the different methods will be evaluated.

The first two simulations compare the two bit flip methods and the expansion methods and show the influence of λ parameter in the Error Function Flip (EFF) method. The Bit Error Rate (BER) and the rate for different error probabilities on the transmission channel and for two different error probabilities on the SI channel can

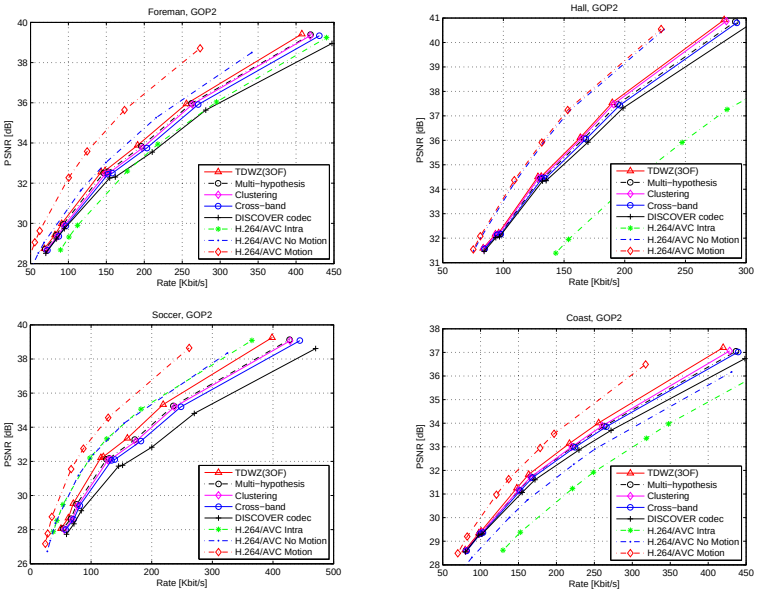


Figure 4: PSNR vs. rate for the proposed TDWZ (3OF) codec for all frames (QCIF, 15Hz, GOP2).

be seen in Fig. 5. It appears that the EFF method has better performance than the Simple Flip for high error probability on the transmission channel. It can also be seen that the λ parameter has a very low impact on the performance of the EFF method, but the best performance is for very low λ parameters which suggests that it is better to disregard the correlation between a received word and a codeword than taking the correlation into account. It is apparent that the expansion methods usually outperform the flipping methods. It also appears that for the good SI the ICW method is performing better than the expansion method[19] with regard to BER. In regards to bitrate the ICW method also outperforms the expansion method when the SI is good, except when there are no errors on the transmission channel.

5.2.2 Gaussian Transmission Channel

In this section the two expansion methods are tested and evaluated in DVC simulations. The simple expansion method is also applied to SI estimated by the 3OF method as described in Section 3. To save computation time, the simulation with 3OF SI has been conducted with SI already calculated in a DVC simulation without channel errors. Thus errors can not propagate down through the bitplanes and the PSNR cannot be calculated. We therefore assume that the PSNR is the same for these simulations as their normal SI counterparts. The simple expansion method is also benchmarked against turbo coding. The noise in the transmission channel is assumed to be Gaussian distributed. Only four different RD points corresponding to four quantization levels are used since they seem to match a concave function in rate-distortion sense.

The rate-distortion plots for the four test sequences appear in Fig. 6 with no errors (NE) on the transmission channel (the punctured lines), with a standard deviation of the Gaussian distribution to match the error probability of $P_e = 0.001$ (dotted lines) and $P_e = 0.01$. From the theoretical point of view we define P_e as the error

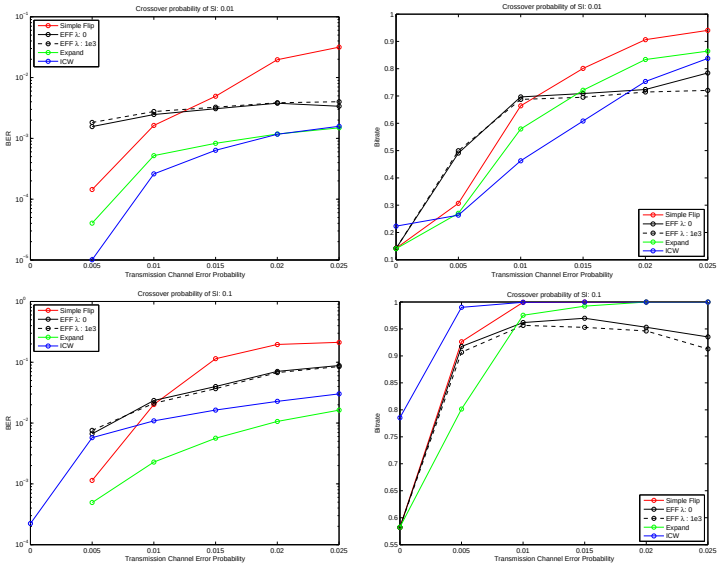


Figure 5: Results for the BSC transmission channel, modeling the SI channel also as BSC.

probability of the Gaussian channel followed by a threshold detector having the threshold at the same distance from the two symbols used. It is evident that the simple expansion method performs best overall, and in most cases the simple expansion with noise is also better than ICW with no noise. It also appears that with 3OF SI the bitrates are lower than a normal SI for the same error rates as expected.

To test the robustness of the LDPC code used versus the robustness of a Turbo code, simulations have been performed with $P_e = 10^{-2}$ (Fig. 7) and with $P_e = 10^{-3}$ (Fig. 8), for the Turbo code 25 iterations are waited before trying the CRC check for the first time. Since the initial simulations showed that the Turbo code depended heavily on the CRC both an 8-bit and a 16-bit CRC has been used in the simulations. From the rate-distortion plots it appears that for an 8-bit CRC the Turbo code has many decoding errors. If the CRC is increased to 16-bit though, the Turbo code has better performance than the LDPCA using the same CRC, which does not improve by the stronger CRC. It has to be noted however that in absence of errors the LDPCA codes outperform Turbo codes. In the presence of errors, 16-bit CRC Turbo coding is better in all the sequences except *Hall* in the case of $P_e = 10^{-2}$. In the case of $P_e = 10^{-3}$ for *Hall* and *Coast* the LDPCA codes outperforms Turbo coding, while on the other two sequences the situation is inverted. The explanation may be that the LDPCA code is built on a Rate 1/2 LDPC code while the Turbo is built on a Rate 1/3 code, thus in high bit rate cases the 1/2 rate LDPC may not provide enough redundancy to correct both errors in the SI and the transmission of syndrome bits.

It can also be noted that in some cases a drop in the PSNR is experienced while increasing the quality level, i.e. increasing the number of bits sent does not improve the PSNR. A possible explanation is that, since increasing the quality is done by increasing the number of sent LSB bitplanes, these new and high error-prone bitplanes increase the number of wrongly decoded bitplanes. Hence skipping a bitplane (i.e. not sending it and using the SI bitplane as substitute) could improve the results, achieving a lower rate and sacrificing PSNR performance in the case of a possible correct decoding. In Table 4, the results are presented for a system in which skipping was

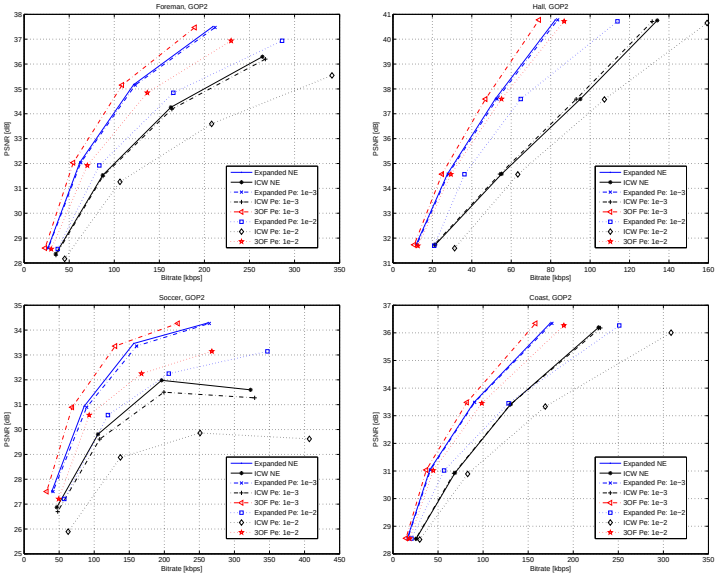


Figure 6: Rate-distortion plots with errors and no errors (NE) on the transmission channel

only allowed for the LSB and it was done only if the estimated conditional entropy is higher than a predefined threshold with $P_e = 10^{-2}$. The results are presented using the Bjøntegaard difference metric between an 8-bit CRC LDCPA-based expanded decoder and the same decoder with the skip strategy implemented. Indeed the skipping improves the performance.

Table 4: Bjøntegaard PSNR and bitrate Improvement over the non-skip decoder for WZ Frames

Sequence	PSNR Difference [dB]	Bit-rate Savings (%)
Foreman	0.61	13.12
Hall	1.09	23.64
Soccer	0.70	13.30
Coast	0.77	15.73

6. CONCLUSION AND DISCUSSION

A new method for side information generation in a DVC setup is presented. The method has been shown to consistently outperform the previously suggested methods, while at the same time being computationally efficient. The novelty of the interpolation method is a setup which includes a symmetric optical flow constraint in the interpolation, and a specialized setup in the motion estimation process, that produces estimates well suited for interpolation purposes. The addition of a symmetric term is not tied to the specific setup, nor the chosen algorithm (TV-L^1), and can easily be incorporated in most motion estimation algorithms, at low cost in terms of computation. A further gain in interpolation accuracy may be obtained from using anisotropic regularization

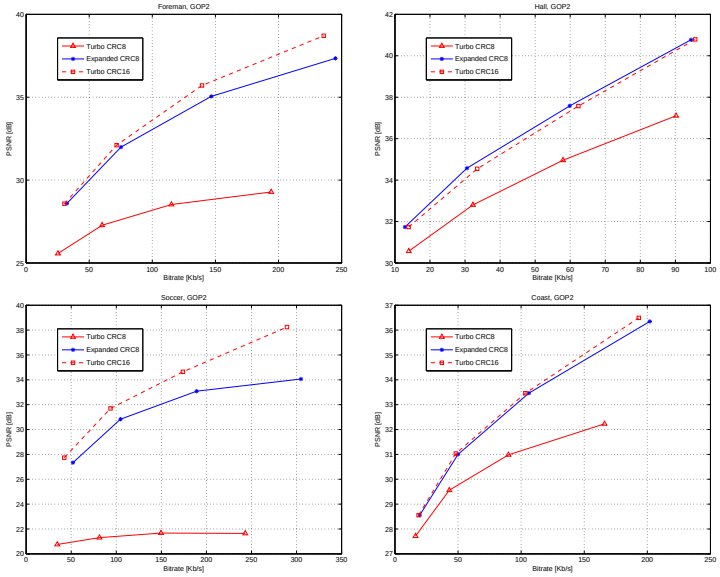


Figure 7: Rate-distortion plots of Turbo and LDPC decoding with 8-bit and 16-bit CRC, with $P_e = 10^{-2}$

instead of TV regularization. In particular, the anisotropic Huber-L^1 algorithm of Werlberger et al.[11] has proved to give good interpolation results[15]. Alternatively one may introduce anisotropy by adaptively adjusting the smoothness weight locally[29], which has recently shown to improve interpolation performance.[15]

In addition we have considered using the Slepian-Wolf decoder to handle transmission errors . Simple bit flip methods are presented to add robustness to the LDPC code in DVC. These methods are simple alternatives to methods where the decoding matrix has to be modified, but the latter shows better performance. Our simulations have shown that there is a difference in performance when assuming a BSC as the transmission channel versus a transmission channel with Gaussian distributed noise. In the BSC case our ICW method outperformed the expanded method when the SI was good, but when the noise in the transmission channel was assumed to be Gaussian distributed the expanded method was the best choice for all of the four test sequences. Our simulations also indicate that the bitrate is still improved when using the 3OF SI and the expanded method with an erroneous transmission channel. Further work with robustness for LDPC in DVC could focus on combining LDPC codes optimized for different intervals of the rate where a PEG-like approach[30] is used to make the LDPC codes rate-adaptive.

The LDPCA code was compared with Turbo coding for DVC. Without errors on parity bits/syndromes LDPCA was the best performing decoder. In the error case, Turbo coding (with a 16 bit CRC) performed best in the high-motion sequences, due to a lower maximum level of redundancy in the investigated LDPCA code. Finally, a proof-of-concept of a decoder-driven skip strategy was presented as a possible remedy to the weakness of the LDPCA code, showing promising results.

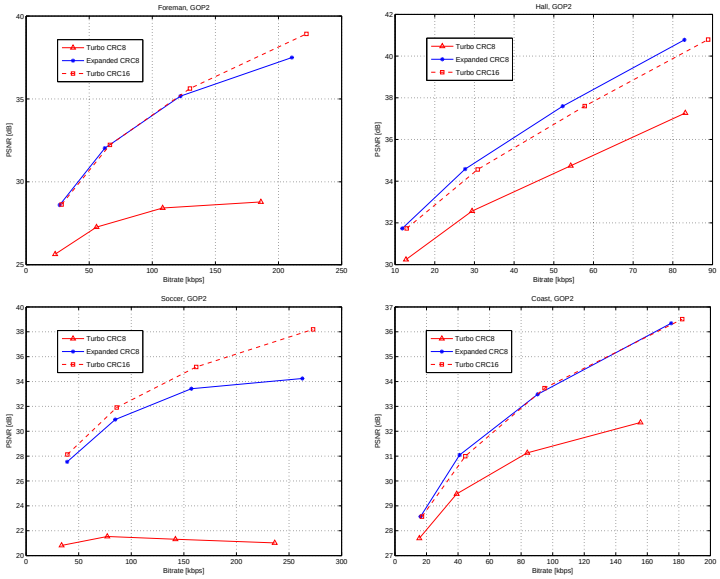


Figure 8: Rate-distortion plots of Turbo and LDPC decoding with 8-bit and 16-bit CRC, with $P_e = 10^{-3}$

REFERENCES

[1] Varodayan, D., Aaron, A., and Girod, B., "Rate-adaptive codes for distributed source coding," *Signal Process.* **86**, 3123–3130 (nov 2006).

[2] Rakêt, L., Roholm, L., Bruhn, A., and Weickert, J., "Motion compensated frame interpolation with a symmetric optical flow constraint," in [*International Symposium on Visual Computing (ISVC)*], (2012).

[3] Huang, X. and Forchhammer, S., "Cross-band noise model refinement for transform domain Wyner-Ziv video coding," *Signal Processing: Image Communication* **27**, 16–30 (2005).

[4] Huang, X., Raket, L., Luong, H. V., Nielsen, M., Lauze, F., and Forchhammer, S., "Multi-hypothesis transform domain wyner-ziv video coding including optical flow," in [*Multimedia Signal Processing (MMSP), 2011 IEEE 13th International Workshop on*], 1 –6 (oct. 2011).

[5] Yasakethu, S. L. P., Weerakkody, W. A. R. J., Fernando, W. A. C., Pereira, F., and Kondoz, A. M., "An improved decoding algorithm for dvc over multipath error prone wireless channels," *IEEE Trans. on Circuits and System for Video Tech.* **19**(10), 1543–1548 (2009).

[6] Girod, B., Aaron, A., Rane, S., and Rebollo-Monedero, D., "Distributed video coding," *Proc. of IEEE (Special issue on advances in video coding and delivery)* **93**(1), 71–83 (2005).

[7] Artigas, X., Ascenso, J., Dalai, M., Klomp, S., Kubasov, D., and Ouaret, M., "The discover codec: Architecture, techniques and evaluation," in [*Proc. Picture Coding Symposium (PCS)*], (2005).

[8] Keller, S., Lauze, F., and Nielsen, M., "Temporal super resolution using variational methods," in [*High-Quality Visual Experience: Creation, Processing and Interactivity of High-Resolution and High-Dimensional Video Signals*], Mrak, M., Grgic, M., and Kunt, M., eds., Springer (2010).

[9] Bresson, X. and Chan, T., "Fast dual minimization of the vectorial total variation norm and application to color image processing," *Inverse Problems and Imaging* **2**(4), 455–484 (2008).

[10] Zach, C., Pock, T., and Bischof, H., "A duality based approach for realtime TV-L^1 optical flow," in [*Pattern Recognition*], Hamprecht, F., Schnörr, C., and Jähne, B., eds., *Lecture Notes in Computer Science* **4713**, 214–223, Springer (2007).
[11] Werlberger, M., Trobin, W., Pock, T., Wedel, A., Cremers, D., and Bischof, H., "Anisotropic Huber-L1 optical flow," in [*British Machine Vision Conference (BMVC)*], (2009).
[12] Rakêt, L., Roholm, L., Nielsen, M., and Lauze, F., "TV-L^1 optical flow for vector valued images," in [*Energy Minimization Methods in Computer Vision and Pattern Recognition*], Boykov, Y., Kahl, F., Lempitsky, V., and Schmidt, F., eds., *Lecture Notes in Computer Science* **6819**, 329–343, Springer (2011).
[13] Chambolle, A., "An algorithm for total variation minimization and applications," *Journal of Mathematical Imaging and Vision* **20**, 89–97 (2004).
[14] Zimmer, H., Bruhn, A., and Weickert, J., "Optic flow in harmony," *International Journal of Computer Vision* **93**, 368–388 (2011).
[15] Baker, S., Scharstein, D., Lewis, J. P., Roth, S., Black, M. J., and Szeliski, R., "A database and evaluation methodology for optical flow," *International Journal of Computer Vision* **31**(1), 1–31 (2011).
[16] Herbst, E., Seitz, S., and Baker, S., "Occlusion reasoning for temporal interpolation using optical flow," Tech. Rep. UW-CSE-09-08-01, Department of Computer Science and Engineering, University of Washington (2009).
[17] Werlberger, M., Pock, T., Unger, M., and Bischof, H., "Optical flow guided TV-L^1 video interpolation and restoration," in [*Energy Minimization Methods in Computer Vision and Pattern Recognition*], Boykov, Y., Kahl, F., Lempitsky, V., and Schmidt, F., eds., *Lecture Notes in Computer Science* **6819**, 273–286, Springer (2011).
[18] Alvarez, L., Castao, C., Garca, M., Krissian, K., Mazorra, L., Salgado, A., and Sánchez, J., "Symmetric optical flow," in [*Computer Aided Systems Theory–EUROCAST 2007*], Daz, R., Pichler, F., and Arencibia, A., eds., *Lecture Notes in Computer Science* **4739**, 676–683, Springer (2007).
[19] Tan, P. and Li, J., "Enhancing the robustness of distributed compression using ideas from channel coding," in [*IEEE GLOBCOM*], (2005).
[20] Heidarzadeh, A. and Lahouti, F., "On robust syndrome-based distributed source coding over noisy channels using LDPC codes," in [*IEEE Int. Conf. on Signal Proc. and Comm.*], (2007).
[21] Gallager, R., "Low density parity check codes," tech. rep., M.I.T. Press (1963).
[22] Kou, Y., Lin, S., and Fossorier, M., "Low density parity check codes based on finite geometries: A rediscovery and new results," *IEEE Trans. on Inform. Theory* **47**(7), 2711–2736 (2001).
[23] Guo, F. and Hanzo, L., "Reliability ratio based weighted bit-flipping decoding for LDPC codes," in [*IEEE Vehicular Technology Conf.*], **61**(1), 709–713 (2005).
[24] Wadayama, T., Nakamura, K., Yagita, M., Funahashi, Y., Usami, S., and Takumi, I., "Gradient descent bit flipping algorithms for decoding LDPC codes," *IEEE Trans. on Comm.* **58**(6), 1610–1614 (2010).
[25] Yedidia, J., Chen, J., and Fossorier, M., "Generating code representations suitable for belief propagation," in [*Proc. 40th Allerton Conf. on Comm., Control and Computing*], (2002).
[26] Luong, H. V., Rakêt, L., Huang, X., and Forchhammer, S., "Side information and noise learning for distributed video coding using optical flow and clustering," *Submitted to IEEE Trans. Image Proc.* (2012).
[27] Luong, H. V. and Forchhammer, S., "Noise residual learning for noise modeling in distributed video coding," in [*Picture Coding Symposium*], (2012).
[28] Bjøntegaard, G., "Calculation of average PSNR differences between RD curves," tech. rep., VCEG Contribution VCEG-M33 (apr 2001).
[29] Rakêt, L., "Local smoothness for global optical flow," in [*International Conference of Image Processing (ICIP)*], (2012).
[30] Jang, M., Kang, J., and Kim, S., "A design of rate-adaptive LDPC codes for distributed source coding using peg algorithm," in [*The 2010 Military Comm. Conf. - Waveforms and Signal Proc. Track*], 277 – 282 (2010).

Bibliography

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

[2] V. Cantoni, L. Lombardi, and P. Lombardi, "Future scenarios of parallel computing: Distributed sensor networks," *Journal of Visual Languages and Computing*, vol. 18, pp. 484–491, 2007.

[3] N. Deligiannis, F. Verbist, A. C. Iossifides, J. Slowack, R. Van de Walle, P. Schelkens, and A. Munteanu, "Wyner-ziv coding video coding for wireless lightweight multimedia applications," *Eurasip Journal on Wireless Communications and Networking*, 2012.

[4] The grand challenges for engineering. [Online]. Available: http://www.engineeringchallenges.org/cms/challenges.aspx

[5] B. Tavli, K. Bicakci, R. Zilan, and J. M. Barcelo-Ordinas, "A survey of visual sensor network platforms," *Multimedia Tools and Applications*, vol. 60, no. 3, pp. 689–726, Oct. 2012.

[6] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. 19, no. 4, pp. 471–480, Jul. 1973.

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

[8] H. V. Luong, L. L. Rakêt, X. Huang, and S. Forchhammer, "Side information and noise learning for distributed video coding using optical flow and clustering," *IEEE Trans. Image Process.*, vol. 21, no. 12, pp. 4782–4796, Dec. 2012.

[9] (2007, Dec.) Discover project. [Online]. Available: http://www.discoverdvc.org/

[10] X. Huang and S. Forchhammer, "Cross-band noise model refinement for transform domain wyner-ziv video coding," *Signal Processing: Image Communication*, vol. 27, no. 1, pp. 16–30, 2012.

[11] H. V. Luong, X. Huang, and S. Forchhamme, "Multiple ldpc decoding using bitplane correlation for transform domain wyner-ziv video coding," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Proc.*, Praha, Czech Republic, Apr. 2011.

[12] ——, "Parallel iterative decoding of transform domain wyner-ziv video using cross bitplane correlation," in *IEEE Int. Conf. on Image Proc.*, Brussels, Belgium, Sep. 2011.

[13] X. Huang and S. Forchhamme, "Improved virtual channel noise model for transform domain wyner-ziv video coding," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Proc.*, Taipei, Taiwan, Apr. 2009.

[14] H. V. Luong, X. Huang, and S. Forchhammer, "Adaptive noise model for transform domain wyner-ziv video using clustering of dct blocks," in *IEEE Int. Workshop Multimedia Signal Proc.*, Hangzhou, China, Oct. 2011.

[15] H. V. Luong and S. Forchhamme, "Noise residual learning for noise modeling in distributed video coding," in *Picture Coding Symposium*, Krakow, Poland, May 2012.

[16] L. L. Rakêt, J. Søgaard, M. Salmistraro, H. V. Luong, and S. Forchhammer, "Exploiting the error-correcting capabilities of low density parity check codes in distributed video coding using optical flow," in *Proc. of SPIE, the Int. Society for Optical Engineering*, San Diego, USA, Aug. 2012.

[17] J. Slowack, S. Mys, J. Skorupa, N. Deligiannis, P. Lambert, A. Munteanu, and R. V. Walle, "Rate-distortion driven decoder-side bitplane mode decision for distributed video coding," *Signal Processing: Image Communication*, vol. 25, no. 9, pp. 660–673, 2010.

[18] P. L. Dragotti and M. Gastpar, Eds., *Distributed Source Coding: Theory, Algorithms and Applications*. Academic Press, Elsevier, 2009.

[19] R. Puri, A. Majumdar, and K. Ramchandran, "Prism: a video coding paradigm with motion estimation at the decoder," *IEEE Trans. Image Process.*, vol. 16, no. 10, pp. 1–13, Oct. 2007.

[20] R. Martins, C. Brites, J. Ascenso, and F. Pereira, "Refining side information for improved transform domain wyner-ziv video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 9, pp. 1327–1341, Sep. 2009.

[21] C. Brites, J. Ascenso, and F. Pereira, "Learning based decoding approach for improved wyner-ziv video coding," in *Picture Coding Symposium*, Krakow, Poland, May 2012.

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

[23] N. Deligiannis, J. Barbarien, M. Jacobs, A. Munteanu, A. Skodras, and P. Schelkens, "Side-information-dependent correlation channel estimation in hash-based distributed video coding," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1934–1949, Apr. 2012.

[24] J. Slowack, J. Skorupa, N. Deligiannis, P. Lambert, A. Munteanu, and R. V. de Walle, "Distributed video coding with feedback channel constraints," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 7, pp. 1014–1026, Jul. 2012.

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

[26] D. Varodayan, A. Aaron, and B. Girod, "Rate-adaptive codecs for distributed source coding," *EURASIP Signal Processing*, vol. 23, no. 11, pp. 3123–3130, 2006.

[27] A. Aaron, S. Rane, E. Setton, and B. Girod, "Transform domain wyner-ziv codec for video," in *Proc. SPIE VCIP*, San Jose, CA, Jan. 2004.

[28] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, and M. Ouaret, "The discover codec: architecture, techniques and evaluation," in *Picture Coding Symposium*, Lisbon, Portugal, Nov. 2007.

[29] X. Huang and S. Forchhamme, "Improved side information generation for distributed video coding," in *IEEE Int. Workshop Multimedia Signal Proc.*, Cairns, Australia, Oct. 2008.

[30] X. Huang, J. Ascenso, C. Brites, F. Pereira, and S. Forchhammer, "Distributed video coding with multiple side information," in *Picture Coding Symposium*, Chicago, USA, May 2009.

[31] X. Huang and S. Forchhamme, "Transform domain wyner-ziv video coding with refinement of noise residue and side information," in *SPIE Visual Communications and Image Processing*, HuangShan, China, Jul. 2010.

[32] D. Varodayan, D. Chen, M. Flierl, and B. Girod, "Wyner-ziv coding of video with unsupervised motion vector learning," *Eurasip Signal Processing Journal*, vol. 23, no. 5, pp. 369–378, 2008.

[33] Y. Vatis, S. Klomp, and J. Ostermann, "Inverse bit plane decoding order for turbo code based distributed video coding," in *Int. Conf. on Image Proc.*, San Antonio, USA, Sep. 2007.

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

[35] G. Bjøntegaard, "Calculation of average psnr differences between rd curves," VCEG Contribution VCEG-M33, Apr. 2001.

[36] D. Kubasov, J. Nayak, and C. Guillemot, "Optimal reconstruction in wyner-ziv video coding with multiple side information," in *IEEE Int. Workshop Multimedia Signal Proc.*, Chania, Greece, Oct. 2007.

[37] J. Skorupa, J. Slowack, S. Mys, N. Deligiannis, J. D. Cock, P. Lambert, C. Grecos, A. Munteanu, and R. V. de Walle, "Efficient low-delay distributed video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 4, pp. 530–544, Sep. 2011.

[38] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Int. Joint Conf. on Artificial Intelligence*, Vancouver, Canada, Aug. 1981.

[39] C. Brites and F. Pereira, "Correlation noise modeling for efficient pixel and transform domain wyner-ziv video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 9, pp. 1177–1190, Sep. 2008.

[40] G. R. Esmaili and P. C. Cosman, "Wyner-ziv video coding with classified correlation noise estimation and key frame coding mode selection," *IEEE Trans. Image Process.*, vol. 20, no. 9, pp. 2463–2474, Sep. 2011.

[41] R. L. Cannon, J. V. Dave, and J. C. Bezdek, "Efficient implementation of the fuzzy c-means clustering algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 2, pp. 248–255, Mar. 1986.

[42] X. Huang, L. L. Raket, H. V. Luong, M. Niesen, F. Lauze, and S. Forchhammer, "Multi-hypothesis transform domain wyner-ziv video coding including optical flow," in *IEEE Int. Workshop Multimedia Signal Proc.*, Hangzhou, China, Oct. 2011.

[43] Joint video team (jvt) reference software. [Online]. Available: http://iphome.hhi.de/suehring/tml/index.htm

[44] X. Fan, O. C. Au, and N. M. Cheung, "Transform-domain adaptive correlation estimation (trace) for wyner-ziv video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 11, pp. 1423–1436, Nov. 2010.

[45] S. Ye, M. Ouaret, F. Dufaux, and T. Ebrahimi, "Improved side information generation for distributed video coding by exploiting spatial and temporal correlations," *EURASIP Journal on Image and Video Processing*, vol. 2009, pp. 15–pages, 2009.

[46] A. Abou-Elailah, F. Dufaux, J. Farah, M. Cagnazzo, and B. Pesquet-Popescu, "Successive refinement of motion compensated interpolation for transform-domain distributed video coding," in *European Signal Processing Conference (EUSIPCO 2011)*, Barcelona, Spain, Aug. 2011.

[47] H. V. Luong, L. L. Rakêt, and S. Forchhammer, "Reestimation of motion and reconstruction for distributed video coding," *submitted to IEEE Trans. Image Process.*, 2013.

[48] J. Slowack, J. Skorupa, S. Mys, P. Lambert, C. Grecos, and R. V. Walle, "Distributed video coding with decoder-driven skip," in *Proceedings of the Mobimedia*, Sep. 2009.

[49] W. J. Chien and L. J. Karam, "Blast: Bitplane selective distributed video coding," *Multimedia Tools and Applications*, vol. 48, no. 3, pp. 437–456, 2010.

[50] J. C. Bezdek, R. J. Hathaway, M. J. Sabin, and W. T. Tucker, "Convergence theory for fuzzy c-means: Counterexamples and repairs," *IEEE Trans. Syst., Man, Cybern.*, vol. 17, pp. 873–877, Sep. 1987.