解碼端視訊預測技術與壓縮系統設計之研究

學生 ： 孫域晨　　指導教授 ： 蔡文錦 教授

李素瑛 教授

國立交通大學

資訊科學與工程研究所

摘　要

在傳統多媒體通訊系統中，解碼端有低運算複雜度的設計限制。隨著半導體技術進步，解碼端的運算能力越來越高。在解碼端可能採用先進的影像處理技術增加系統效能。本論文討論如何利用解碼端影像技術增進系統效能。我們將這樣的概念利用在兩個不同的應用上：分散式編碼系統和穩健編碼系統。

分散式編碼系統利用解碼端影像處理技術來作畫面間預測，系統中將複雜的畫面間預測從編碼端移到解碼端，因此可以大幅度降低編碼端的複雜度。我們提出利用解碼端畫面間預測誤差分析來最佳化編碼系統。我們提出預測誤差分類的概念，先分析解碼端預測誤差分佈然後分類，讓每一個分類裡的預測誤差統計特性接近。最後再分別將每一類誤差編碼。這樣分類編碼的方法可以節省 3%~10%的位元數。我們更進一步提出基於視覺分析的編碼方法，我們提出解碼端預測誤差的視覺分析，從視訊紋理分佈、運動分佈、、、等特性分析視覺誤差的分佈，接下來加強視覺誤差的編碼。這樣的基於視覺分析的編碼方法和傳統編碼法相比可以節省 10%~18%位元數，在主觀視覺上更有明顯的改善。我們提出了分散式編

碼系統中編碼方法的非平等數據保護版本，實驗結果顯示我們提出的編碼法可以提供非平等數據保護的特性，這個特性可以開啟很多分散式編碼系統新的研究方向。

我們也將解碼端影像處理技術用在穩健編碼系統的設計。我們基於階層式 B 畫面編碼架構，提出了兩個穩健編碼系統。第一個系統是利用一種混合的階層式 B 畫面編碼架構可以提升解碼端錯誤修復的效果。第二個系統是基於多重描述編碼，我們利用最佳化的方法來設計一套可以根據影像內容和網路品質動態調整的編碼方法，實驗結果顯示我們提出來的系統都有較好的編碼容錯能力。

關鍵字：分散式編碼、多重描述編碼、解碼端畫面間預測、影像修復

A Study of Decoder Side Video Prediction Technologies and Compression System Designs

Student: Yu-Chen Sun Advisor : Dr. Wen-Jiin Tsai

Dr. Suh-Yin Lee

Institute of Computer Science and Information Engineering
National Chiao Tung University

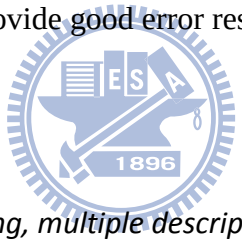
Abstract

In traditional multimedia communication systems, the computation complexity of the decoder is constrained. Due to advance of semiconductor technologies, the computation capability of decoder increases. It is possible to apply advanced image processing technologies on the decoder side. This thesis discusses how to utilize decoder side image processing technologies to improve system performance. We apply this concept on two applications: distributed video coding and robust video coding.

Distributed video coding adopts decoder side video prediction technologies to do inter-frames coding. The systems move complexity of inter-frame prediction from encoder to decoder, so the complexity of encoders decreases dramatically. We utilize analysis of inter-frame prediction error to optimize the compression system. The error classification is proposed: the decoder analyze macroblocks and classify them into different groups based on prediction error characteristics. Macroblocks in the same group have similar error characteristics and are encoded by a channel code. The classification based coding method would save 3%-10% bit rates. Furthermore, we

proposed a perceptual based coding method. Perceptual error on the decoder side is identified by analyzing image texture distribution, motion behavior, and so on. Wyner-Ziv (WZ) bits are then concentrated on correcting these perceptual error. Compared with original coding system, the proposed perceptual based coding can save 10%-18% bit rates. Subjective quality improvement is even more.

We also apply decoder side video processing technologies in robust video coding systems. Two systems are proposed based hierarchical-B coding structures. The first system adopts a hybrid hierarchical-B structure to improve the quality of error concealment method. The second system is based on multiple description video coding. We proposed a R-D optimization framework that adaptively allocates redundancy based on video content and channel conditions. Experimental results show the proposed system can provide good error resilient capability.



Keywords: *distributed video coding, multiple description video coding, decoder side inter-frame prediction, video restoration.*

Acknowledgement

能夠順利完成博士論文首先要感謝我的指導教授：蔡文錦老師 和 李素瑛老師。蔡文錦老師研究上提供我各種資源，論文討論中提供我各種想法，一路上給了我很多的建議和支持。無論大小問題，蔡文錦老師都會親切用心的替學生著想，給予各種幫助，真的很感激老師的幫忙。在博士研究過程中，李素瑛老師就像一個慈祥的大家長一樣無私的給予學生各種幫助，在學生低潮的時候給學生鼓勵和替學生打氣，學生遇到瓶頸時，老師立刻就會幫學生想各種解決的方法。老師時時刻刻都能給學生一股安定的力量，這樣的力量在我的博士生涯中是非常重要的。很感激兩位老師的幫忙，因為有老師的幫忙，我才能夠完成今天的博士論文。

也很感謝撥冗參加學生口試的林嘉文教授、杭學鳴教授、張隆紋教授、張寶基教授和廖弘源教授，委員們的建議對我的研究論文幫助很大。

也謝謝一路上學長姐、同學和學弟妹的幫忙：蕭永慶學長、何健鵬學長和陳涛紋學長一路上給予我各種建議，也教了我很多事情；林憲正、林喚宇和我一起互相鼓勵，很高興我們都畢業了；還有一起準備資格考的讀書會同學們；謝謝實驗室小靜、小啾、巧安、閃六、建儒、小高、史達 和 SA，很高興可以和你們共度歡樂的研究時光。

當然一定要謝謝我的家人一路上的鼓勵，這讓我有堅持下去的力量。

最後再次感謝所有幫助我的人，謝謝你們。

Table of Contents

摘要 i

ABSTRACT ..... **iii**

ACKNOWLEDGEMENT ..... **v**

CHAPTER 1 INTRODUCTION **1**

1.1 DISTRIBUTED VIDEO CODING 3

1.2 ROBUST VIDEO CODING 5

CHAPTER 2 DISTRIBUTED VIDEO CODING **10**

2.1 MACROBLOCK GROUPING FOR WZ CODING 12

2.1.1 Side Information Error Analysis 12

2.1.2 DVC With Macroblock Grouping 14

2.1.2.1 System Architecture 15

2.1.2.2 Side Information Generation Algorithm 18

2.1.2.3 Error Model Estimator 20

2.1.3 Experimental results 23

2.1.3.1 Test Conditions 24

2.1.3.2 WZ Coding Efficiency Evaluation 24

2.2 PERCEPTUAL BASED DISTRIBUTED VIDEO CODING 25

2.2.1 Side Information Perceptual Analysis 25

2.2.2 The proposed DVC framework 28

2.2.2.1 Texture Distribution Similarity (TDS) Analysis 29

2.2.2.2 Motion Consistency (MC) Analysis 30

2.2.2.3 Texture Structure Consistency (TSC) Analysis 31

2.2.2.4 Valid Motion Projection (VMP) Analysis 32

2.2.2.5 Determination of Regions of Interest (ROI) 33

2.2.2.6 Impact of large GOP sizes on the proposed perceptual metrics 36

2.2.2.7 Complexity analysis of the proposed perceptual DVC codec 37

2.2.3 Experimental Results 40

2.2.3.1 R-D Performance Evaluation 40

2.2.3.2 Visual Quality Comparisons 45

CHAPTER 3 ROBUST VIDEO CODING **52**

3.1 ERROR-RESILIENT VIDEO CODING USING MULTIPLE REFERENCE FRAMES 53

3.1.1 Related Works 53

3.1.1.1 Rate-Distortion Optimization in JM 53

3.1.1.2 Expected End-to-End Distortion Model 54
3.1.2 The proposed method 55
3.1.2.1 Candidate Reference Frames 55
3.1.2.2 Error Resilient RDO 57
3.1.3 Computational Cost Reduction 58
3.1.3.1 Reference Frame Selection 59
3.1.3.2 Reference Frame Skipping 59
3.1.3.3 Long Term Motion Search Center Prediction 60
3.1.3.4 Summarization of the Proposed Method 62
3.1.4 Experimental Results 63
3.1.4.1 Overall Rate Distortion Performance 63
3.1.4.2 Mismatch of Packet Loss Rate 66
3.1.4.3 Effect of ER-frames 67
3.1.4.4 Computational Cost 68
3.2 ERROR RESILIENT VIDEO CODING BASED ON HIERARCHICAL B PICTURES 71
3.2.1 Introduction 71
3.2.2 Motivation 71
3.2.3 Proposed Method 74
3.2.4 Estimation of lost pictures 77
3.2.5 Experimental Results 80
3.2.5.1 Effects of Hybrid Structures 80
3.2.5.2 Effects of Hybrid Structure Variations 80
3.2.5.3 Packet Loss Performance 81
3.2.5.4 Error Free Performance 84
3.3 RATE-DISTORTION OPTIMIZED MODE SELECTION METHOD FOR MULTIPLE DESCRIPTION VIDEO CODING 87
3.3.1 Proposed MDC based on a hierarchical B-picture structure 88
3.3.1.1 The encoder architecture 89
3.3.1.2 The decoder estimation methods 91
3.3.2 Rate-Distortion mode selection method 92
3.3.2.1 Rate-Distortion optimization on an ideal MDC channel 92
3.3.2.2 Rate-Distortion estimation 95
3.3.2.3 Rate-Distortion optimization on a packet loss channel 99
3.3.2.4 Summary of proposed Rate-Distortion mode selection method 101
3.3.3 Experimental Result 103
3.3.3.1 Packet Loss Performance 103
3.3.3.2 Side Reconstruction Performance 109
CHAPTER 4 CONCLUSION AND FUTURE WORKS 113

4.1 COMMENTS ON DISTRIBUTED VIDEO CODING 113
4.2 COMMENTS ON ROBUST VIDEO CODING 114

REFERENCE 116

List of Figures

FIG. 1 SI OF 70TH FRAME OF FOREMAN (LEFT), AND ITS ERROR IMAGE (RIGHT). 13

FIG. 2 PROPOSED DVC ARCHITECTURE WITH MACROBLOCK GROUPING 14

FIG. 3 QUANTIZATION MATRIX FOR WZ FREQUENCE COMPONENT 17

FIG. 4 BI-DIRECTIONAL MOTION REFINEMENT. *C* IS THE SI BLOCK TO BE PREDICTED. *A* AND *B* ARE MATCHING BLOCKS THAT ARE LOCATED ALONG THE TRANSLATIONAL MOTION PATH FROM KEY FRAME *N* TO *N*+1, RESPECTIVELY. 20

FIG. 5 MOTION REFINEMENT USING HIERARCHICAL SEARCH. THE MV REPRESENTED IN DASHED LINE IS THE REFINED MV FOR SMALL BLOCKS TO BE ESTIMATED FROM THE COASER MVS REPRESENTED IN SOLID LINES FOR LARGE BLOCKS. 20

FIG. 6 SIDE INFORMATON ERROR HISTOGRAMS. 22

FIG. 7 ERROR CLASSIFICATION RESULT. 23

FIG. 8 AN EXAMPLE OF SI WITH GOOD PERCEPTUAL QUALITY BUT LOW PSNR. THE PSNR OF THE SI FRAME IS ONLY 17.5 DB, BUT ITS VISUAL QUALITY IS COMPARABLE TO THE NEIGHBORING KEY FRAMES (AROUND 30 DB). 26

FIG. 9 PERCEPTUAL-BASED RECONSTRUCTION OF SI PREDICTION ERRORS. THE RECONSTRUCTED WZ FRAME IN (B) TRIES TO CORRECT THE PREDICTION ERRORS OF THE WHOLE SI FRAME USING 12.38 KBITS OF LDPCA CODES. THE RECONSTRUCTED WZ FRAME IN (D) ONLY CORRECTS A RECTANGULAR AREA THAT CONTAINS THE FACE OF FOREMAN USING 10.46 KBITS OF LDPCA CODES. 28

FIG. 10 THE PROPOSED DVC ARCHITECTURE. GRAY BLOCKS ARE PROPOSED MODULES. 29

FIG. 11 EXAMPLES OF SI FRAMES (LEFT) AND DETECTED VISUALLY DISTORTED MACROBLOCKS (RIGHT). 34

FIG. 12 AN EXAMPLE OF THE 72ND SI FRAME AND THE DETECTED VISUALLY DISTORTED MACROBLOCKS. 34

FIG. 13 NEIGHBORHOOD STRUCTURE OF THE ROI REFINEMENT PROCESS. THE SQUARES ARE THE MACROBLOCKS UNDER CONSIDERATION AND THE CIRCLES ARE THEIR NEIGHBORS. 35

FIG. 14 EXAMPLES OF ROI DETECTION RESULTS. MACROBLOCKS WITH NORMAL GRAY LEVELS ARE IN Ω_{ROI}. 36

FIG. 15 QCIF SEQUENCES R-D PERFORMANCE COMPARISONS USING PSNR AND SSIM. THE AVERAGE BD PSNR GAIN OVER DISCOVER IS 0.71DB, AND THE AVERAGE SSIM GAIN IS 0.016. 43

FIG. 16 R-D PERFORMANCE COMPARISONS USING PSNR AND SSIM. THE AVERAGE PSNR BD GAIN OVER DISCOVER IS 0.41DB, AND THE AVERAGE SSIM GAIN IS 0.010. 44

FIG. 17 THE PSNRS OF RECONSTRUCTED FRAMES OF THE FOREMAN (160 KBPS) AND COASTGUARD (100 KBPS) SEQUENCES USING THE PROPOSED DVC CODEC. 46

FIG. 18 VISUAL COMPARISONS BETWEEN THE PROPOSED CODEC (TOP ROW) AND THE DISCOVER CODEC

(BOTTOM ROW) AT FRAME POSITIONS WITH HIGHEST PSNR VARIATIONS. THE BITRATE OF THE PROPOSED CODEC IS 160.0 KBPS, AND THE BITRATE OF DISCOVER IS 161.6 KBPS 47
FIG. 19 THE SI FRAME OF THE PROPOSED CODEC (LEFT) USED IN 0, AND ITS ERROR IMAGE (RIGHT). THE SI FRAME OF THE DISCOVER CODEC IS NOT AVAILABLE 47
FIG. 20 VISUAL COMPARISONS BETWEEN THE PROPOSED CODEC (TOP ROW) AND THE DISCOVER CODEC (BOTTOM ROW) AT FRAME POSITIONS WITH HIGHEST PSNR VARIATIONS. THE BITRATE OF THE PROPOSED CODEC IS 99.4 KBPS, AND THE BITRATE OF THE DISCOVER CODEC IS 101.4 KBPS. 48
FIG. 21 THE SI FRAME OF THE PROPOSED CODEC (LEFT) USED IN 0, AND ITS ERROR IMAGE (RIGHT). THE SI FRAME OF THE DISCOVER CODEC IS NOT AVAILABLE 48
FIG. 22 VISUAL COMPARISONS BETWEEN THE PROPOSED CODEC (TOP ROW) AND THE DISCOVER CODEC (BOTTOM ROW) AT FRAME POSITIONS WITH NOTICABLE VISUAL IMPROVEMENTS. THE BITRATE OF THE PROPOSED CODEC IS 127.8 KBPS, AND THE BITRATE OF THE DISCOVER CODEC IS 131.5 KBPS. 49
FIG. 23 THE SI FRAME OF THE PROPOSED CODEC (LEFT) USED IN 0, AND ITS ERROR IMAGE (RIGHT). THE SI FRAME OF THE DISCOVER CODEC IS NOT AVAILABLE 49
FIG. 24 VISUAL COMPARISONS BETWEEN THE PROPOSED CODEC (TOP ROW) AND THE DISCOVER CODEC (BOTTOM ROW) AT FRAME POSITIONS WITH NOTICEABLE VISUAL IMPROVEMENTS. THE BITRATE OF THE PROPOSED CODEC IS 134.1 KBPS, AND THE BITRATE OF THE DISCOVER CODEC IS 134.4 KBPS. 50
FIG. 25 THE SI FRAME OF THE PROPOSED CODEC (LEFT) USED IN 0, AND ITS ERROR IMAGE (RIGHT). THE SI FRAME OF THE DISCOVER CODEC IS NOT AVAILABLE 50
FIG. 26 VISUAL COMPARISONS BETWEEN THE PROPOSED CODEC (TOP ROW) AND THE DISCOVER CODEC (BOTTOM ROW) AT FRAME POSITIONS WITH POOREST SI QUALITY. THE KEY FRAMES FOR BOTH CODECS ARE THE SAME. THE WZ RATE FOR THE CORRESPONDING FRAME ARE THE SAME TOO. THE BITRATE OF THE PROPOSED CODEC IS 150.4 KBPS, AND THE BITRATE OF THE PROPOSED CODEC IS 161.6 KBPS 51
FIG. 27 THE SI FRAMES OF THE PROPOSED CODEC (TOP ROW) USED IN 0, AND ITS ERROR IMAGE (BOTTOM ROWS). 51
FIG. 28 ER-FRAMES AS PART OF REFERENCE FRAMES. 57
FIG. 29 MOTION VECTOR COMPOSITION USING FDVS AND ACCUMULATED FDVS 62
FIG. 30 THE FLOW CHART OF THE PROPOSED MRF-MCP WITH FAST MOTION ESTIMATION 63
FIG. 31 R-D PERFORMANCE COMPARISON USING FIVE REFERENCE FRAMES 65
FIG. 32 PERFORMANCE FOR MISMATCH WITH AN ASSUMED PLR OF (A) 5%; (B) 10% 66
FIG. 33 FRAME REFERENCE DISTRIBUTION 67
FIG. 34 EXECUTION TIME RATIO OF DIFFERENT METHODS 69
FIG. 35 PERFORMANCE WITH AND WITHOUT COMPUTATIONAL TIME REDUCTION TECHNIQUES. 70
FIG. 36 HIERARCHICAL B-PICTURE PREDICTION STRUCTURE 74
FIG. 37 EXPERIMENTAL SETTING FOR DIFFERENT COMBINATIONS OF MOTION FRAMES (DF1, DF2, AND

DF3) AND DATA FRAMES (MF1, MF2, AND MF3). 74
FIG. 38 THE PROPOSED HYBRID MODEL BASED ON HIERARCHICAL B STRUCTURE 75
FIG. 39 ARCHITECTURE OF THE PROPOSED HYBRID MODEL HN+M 77
FIG. 40 MOTION INTERPOLATION, COMPOSITION, AND EXTRAPOLATION 79
FIG. 41 CODING STRUCTURES OF HYBRID MODEL, H_{4+4}, AND ORIGINAL MODEL. 82
FIG. 42 PACKET-LOSS PERFORMANCE OF FOUR HYBRID MODELS. 84
FIG. 43 RATE-DISTORTION PERFORMANCE COMPARISON IN ERROR FREE ENVIRONMENT. 86
FIG. 44 THE ENCODER ARCHITECTURE OF THE PROPOSED MDC SYSTEM 89
FIG. 45 SPATIAL SPLITTING OF THE PROPOSED MDC. 90
FIG. 46 PROPOSED MDC BASED ON HIERARCHICAL B-PICTURE PREDICTION. 91
FIG. 47 AN EXAMPLE OF R-D OPTIMIZATION. 94
FIG. 48 ILLUSTRATION OF ERROR WEIGHT. 96
FIG. 49 FITTING RESULT OF PROPAGATION DECAYS FACTORS, αPD. 98
FIG. 50 FITTING RESULT OF B IN EQ. (15). 102
FIG. 51 R-D PERFORMANCE OF THE FORMAN SEQUENCE. (A) PACKET LOSS RATE = 1%. (B) PACKET LOSS RATE = 5%. (C) PACKET LOSS RATE = 10%. (D) PACKET LOSS RATE = 20%. 105
FIG. 52 R-D PERFORMANCE OF THE NEWS SEQUENCE. (A) PACKET LOSS RATE = 1%. (B) PACKET LOSS RATE = 5%. (C) PACKET LOSS RATE = 10%. (D) PACKET LOSS RATE = 20%. 106
FIG. 53 R-D PERFORMANCE OF THE STEFAN SEQUENCE. (A) PACKET LOSS RATE = 1%. (B) PACKET LOSS RATE = 5%. (C) PACKET LOSS RATE = 10%. (D) PACKET LOSS RATE = 20%. 107
FIG. 54 R-D PERFORMANCE OF THE TABLE TENNIS SEQUENCE. (A) PACKET LOSS RATE = 1%. (B) PACKET LOSS RATE = 5%. (C) PACKET LOSS RATE = 10%. (D) PACKET LOSS RATE = 20%. 108
FIG. 55 SIDE DECODING R-D PERFORMANCE. (A) FOREMAN. (B) NEWS. (C) STEFAN. (D) TABLE TENNIS. 111
FIG. 56 CENTER DECODING R-D PERFORMANCE. (A) FOREMAN. (B) NEWS. (C) STEFAN. (D) TABLE TENNIS. 112

List of Tables

TABLE. I BIT RATE REDUCTION WITH PROPOSED MACROBLOCK GROUPING 24
TABLE. II IMPACT OF GOP SIZE ON THE PROPOSED PERCEPTUAL METRICS. 37
TABLE. III ENCODING TIME COMPARISON FOR FOREMAN, QCIF@15FPS. 38
TABLE. IV DECODING TIME COMPARISON FOR FOREMAN, QCIF@15FPS. 38
TABLE. V BREAKDOWN OF ENCODING TIME PER FRAME (IN MSEC) FOR THE PROPSOED CODEC. 39
TABLE. VI BREAKDOWN OF DECODING TIME PER FRAME (IN MSEC) FOR THE PROPSOED CODEC. 39
TABLE. VII QUANTIZATION SETTING (QP,QM) OF DVC CODECS IN THE EXPERIMENTS. 41
TABLE. VIII BD RESULTS OF THE TEST SEQUENCES. 41
TABLE. IX EXPERIMENTAL RESULT FOR ALL COMBINATIONS OF MOTION FRAMES AND DATA FRAMES. 74
TABLE. X MINIMAL PIXEL RECOVERING DISTANCE FOR LOST FRAMES AT DIFFERENT HIERARCHICAL LEVELS 74
TABLE. XI PERFORMANCE COMPARISON BETWEEN HYBRID MODEL, H4+4, AND THE ORIGINAL MODEL. BOTH MODELS ENCODE FOREMAN SEQUENCE (CIF) AT 800KBPS. 83
TABLE. XII PACKET-LOSS PERFORMANCE COMPARISON. 83
TABLE. XIII THE BIT-RATE REDUNDANCY COMPARISON. THE REDUNDANCY IS DEFINED AS THE BJONTEGARRD BIT-RATE DIFFERENCE[55] BETWEEN JM AND EACH METHOD. 86
TABLE. XIV SUMMARY OF THE CASES FOR DIFFERENT ESTIMATION METHODS. 92
TABLE. XV BD RESULTS OF THE PROPOSED FRAMEWORK ON PACKET LOSS CHANNELS. THE COLUMN OF "COMPARING WITH THE MDC SYSTEM IN [89]" SHOWS THE BD DIFFERENCE BETWEEN THE PROPOSED METHOD AND THE MDC SYSTEM IN [89]; THE COLUMN OF "COMPARING WITH DO-MDC" SHOWS THE DIFFERENCE BETWEEN THE PROPOSED METHOD AND DO-MDC. 104
TABLE. XVI SIDE DECODING BD RESULTS OF THE PROPOSED FRAMEWORK. THE COLUMN OF "COMPARING WITH THE MDC SYSTEM IN [89]" SHOWS THE BD DIFFERENCE BETWEEN THE PROPOSED METHOD AND THE MDC SYSTEM IN [89]; THE COLUMN OF "COMPARING WITH DO-MDC" SHOWS THE DIFFERENCE BETWEEN THE PROPOSED METHOD AND DO-MDC. 109
TABLE. XVII CENTER DECODING BD RESULTS OF THE PROPOSED FRAMEWORK. THE COLUMN OF "COMPARING WITH THE MDC SYSTEM IN [89]" SHOWS THE BD DIFFERENCE BETWEEN THE PROPOSED METHOD AND THE MDC SYSTEM IN [89]; THE COLUMN OF "COMPARING WITH DO-MDC" SHOWS THE DIFFERENCE BETWEEN THE PROPOSED METHOD AND DO-MDC. 110

Chapter 1 Introduction

In traditional video communication systems, the computational complexity of the decoder is constrained because the computational power of the decoder is usually low. However, in recent years, the computational power of the decoder is increasing due to advance of semiconductor technologies. Therefore, it is possible to apply advanced image processing technologies on the decoder side to improve system performance. Moreover, lots of new applications, such as mobile visual network, have different system requirements: low complex encoding. In these applications, the computation power of the decoder is better than that of the encoder. The advanced compression tools on the decoder side play an important role to improve the system performance.

In this thesis, we study decoder side image prediction technologies. Different image processing technologies on the decoder are proposed to do inter-frame prediction or image restoration. We combine the proposed technologies into several coding systems. The proposed systems are categorized into two parts:

The first part is distributed video coding system (DVC) [1]. In recent years, DVC has been considerably investigated because, theoretically, DVC allows flexible distribution of coding complexity between the encoder and the decoder without losing compression efficiency. This characteristic makes DVC a potential solution for emerging applications such as Mobile 2.0, surveillance systems, and sensor networks where encoders have limited computation ability due to power consumption constraint[2]. The theoretical bound for this coding scheme has been investigated by Slepian-Wolf [3] and Wyner-Ziv[4]. Based on these studies, it is possible to shift coding complexity from the encoder to the decoder using distributed principle while still approaching the coding efficiency of the traditional closed-loop coding schemes.

The second part is robust video coding. During the stage of transmission through the error-prone environment, packet loss might occur. In the case of transmission of compressed video sequences, robust video coding plays an important role to handle packet loss that may result in a completely damaged stream at the decoder side. In recent years, several robust video coding methods have been developed, such as forward error correction (FEC) [5], intra/inter coding mode selection [6], temporal error concealment [7], and multiple description coding (MDC)[8][9].

In this thesis, five technologies have been proposed. Some are for DVC; while others for robust video coding as listed below:

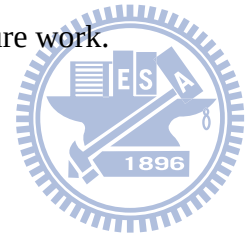
DVC:

- an Adaptive source grouping coding method
- a perceptual-based decoder-side skip mode strategy

Robust Video Coding:

- error-resilient video coding using multiple reference frames
- error resilient video coding based on hierarchical b pictures
- rate-distortion optimized mode selection method for multiple description video coding

The following sections of Chapter 1 introduce the related works of DVC and robust video coding. The proposed technologies of DVC and robust video coding are depicted in Chapter 2 and Chapter 3, respectively. Finally, Chapter 4 makes a conclusion and discusses the future work.



1.1 Distributed Video Coding

In many DVC implementations [1][10]-[12], a video source sequence is divided into two interleaving sub-sequences: key frame subsequences and Wyner-Ziv (WZ) frame subsequences. Key frames are encoded using a traditional low-complexity encoder (such as the motion JPEG encoder or the intra-coder of any video codecs). For WZ frames, the decoder first applies sophisticated algorithm to construct a predictor (called side information) from the previously received data (typically key frames). On the encoder side, it takes the original video frames as input and applies a low-complexity algorithm to generate error-correction bits (refer to as the WZ bits) that can help the decoder correct any side information (SI) prediction errors such that the resulting WZ frames are close to the original video frames. The key components in a DVC framework are the SI generator and the WZ bits encoder. For SI generator, many researchers adopt motion compensated frame interpolation method [13][14]. In [13], an advanced true motion estimation technique is proposed to improve SI quality; in [14], error surface of motion estimation is integrated into WZ decoding iteration to find MAP solution of motion field for improving the quality of the reconstructed frame. For WZ encoding, different channel codes are adopted in several implementations [15][16]. Note that the efficiency of the channel codes is highly dependent on the side information error statistics. In [18][19], the SI error model is studied, and then a MAP-based decoding method is proposed.

In traditional closed-loop codec, a key technique to improve coding efficiency is the mode decision tool. Based on the characteristics of the next group of data to be coded, the mode decision module selects the most appropriate set of coding tool that achieves best coding efficiency. Ascenso et al. [20] propose to adaptively adjust the size of GOP according to the motion activity. Zhang et al. [21] propose to use the difference of co-located pixels as a mode decision measure to switch between zero-motion skip and WZ code, and [22][23] propose similar idea between WZ-coding and intra-coding mode decision. In [24], co-located pixel error measure is also used to decide quantization level of each region. However, the co-located pixel error measure fails to recognize that a textured region with consistent translational motion should be coded in WZ mode. A more sophisticated mode decision measure is proposed in [25], where a low complexity estimation of optical flow is used at the

encoder side to predict the quality of the side information generated at the decoder (and hence the efficiency of WZ coder). Although the concept of mode decision is important in fine-tuning the performance of DVC, it does not address the crucial issue of how to improve the coding efficiency of the WZ coding tool.

In this thesis, a new coding strategy with adaptive source grouping for prioritized, unequal error correction is proposed. In short, the proposed technique classifies side information data into several groups based on estimated prediction error statistics. Different groups of video data are channel-coded together at the encoder side. The decoder requests WZ bits to correct corresponding groups of SI data out-of-order, based on their error levels. An early concept of the adaptive source grouping principle is presented in [26], where a binary decision is used to classify the importance of each macroblock. Comparing with previous implementation, in this thesis, more advanced SI generation algorithm is adopted to produce better SI as well as error estimates. Furthermore, multi-level classification techniques are adopted. As the experimental results in this thesis show, accurate error model estimation is the key to the efficiency of WZ coders.

In addition to prediction error analysis, we also propose a very different approach of perceptual-based decoder-side skip mode strategy in this chapter. The proposed technique comes from the key observation that SI frames predicted using motion-projection algorithms often contain image areas with large prediction errors (in MSE sense) but small visual distortions. One such example is a video sequence of a low-motion scene taken by a shaky camera. If we remove every other frame and interpolate the missing frames with motion-projection algorithms, the resulting video may become smoother without major visual distortions except at image boundaries. However, to reconstruct the original shaky video from the interpolated frames using channel codes requires significant amount of parity bits, which is not worthwhile from perceptual rate-distortion perspective. In short, the proposed technique performs perceptual-based analysis to determine the SI regions where visual distortions are noticeable, and only uses channel codes to correct these regions.

1.2 Robust Video Coding

In recent years, several robust video coding methods have been developed, such as forward error correction (FEC) [5], intra/inter coding mode selection [6] ,temporal error concealment [7], and multiple description coding (MDC)[8][9]. In this chapter, we proposed three different approaches.

One way to reduce error propagation is to insert intra-coded macroblocks in inter-coded frames [6]. The common drawback of these techniques is that network conditions and error concealment are not considered. On the other hand, a number of error-resilient RDO (ER-RDO) techniques accounting for the impact of packet loss have been proposed [28]-[30]. Recursive optimal per-pixel estimate (ROPE) method [28] has been recognized as an effective method to estimate the expected end-to-end distortion. In [29], the expected end-to-end distortion is estimated in a manner of independently operating K copies of the random variable channel behavior and decoder pairs in the encoder. Since this method suffers from high computational cost, Y. Zhang *et al*. [30] proposed a new model, in which the overall distortion is calculated and stored in block-level instead of pix-level. Since Y. Zhang *et al*.'s ER-RDO focuses on coding-mode selection, it combats the error propagation by intra-macroblock insertion. However, intra macroblocks have much lower coding efficiency than inter macroblocks so that the overall coding efficiency may be degraded significantly if a large number of intra macroblocks are inserted.

Alternatively, Yang *et al*. [31] recognized that motion prediction has a considerable impact on error resilience. As an example, predicting current block from an intra-coded block is a more robust choice than predicting it from an inter-coded block that may entail propagated error. Therefore, they integrate ROPE-based distortion estimation into RDO for motion vector selection. A number of work addressing similar objectives can also be found in [32][33]. However, in their works, only single reference frame is adopted. If their approach is applied for multiple reference frames, impractical extra complexity will be induced. Wan *et al*. addressed this problem and proposed a method in [34] to solve it.

Multiple reference frame motion compensated prediction (MRF-MCP) is adopted in H.264/AVC to enhance the coding efficiency of the compressed video stream. This feature has been investigated earlier in H.263 for error resilience. Rather than using

near frames for reference, [35] develop a long-term reference frame selection method, in which for every k^{th} frame, select n macroblocks, called *Periodic Macroblock*, to predict from the frame that is k frames away. In their approach, the values of both n and k are predefined constants; and the locations of Periodic Macroblocks are determined according to expected distortion of macroblocks. The authors in [36] proposed an alternative scheme, where every P-frame selects a number of macroblocks called *Robust Macroblock*, to predict from the nearest intra-coded frame. In their approach, the number of Robust Macroblocks per frame is a fixed number; and the locations of Robust Macroblocks are also determined according to a distortion estimation model. A number of works using long-term reference frames can also be found in [37]-[39]. These works adopt dual frames, one long-term and one short- term reference frames, to achieve error resilience with low computational cost. The results in [35]-[39] have shown that error propagation can be effectively suppressed by using long-term reference frames. However, the approach in [35] used predefined constants for some parameters, such as the period and the number of Periodic Macroblocks, which are intuitively related to channel conditions and video characteristics; the approach in [36] assumed that all intra-coded frames are intact and fixed the number of Robust Marcoblocks for every P-frame; and approach in [38] selected every 10^{th} frame as a long-term reference and encoded these frames with a quantization parameter (QP) lower by 7 compared to the general QP for the entire sequence. These predefined constants and constraints make their approaches hard to adapt to various content characteristics and channel conditions.

Error concealment (EC) techniques are another key issue of robust video coding. In case of packet loss, EC techniques can be used to recover the lost information. There are many existing EC algorithms, such as spatial interpolation, frequency domain interpolation, and temporal compensation based on inter-frame correlation. Among them, temporal error concealment is the most widely used approach, especially to combat the whole-frame loss problem, when hierarchical B-picture coding is used. The simplest temporal EC method is frame copy, in which each damaged macroblock is directly replaced by the co-located one in the temporally previous picture. Although it seems to be simple and fast, it suffers from large distortion in case of fast motion in the erroneous block area. Thus, some methods based on motion compensation have been proposed, which replace the lost block with

the one from previous frame that is shifted to compensate the estimated motion. To eliminate the complexity of motion estimation in these methods, an approach based on temporal direct mode [76] has been adopted in H.264/AVC (SVC). It derives the motion vector for each block in the lost B-picture according to the motion vector of the co-located block in the temporally subsequent reference picture. This method has low computational complexity due to no motion estimation. However, its EC efficiency is usually unsatisfactory. Thus, improved approaches have been proposed. Ji et al. [7] proposed a method based on *enhanced temporal direct mod*e. in which the motion vectors for each block are allowed to be derived from the temporally not only subsequent reference picture, as in H.264/AVC, but also previous reference picture. Thus, the approach in [7] derived motion of each block in the lost picture from the motion vector of the co-located block in the temporal subsequent or previous reference picture. In addition, they also proposed that the motion of the damaged block can be derived from the motion vectors of the co-located blocks in the temporally neighboring left and/or right B-pictures at next higher temporal level. Their experimental results show that motion prediction in this way can improve the EC performance.

Multiple description coding (MDC) is another approach of robust video coding. MDC encodes a single video stream into two or more equally important sub-streams, called descriptions, each of which can be decoded independently. Different from the traditional single description coding (SDC) where the entire video stream (single description) is sent in one channel, in MDC, these multiple descriptions are sent to the destination through different channels, resulting in much less probability of losing the entire video stream (all the descriptions), where the packet losses of all the channels are assumed to be independently and identically distributed. The first MD video coder, called multiple description scalar quantizer (MDSQ)[81], has been realized in 1993 by Vaishampayan who proposed an index assignment table that maps a quantized coefficient into two indices each could be coded with fewer bits. Due to effectiveness in providing error resilience, a variety of research on different MDC approaches had been proposed afterwards. These approaches can be intuitively classified through the stage where it split the signal, such as, frequency domain[81][82], spatial domain [83][84], and temporal domain [85][86]. In our previous works [87], a hybrid MDC method has been proposed, which applies MDC first in spatial domain to split motion

compensated residual data, and then in frequency domain to split quantized coefficients. A hybrid MDC method with spatial and temporal splitting was proposed in [88] and a hierarchical B-picture based hybrid MDC method was proposed in [89]. The results in [87]-[89] show that, by properly utilizing more than one splitting technique, the hybrid MDC method can improve error-resilient performance.

To improve coding performance, some researchers proposed to optimize the encoding coefficient for rate-distortion performance. In [91], a R-D optimization technique is proposed for the MDC with one descriptor containing all DCT coefficients and the second one containing only few low frequency coefficients. The R-D technique aims at optimizing the number of pruning coefficients. In [92], the method to find out optimized quantization parameters was proposed for the MDC based on H.264/AVC redundant slices[93]. Then, Lin et al.[94] extended the method from the slice level to the macroblock level.

There are two major benefits of the rate-distortion optimization concept. First, video contents vary spatially and temporally, so it would be inefficient to use a fixed encoding method to encode whole contents. In addition, the importance of different parts of video contents may be different, so adopting an unequal error protection can achieve better rate-distortion performance. Second, the channel condition also varies over time, so a mechanism to dynamically adjust protection level is necessary. With rate-distortion optimization, the encoder can change coding strategy according to video contents and channel conditions, and therefore improve the performance. However, the previous optimization frameworks were based on the specific MDC systems. Since a variety of new MDC coding tools are being proposed and each tool has different characteristics. To enable the rate-distortion optimization concept on these MDC tools, a general framework is desirable.

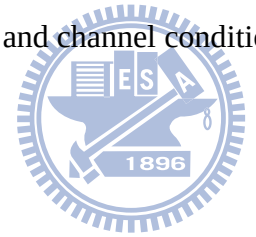
In this thesis, three technologies of robust video coding are proposed:

First, a MRF-MCP based error resilient scheme is proposed, which employs the nearest *error-resilient frame (ER-frame)* as one of the reference frames and adopts error-resilient RDO (ER-RDO) for optimal reference block selection. The ER-frame in our approach is a frame capable of suppressing error propagation, which can be an intra-coded frame, or an inter-coded frame with high ratio of intra-coded macroblocks. Incorporating ER-RDO in our approach is for the purpose of making the choice of the *number* and *location* of the macroblocks referring to ER-frames to be decided

adaptively by using rate-distortion technique. Significant performance gains in the experiments confirm that our approach has substantial improvement over competed schemes in providing error resilience using MRF-MCP. Besides, some techniques based on our error resilient scheme are further proposed to reduce the computational cost. These techniques include moving ER-RDO from motion vector to reference frame selection, skipping unnecessary reference frames, and predicting precise motion search centers.

Second, an error resilient coding based on hierarchical B pictures is proposed. In this approach, a new hierarchical coding structure which combines two conventional hierarchical coding structures is employed to reduce the distance between a lost picture and its recovering pictures. In addition, based on the new structure, an improved estimation method is also proposed to further increase the accuracy of recovered motion.

Third, a rate-distortion optimization framework for MDC systems is proposed. With the proposed framework, the encoder can dynamically adjust coding strategy according to both video contents and channel conditions.



Chapter 2 Distributed Video Coding

In this chapter, a new coding strategy with adaptive source grouping for prioritized, unequal error correction is proposed. In short, the proposed technique classifies side information data into several groups based on estimated prediction error statistics. Different groups of video data are channel-coded together at the encoder side. The decoder requests WZ bits to correct corresponding groups of SI data out-of-order, based on their error levels. An early concept of the adaptive source grouping principle is presented in [26], where a binary decision is used to classify the importance of each macroblock. Comparing with previous implementation, in this thesis, more advanced SI generation algorithm is adopted to produce better SI as well as error estimates. Furthermore, multi-level classification techniques are adopted. As the experimental results in this thesis show, accurate error model estimation is the key to the efficiency of WZ coders.

In addition to prediction error analysis, we also propose a very different approach, i.e., the of perceptual-based decoder-side skip mode strategy, in this chapter. The proposed technique is inspired from the observation that SI framespredicted using motion-projection algorithms often contain image areas with large prediction errors (in MSE sense) but small visual distortions. One such example is a video sequence of a low-motion scene taken by a shaky camera. If we remove every other frame and interpolate the missing frames with motion-projection algorithms, the resulting video may become smoother without major visual distortions except at image boundaries. However, to reconstruct the original shaky video from the interpolated frames using channel codes requires significant amount of parity bits, which is not worthwhile from perceptual rate-distortion perspective. In short, the proposed technique performs perceptual-based analysis to determine whether the SI regions have noticeable visual distortions because our approach allocates channel codes only for these regions.

This chapter is organized as three parts as follows:

In section 2.1, the proposed new coding strategy with adaptive source grouping is introduced. Some discussions on the error patterns of the side information are presented. The key observation highlighted in this section is that SI prediction errors are highly dependent on true motion and should not be modeled by an i.i.d. process. Therefore, regrouping of data with similar error levels into the same coding block can

significantly improve the coding efficiency. Based on this observation, the proposed DVC framework is presented in section 2.1.2. Both the SI generator and the WZ coding method are described in detail in section 2.1.2. Although the SI generator is based on the algorithm in [13], it is not straightforward to match the SI quality demonstrated in [13]. Hence, we present our implementation in detail in section 2.1.2 so that others can repeat our result. In section 2.1.2.3, the major contribution is presented. Namely, macroblock grouping for WZ coding. Experimental results show that the proposed framework achieves as much as 3% ~ 10% bit saving over a re-implementation of the DISCOVER codec.

In section 2.2, the perceptual-based decoder-side skip mode strategy are proposed. Section 2.2.1 presents the rationale behind perceptual-based coding for DVC. Some examples comparing perceptual-based and SAD-based WZ coding are shown in this section to shed light on the proposed scheme. The proposed perceptual-based DVC codec is described in section 2.2.2. Experimental results are presented in section 2.2.3.

2.1 Macroblock Grouping for WZ Coding

2.1.1 Side Information Error Analysis

In DVC, side information (SI) at the decoder can be considered as a noisy version of the original video transmitted through a virtual channel from the encoder. Based on this model, the encoder computes the channel codes using the original video so that the SI errors could be corrected at the decoder side. In existing DVC frameworks, modified channel codes [15] are adopted and the parity bits are the only physical data that are transmitted to the decoder for WZ frames. Obviously, the error-correction efficiency of the parity bits translates directly into compression efficiency of the DVC codecs.

In many DVC systems, decoders use the motion compensated frame interpolation method to generate SI [13]. This method assumes that the motions of objects in video sequences are translational motions with constant velocity. Therefore, the observation (i.e. SI) of WZ frames through the virtual channel could be predicted by linear interpolation from neighboring key frames. There are three kinds of prediction errors of SI in the motion compensated frame interpolation process. The first type of error is due to compression of key frames. Since key frames are compressed with quality loss, the distortions in key frames will propagate to the SI.

The second type of error is from the uncertainty of estimated motions. Unlike traditional closed-loop coding where true motions may not be critical to achieving good compression efficiency, motion compensated frame interpolation algorithms for SI generation require estimation of true motion fields between reference (key) frames to reduce SI prediction error. The third type of error is due to the assumption that objects move in constant velocity. In case of non-translational and/or non-constant translational motion, linear interpolation would produce texture position shift which would cause large PSNR degradation in textured region. However, such position shift may not degrade visual quality since human visual system is insensitivity to consistent pixel-wise image shift.

Fig. 1 SI of 70th frame of FOREMAN (left), and its error image (right).shows an example of the SI frame generated by a motion compensated frame interpolation method. It is obvious from Fig. 1 SI of 70th frame of FOREMAN (left), and its error image (right). that the error distribution is not spatially invariant. The SI error characteristics are affected by both the texture and the motion complexities in video content. Since both texture and motion fields are not spatially invariant, it is ineffective to assume a spatially invariant i.i.d. SI error model and try to use channel code to correct these errors in a uniform manner. A possible solution is to estimate prior probability of each pixel using error statistics derived from the SI generation process and to adjust likelihood function before the decoding procedure [18].

Fig. 1 SI of 70th frame of FOREMAN (left), and its error image (right).

In existing rate-adaptive DVC schemes [1][10]-[12], the decoder continuously requests parity bits to correct SI errors in a uniform manner until the bit budget is empty. Since some pixels in SI are close to the original pixels, it is inefficient to request too many parity bits for these ‘good’ pixels because these bits are simply discarded at the decoder. Consequentially, pixels with different error levels should not be mixed in the same coding block and protected (corrected) together as a group since many parity bits are wasted. To improve the coding efficiency, macroblock grouping of data with similar (estimated) error statistics at encoder side is proposed in this paper. Macroblock grouping can be considered as an unequal or prioritized error correction scheme where SI areas with high probability of errors are protected by more parity bits.

Although, macroblock grouping of pixels on the encoder side can improve coding efficiency, it is not trivial to get error statistics of SI at the encoder side since SI is only available in the decoder while original video data is available only to the encoder.

In practice, if the encoder simply adopts the same SI generation algorithm used in the decoder, the true error model can be obtained. In fact, this is exactly what the closed-loop coding schemes, such as AVC/H.264, are doing. But this violates the key objective of DVC, namely, low complexity encoding. For applications with feedback channel, error statistics could be estimated at decoder side and send back to the encoder. The proposed DVC framework with macroblock grouping is proposed in next section

2.1.2 DVC With Macroblock Grouping

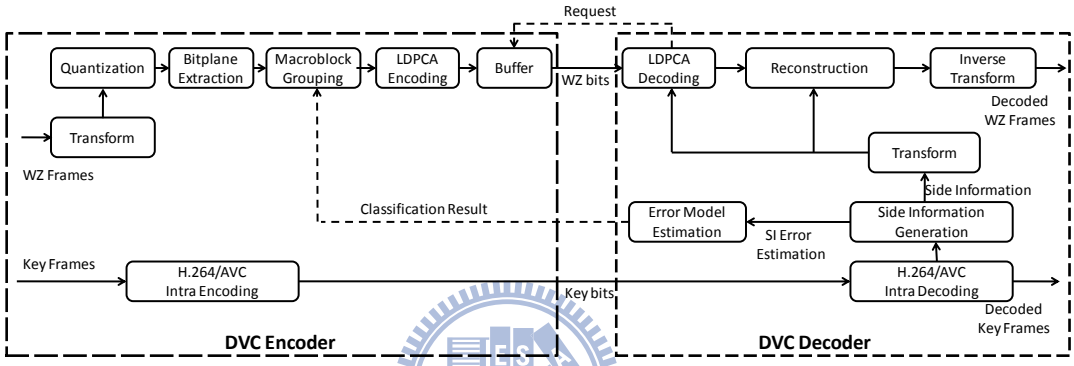


Fig. 2 Proposed DVC Architecture with Macroblock Grouping

In the proposed DVC framework, key frames are coded using an intra video codec and transmitted to the decoder. The decoder uses motion-projection based algorithm to generate SI for a WZ frame using the received key frames. SI prediction errors are estimated by the decoder and transmitted back to the encoder using an uplink channel. To reduce uplink bandwidth, error model is computed macroblock-wise. The encoder then groups original video macroblocks into different coding blocks based on their error model. The proposed coding framework has the following advantages:

1. Since video data in the same coding block has similar error characteristics, the corresponding hypothetical virtual channel noises can be more accurately modeled as i.i.d. noises. In addition, grouping allows large coding block length and improves channel code efficiency.

2. Grouping enables unequal error correction on the decoder side. When combined with proper bit allocation algorithm, one can achieve better R-D performance.

3. Based on our experiments, unequal error correction of SI prediction error also reduces visual quality variation between key frames and WZ frames. Without macroblock grouping, enforcing constant quality constraint across all coded video frames degrades R-D performance noticeably.

The detailed design of the proposed DVC framework is described in the following sections.

2.1.2.1 System Architecture

The system block diagram of the proposed DVC codec with macroblock grouping is illustrated in Fig. 2 Proposed DVC Architecture with Macroblock Grouping. The basic structure of the codec is based on the DISCOVER DVC codec [12]. At the encoder side, the video sequence is divided into odd frames (key frames) and even frames (WZ frames). Key frames would be encoded by H.264/AVC main profile intra encoder. WZ frame would be encoded by transform domain WZ codec. The encoder will receive macroblock error model information from the decoder. Before WZ encoding, it will reorder and group macroblocks according to this information. Then, each group of macroblocks form a basic Low-Density Parity-Check Accumulate (LDPCA) coding block and parity bits are encoded into the bitstream. At the decoder side, for every two received key frames, a motion-compensated frame interpolation procedure is used to generate the SI of the WZ frames. The decoder will classify macroblocks into several groups based on an error model estimator. The group index (of the error model) will be sent back to the encoder through an uplink channel. Before LDPCA decoding, the decoder will group macroblocks of SI in the same manner.

Detail of operations of different modules in Fig. 2 are described as follows:

- Transform/Inverse Transform: the 4-by-4 integer block transform from AVC/H.264 is used here. This integer transform has very low complexity and is suitable for low complexity encoder implementation.

- Quantization: After integer transform, each coefficient component is quantized. The quantization matrices for different distortion levels are shown in Fig. 3. For DC coefficients, uniform quantizer with range between

0 and 1024 is adopted. For AC coefficients, double dead zone quantizer is used, and adaptive quantizer range of each band is sent to decoder on a frame-by-frame basis. Comparing traditional uniform quantizer with double dead zone quantizer, we found that the latter improves R-D performance significantly. The reason is that lots of AC coefficients are small and near zero. For these coefficients, small errors in side information prediction could change its sign. This phenomenon impacts bit-plane coding seriously because small errors may corrupt the first bit-plane (the most significant one) and influence the initial state of the following bit-plane decoding process. Since these small errors should not dominate decoding quality, double dead zone quantizer eliminates these errors and gains better R-D performance.

- Bit-plane Extraction: After quantization, frequency components in each band are separated into bit-planes. For example, in WZ distortion level 2 (QM=2), a DC coefficient is quantized to 32 level represented by a 5-bit number. Therefore, DC coefficients are separated into 5 bit-plane. AC coefficients are organized into bit-planes in the same manner. In QM=2, there are 11 bit-planes in total (5+3+3=11), 5 for DC coefficient and 6 for two AC bands (note that there are only two AC bands for QM = 2).

- Macroblock Grouping: Based on error model classification information form the decoder, data bits in each bit-plane are rearranged into different coding blocks. Each bit-plane is divided into 4 groups, and the size of each group varies. After grouping, each bit-plane is divided into several sub-planes with similar characteristics. The classification algorithm is described in next section.

- LDPCA Encoding: The LDPCA channel code proposed in [15] is used to encode the coding blocks. Note that at this point, each coding block contains bit-plane data bits from macroblocks with similar estimated error characteristics.

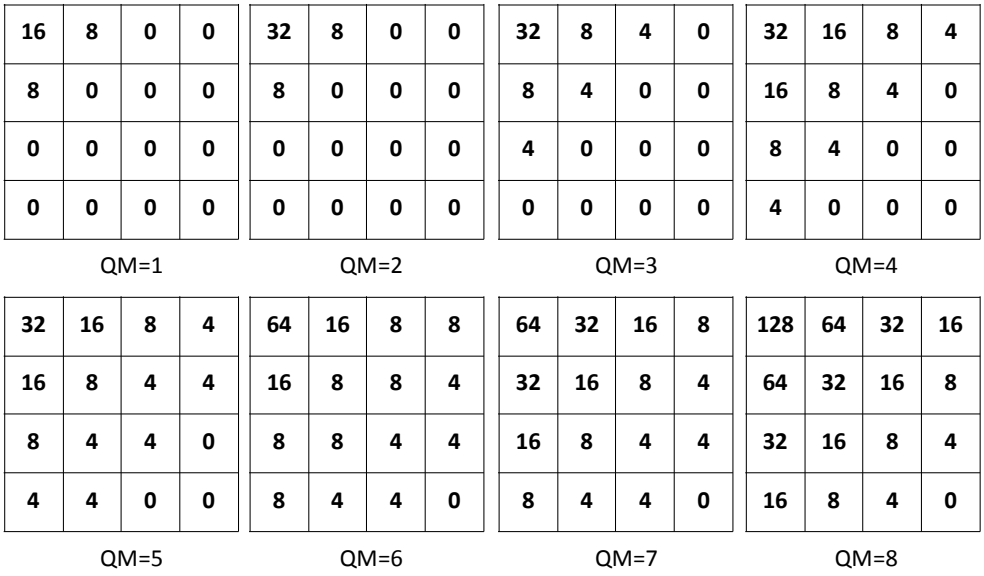


Fig. 3 Quantization Matrix for WZ Frequence Component

- Side Information Generation: After intra decoding of key frames, SI is predicted from neighboring key frames using a motion compensated frame interpolation technique [13]. The implementation details will be presented in section 3.2.

- Error Model Estimation: The SI prediction error of each 16x16 macroblock will be estimated using an estimator to be described in section 3.3. Exact error model can not be obtained because decoder does not have the original video data. This is the key reason why all existing DVC schemes perform worse than closed-loop codecs in practice. With a sophisticated error model estimator, one can improve the performance of DVC significantly.

- LDPCA Decoding: Using grouped WZ bits received from the encoder and SI generated in the decoder, LDPCA decoder can correct side information errors. The LDPCA decoder uses the belief propagation technique and the SI error statistic for each coding block is modeled as a Laplacian distribution during the LDPCA decoding process.

- Reconstruction: According to the LDPCA decoded bit-plane, the coded picture is reconstructed using MAP principle. In other words, after LDPCA decoding, if the original prediction value is inside decoded quantization range, the predicted SI pixel value remains unchanged. Otherwise, the

predicted SI value will be modified to the nearest bound of decoded quantization level.

2.1.2.2 Side Information Generation Algorithm

The side information generation algorithm adopted in the proposed system is a motion compensated frame interpolation algorithm [13]. For DVC codec, the quality of the predicted SI is one of the dominating factors in improving overall coding efficiency. The SI generation algorithm outlined in [13] is one of the best SI generation algorithms among existing DVC codecs. However, our re-implementation does not match the published SI quality in [13]. Therefore, in this subsection, we explain the details of our implementation in this section so that others can repeat our results. The detail SI generation procedure is as follows.

Step 1. Motion estimation with smoothness constraint:

The first step is to find initial motion field of WZ frame. The initial motion field is interpolated from the motion field between the neighboring key frames. The motion field resolution is one motion vector per 16×16 block. Smoothness constraint is applied during motion search between key frames in to motion vector outliers caused by the aperture problem. The cost function of motion search is

$$\bar{v}_i = \arg\min_{v}\left\{ D_i(v) + \lambda \sum_{B_j \in U(B_i)} \left(D_j(v) - D_j(\bar{v}_j) \right) \right\}, \tag{2.1.1}$$

where $\bar{v}_i$ is the estimated MV of block i, and $D_i(v)$ is the mean absolute difference (MAD) of matching block pair between key frames. The first term is common in traditional motion search as cost function. The second term is smoothness constraint. The smoothness is calculated with eight B_i neighboring blocks $U(B_i)$. The Lagrange multiplier is calculated adaptively as:

$$\lambda = \frac{1}{\beta N} \sum_{i=1}^{N} D_i\left(v_i^{+}\right). \tag{2.1.2}$$

In this paper, β is set to 10. An iterative regularization process based on this cost function is performed to solve for the optimal solution.

Step 2. Bi-directional motion refinement:

After the first step, the motion field is further refined by "bi-directional" motion estimation. The terminology (from [13]) is somewhat misleading since the process is different from the bidirectional motion estimation process in traditional closed-loop codec where the target block to be coded is used as a search template to perform motion search in both backward and forward frames (hence, bi-directional). In DVC, the target block of the original video frame is not available (at the decoder side). Therefore, what this "bi-directional" motion refinement process really does is the refinement of the motion vectors produced in step 1 (and step 3) so that the constraint that one of the matching block in the key frames must be located at a macroblock grid position is removed. The motion estimation process is still uni-directional, from one key frame to another. The refinement search is conducted around each initial motion vector. As shown in Fig. 4, a small search ranges on both ends of the initial motion vector (from one key frame to another) are select for the refinement process. The position of *A* is symmetric to that of *B*, with respect to the position of *C*. The iterative search method and cost function are the same as those described in step 1. The search range is adaptively calculated based on the motion vectors of neighboring blocks. The method used to calculate search range will be discussed later. Bi-directional motion refinement improves SI quality significantly.

Step 3. Hierarchical motion estimation:

A coarse-to-fine motion search strategy is also adopted in our implementation. At the coarsest level of motion search, initial search block size is 16×16. After the first bi-directional motion refinement, the resulting motion field would then be used to drive the second level motion estimation for each 8×8 blocks. The initial motion vectors for the second-level search is computed from the top-level motion vectors using an affine model, as illustrated in Fig. 5. As Fig. 5 shows, three neighboring MVs (shown as solid lines) are used to estimate local affine parameters, and compute the 8×8 motion vector (shown as dashed line). During this process, new search ranges will be calculated for another iteration of bi-directional motion refinement. Finally, a third-level of hierarchical motion search will be performed to obtain motion fields at the density of one vector for each 4×4 block.

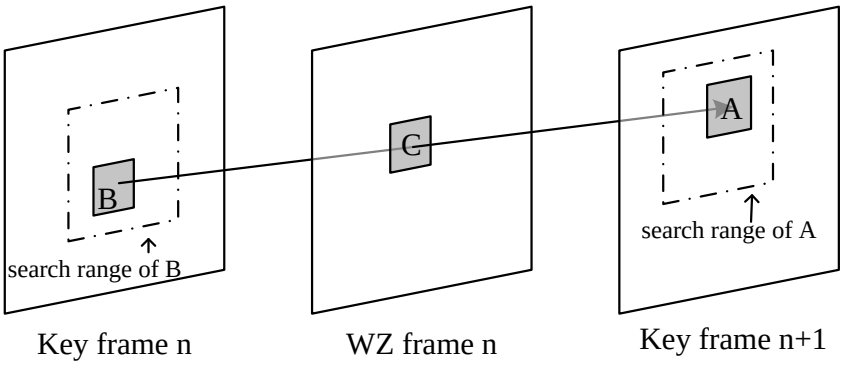


Fig. 4 Bi-directional motion refinement. *C* is the SI block to be predicted. *A* and *B* are matching blocks that are located along the translational motion path from Key frame *n* to *n*+1, respectively.

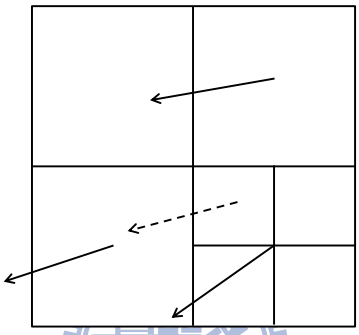


Fig. 5 Motion refinement using hierarchical search. The MV represented in dashed line is the refined MV for small blocks to be estimated from the coaser MVs represented in solid lines for large blocks.

2.1.2.3 Error Model Estimator

Since the decoder does not have the original frames, it can only estimate the SI prediction error using the texture and motion characteristics of the neighboring key frames. Since the SAD value of a matching block pair of key frames obtained during SI generation process indicates consistency of the texture alone motion path, it is a strong cue that indicates the reliability of the predicted SI block. Therefore, cost function defined in (1) is used to estimate the error levels of predicted SI blocks. There are other possible cues for estimating SI quality. For example, the number of corner points and the strength of edge pixels in an SI block indicate the complexity of the texture. The regularity of the dense motion field between neighboring key frames indicates how well the true motion matches the block-based constant-velocity translational motion model. In current implementation, only the SAD measure is used

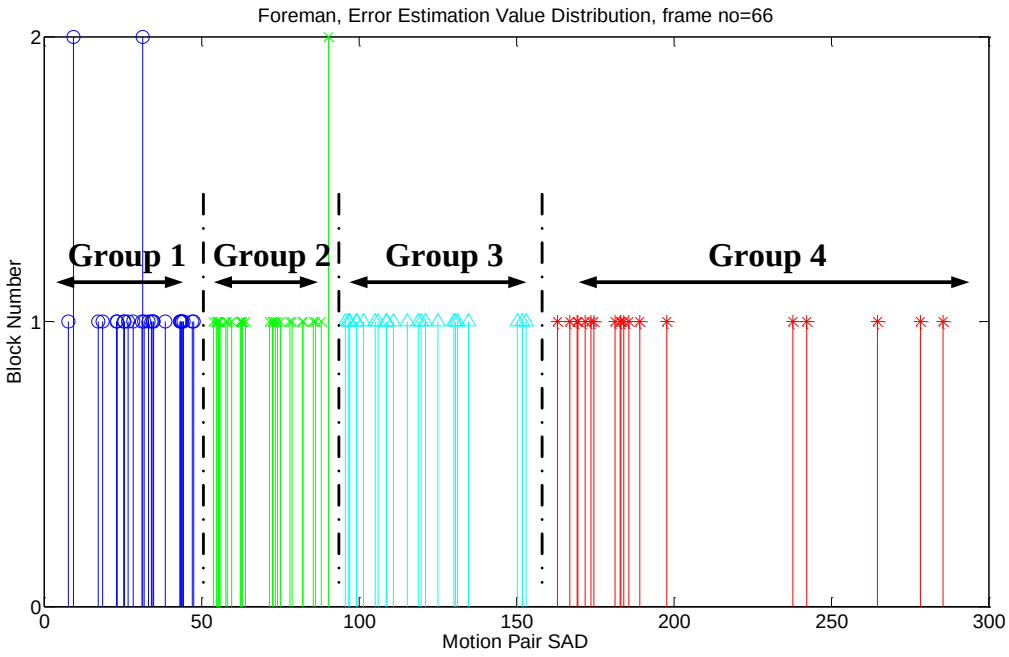
for the proposed DVC framework. Other sophisticated measures will be investigated in the future.

Based on the estimated error level, the macroblocks will be classified into several groups (4 groups per video frame are used in current implementation). Mean-Shift algorithm [45][46] is used for classification of macroblocks. The classification process is described as follows:

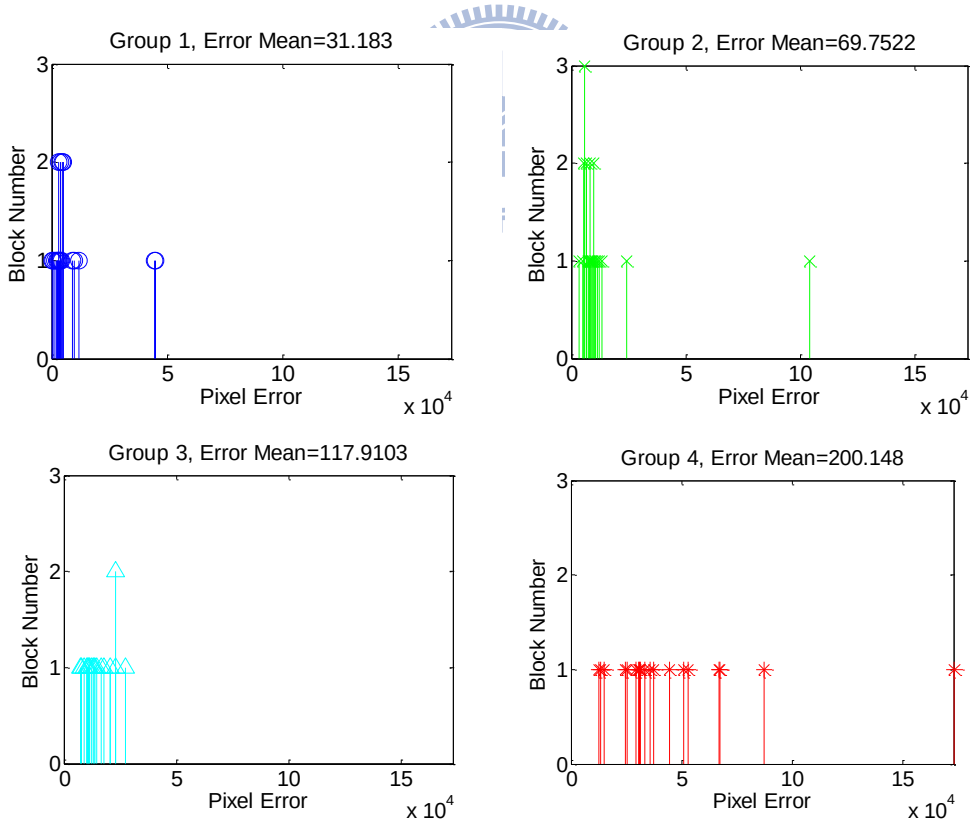
Step 1. Initially sort macroblocks evenly into 4 groups according to their error levels. Calculate mean of error level in each group.

Step 2. Re-group the macroblocks so that each macroblock is classified into the group with nearest error level mean.

Step 3. If the classification process converges (i.e. no macroblock is moved to a different group), terminate the process, otherwise, go back to step 2.



(a) Example of estimated error distribution.



(b) True error distributions.

Fig. 6 Side informaton error histograms.

Fig. 7 Error classification result.

Fig. 6 and Fig. 7 show the classification results of 66th frame of the Foreman sequence. Fig. 6(a) is the histogram of estimated error levels for all 16×16 macroblocks and the classification of errors into four groups. In current implementation, error model estimator use the motion search cost defined in (1) as the error level of a macroblock in the SI. In the future, more sophisticated error model estimator will be used to classify macroblocks based on error levels as well as error types. Fig. 6(b) shows true error histograms of each classified groups, where the original video frame is used to compute the true error levels of the SI's. Fig. 7 shows the distribution of the four groups of macroblocks in the video frame. It is interesting to see that the proposed error model estimator classifies the face area, the textured background region, and the smooth background region into different groups.

After the decoder classifies macroblocks into four groups, the information (two-bit per macroblock) is entropy coded and sent back to the encoder. On the encoder side, group-wise bit allocation would be performed so that channel codes can be used more efficiently to cope with different level of errors.

2.1.3 Experimental results

This section presents several experiments to demonstrate the efficiency of the proposed framework. We have implemented the DISCOVER codec as described in [12] as a baseline codec for comparison (referred to as the DISCOVER* codec in this section). The proposed techniques are then integrated into this baseline codec. However, we have found that our baseline implementation of the DISCOVER codec

has worse performance than the original DISCOVER codec, probably due to the inferior SI generator as mentioned in section 3.2. Therefore, a binary executable of the DISCOVER codec provided by the original authors is also used in the following evaluations. Nevertheless, to make fair judgments of the performance gain from the proposed macroblock grouping WZ coding tool, one should still compare the proposed DVC codec against the DISCOVER* codec. In addition, the AVC/H.264 intra coder and AVC/H.264 zero-motion inter coders are used for R-D performance comparison against all DVC codecs.

2.1.3.1 Test Conditions

The QCIF version of eight standard test sequences, Foreman, Hall Monitor, Soccer, Coastguard, Mobile, Car Phone, Table Tennis and Motion-and-Daughter, are used in the experiments. The temporal resolution of these sequences is 15Hz. For all sequences, the DVC GOP size is 2. The key frames are coded using AVC/H.264 main profile intra coder. Reference software JM [47] is used and the RDO is turned on.

2.1.3.2 WZ Coding Efficiency Evaluation

In this section, an experiment is conducted to verify the WZ coding gain from the proposed macroblock grouping scheme directly. An LDPCA channel coder is used to generate syndrome bits of the original video sequences with and without the macroblock grouping tool. The two decoders (with or without macroblock grouping) then request just enough syndrome bits to correct the SI's to produce exactly the same reconstructed frames. Take Foreman sequence for example, when the quantization level of all WZ frames is set to QM=3 and the QP of Key frames is 33, the required syndrome bit rate is 7.72 kbps without the proposed macroblock grouping tool. However, after macroblock classification and grouping, the required syndrome bit rate is reduced to 6.56kpbs. The bit rate saving is 15%. The experiment is conducted over all test sequences and over all combinations of encoder settings, Key QP=25~40 and WZ QM=1~8 (i.e. 128 WZ rate settings in total). TABLE. I summarizes the maximum, minimum, and mean of bit rate change of the proposed coding tool over these 128 decoding rates (note that a negative percentage means bit rate reduction).

TABLE. I Bit Rate Reduction with Proposed Macroblock Grouping

Sequence	Maximum	Minimum	Mean
Foreman	-5.10%	0.47%	-2.68%
Hall	-22.33%	-0.82%	-9.73%
Coastguard	-12.64%	0.86%	-3.21%
Soccer	-15.38	0.8%	-4.85%
Carphone	-13.43%	-0.17%	-4.92%
Mobile	-5.66%	1.48%	-1.24%
Table Tennis	-20.59%	-1.85%	-8.24%
Mother-and-Daughter	-8.97%	-0.39%	-4.05%

As the results shown, WZ coding with macroblock grouping can improve coding efficiency about 5% on average. The amount of bit rate reduction depends on how well the error estimator classifies the SI error levels of the corresponding video sequence. With more sophisticated error model estimator, the performance of the proposed techniques can be further improved.

2.2 Perceptual Based Distributed Video Coding

In this section, we propose a very different approach of perceptual-based decoder-side skip mode strategy. The proposed technique comes from the key observation that SI frames predicted using motion-projection algorithms often contain image areas with large prediction errors (in MSE sense) but small visual distortions. the proposed technique performs perceptual-based analysis to determine whether the SI regions have noticeable visual distortions. Our approach allocates channel codes only for these regions.

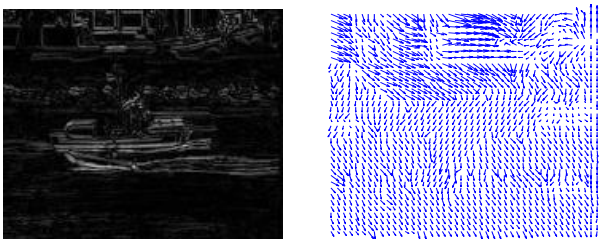
2.2.1 Side Information Perceptual Analysis

Fig. 8 shows the SI prediction frame generated by motion-projection using two neighboring key frames. The SI in this example is particularly interesting because the PSNR differences between the key frames (31.2 dB and 29.9 dB) and the in-between SI frame (17.5 dB) are more than 10 dB, but visually, video quality across the key

frames and the SI frame are still consistent. Therefore, if perceptual quality is the coding goal, there is no need to request parity bits to correct the large amount of errors in this SI frame. Note that the main reason that the PSNR of the SI is low is because the scene is shaky due to slight camera motion. The motion field of the SI frame predicted from the two key frames is a smooth field different from the true motion field. The interpolated SI frame using this predicted motion field is visually appealing, but is different from the original frame.

(a) Original frames 37, 38, and 39 (from left to right) of the 15 Hz Coastguard sequence.

(b) The SI frame (middle) and the two key frames (left and right) used for SI prediction.



(c) The error image of SI and the projected motion field between the SI frame and key frame 37.

Fig. 8 An example of SI with good perceptual quality but low PSNR. The PSNR of the SI frame is only 17.5 dB, but its visual quality is comparable to the neighboring key frames (around 30 dB).

Another issue with WZ frame reconstruction is that the error distribution of SI prediction is spatially varying [48].

Fig. 9 shows one such example.

Fig. 9(a) is a predicted SI frame of the 48th frame of the Foreman sequence (QCIF@15Hz). The SI error characteristics are affected by both the texture and motion complexities in video contents.

Fig. 9(b) shows the reconstructed WZ frame using LDPCA code (the amount of parity bits are 12.38 kbits). The PSNR is 31.22 dB. There are spatially varying burst errors in the SI frame and it is ineffective to assume a spatially invariant i.i.d. SI error model and try to use channel codes to correct these errors. Burst errors usually happen at moving edge boundaries. However, existing DVC techniques [1][10]-[12] group consecutive macroblocks in scan line order into a coding block without taking into account the texture and motion characteristics of these macroblocks.

If we only correct the facial area (using 10.46 kbits), the partially reconstructed WZ frame looks visually more appealing, as shown in

Fig. 9(d). Note that the sharp straight lines of the background building in the SI frame have uniform pixel-shift errors that cause a low PSNR value (28.62 dB). However, visually, there are no perceptible errors. On the other hand, the image in

Fig. 9(b) uses part of the bit budget to correct the sharp edges towards the correct pixel position to certain degree. Unfortunately, such halfway correction produces fuzzy edges and degrades visual quality. In short, if a decoder can determine the regions of interest (ROI) automatically, and applies WZ reconstruction only in the ROI, we can achieve better visual quality at lower WZ rates. In addition, more bit budget can be allocated to key frames to further improve overall R-D performance [49]. In this paper, we define ROI as the areas in the SI frame where distortions are perceptually salient.

(a) SI frame (b) Full correction (c) Error of (b)

(d) Partial correction (e) Error of (d)

Fig. 9 Perceptual-based reconstruction of SI prediction errors. The reconstructed WZ frame in (b) tries to correct the prediction errors of the whole SI frame using 12.38 kbits of LDPCA codes. The reconstructed WZ frame in (d) only corrects a rectangular area that contains the face of Foreman using 10.46 kbits of LDPCA codes.

It is important to point out that the main strength of the proposed scheme is not just to deal with the extreme cases illustrated in Fig. 8 or

Fig. 9 Since all video frames are captured with noises, as a result, the predicted SI frame usually contains noises inherited (motion-compensated) from the key frames that are different from the noises in the original WZ frames. With the proposed approach, we will not waste syndrome bits on the correction of one set of sample noises to another set of sample noises, unless they are visually significant.

2.2.2 The proposed DVC framework

Fig. 10 is the block diagram of the proposed DVC codec. We have added perceptual-based coding tools to a transform-domain DVC framework. The baseline implementation of the DVC codec is similar to the DISCOVER codec [12], plus prioritized macroblock grouping [50]. An AVC/H.264 intra coder is used to encode key frames and an LDPCA code [16] is used to correct SI frames. For each SI macroblock, the decoder performs perceptual distortion analysis and discriminates whether the macroblock belongs to the ROI. The encoder receives the ROI information from the decoder via a feedback channel and groups ROI macroblocks into the same coding block. Since the coding block size varies from frame to frame, the LDPCA module must handle variable block-length (VBL) coding. The rest of this section describes the proposed perceptual-based error analysis.

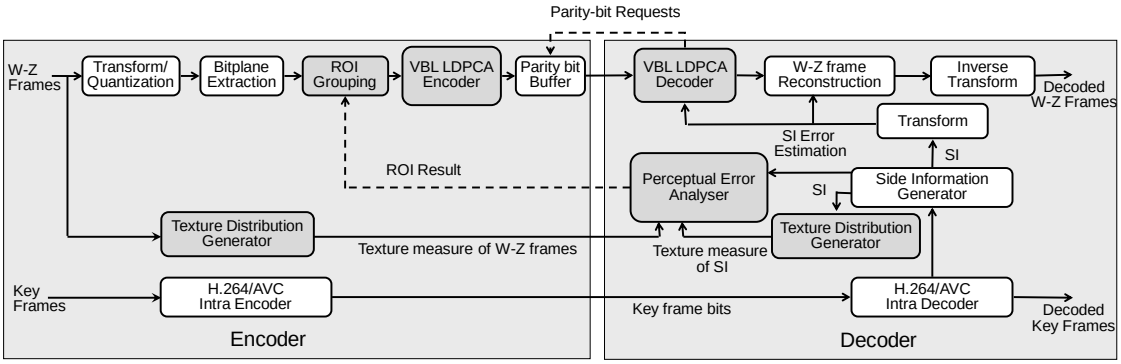


Fig. 10 The proposed DVC architecture. Gray blocks are proposed modules.

2.2.2.1 Texture Distribution Similarity (TDS) Analysis

Based on our empirical investigations, bursty SI prediction errors often happen at the boundaries of texture-rich moving objects. Therefore, the first step in the proposed perceptual-based error analysis is to identify whether the distribution of texture-rich macroblocks in the SI frame is the same as that in the quantized original frame. If an SI macroblock is texture-rich while the corresponding quantized original macroblock is not (or vice versa), the SI macroblock should be corrected by parity bits. Since the original video frames are available only to the encoder, the encoder must compute a texture distribution map of the quantized original frame and transmit it to the decoder for analysis.

The texture distribution generator first determines edge pixels in a frame using the Sobel edge operator and a threshold θedge. A macroblock is considered a texture-rich block if the percentage of edge pixels in the block is larger than a threshold θtexture. Finally, the distribution of texture-rich blocks is recorded using a bit map, one bit per macroblock. In the bit map, a '1' signals a texture-rich macroblock while a '0' signals a regular macroblock.

The selection of the two thresholds, θedge and θtexture, are described as follows. The thresholds are adaptive to the video contents. To determine the distribution of texture-rich blocks of a WZ frame at t, we compute θedge and θtexture of a frame at time t as in Eq. (2.2.1):

$$\theta_{edge}(t) = \frac{1}{M}\sum_{\mathbf{p}=1}^{M} s_t(\mathbf{p}),$$

$$\theta_{texture}(t) = \frac{100}{M}\sum_{\mathbf{p}=1}^{M} P_{edge,t}(\mathbf{p}) \tag{2.2.1}$$

where $s_t(\mathbf{p})$ is the sum of magnitudes of horizontal and vertical Sobel edge strength of pixel **p** of the WZ frame at time t, and M is the number of pixels in a key frame. $Pedge,t(\mathbf{p})$ is a binary function of edge map, i.e. $Pedge,t(\mathbf{p})$ is 1 if **p** is an edge pixel and $Pedge,t(\mathbf{p})$ is 0 otherwise. While the encoder generates the texture distribution map of the WZ frame, the decoder uses the same algorithm to generate the texture distribution map of the SI. The decoder can then compare the received texture distribution map with the texture distribution map of SI. The set of SI macroblocks that have the same texture property as the corresponding original macroblocks is denoted by Ω_{TDS}.

2.2.2.2 Motion Consistency (MC) Analysis

Motion behavior is a useful cue for estimating SI prediction quality. For example, the optical flow field between neighboring key frames is a good indication of how well the true motion field matches the block-based motion model [25]. If the motion field is irregular in a highly textured area, the visual distortion of the predicted SI may be large. For the SI frame at time *t*, we first calculate the motion smoothness $\delta_t(\mathbf{k})$ measure of a 4×4 block at block position **k** using Eq.(2.2.2)

$$\delta_t(\mathbf{k}) = \frac{1}{8}\sum_{\mathbf{j}\in\Omega_N(\mathbf{k})}\left\|\mathbf{v}_f(\mathbf{k}) - \mathbf{v}_f(\mathbf{j})\right\|_2, \tag{2.2.2}$$

where $\Omega_N(\mathbf{k})$ is the set of eight direct neighbors of block **k**, and $\mathbf{v}_f(\mathbf{k})$ is the estimated forward motion vector of block **k** for the SI frame at time *t*. The motion consistency measure $\Delta_t(i)$ of macroblock *i* is defined using Eq.(2.2.3):

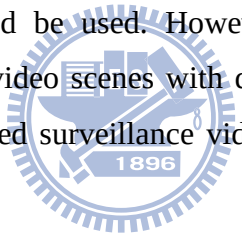
$$\Delta_t(i) = \underset{\mathbf{k}\in\Omega_B(i)}{Max}\left(\delta_t(\mathbf{k})\right), \tag{2.2.3}$$

where $\Omega_B(i)$ is the set of sixteen 4×4 blocks of macroblock *i*. Macroblocks whose motion consistency measures are smaller than a threshold θ_{MC} belong to the set Ω_{MC} of macroblocks with consistent motion. The threshold is adaptively calculated using Eq.(2.2.4):

$$\theta_{MC}(t) = \mu_{\Delta_t} + \sigma_{\Delta_t}, \tag{2.2.4}$$

where μ_{Δ_t} and σ_{Δ_t} are the mean and standard deviation of $\Delta_t(\cdot)$ of SI frame t, respectively. Note that, the motion vector estimates at low-texture areas are unreliable. Thus, we set θ_{MC} to infinity for macroblocks with no textures so that such macroblocks are always counted as motion-consistent macroblocks. We use the average Sobel edge strengths of a macroblock to determine its texture level. If the total edge strength of a macroblock is larger than a threshold θ_{TL}, it is treated as a texture-rich macroblock. The threshold θ_{TL} is set to 50 in this paper and it is not a sensitive parameter (any values from 50 to 100 produces similar results for all the test sequences).

Statistically speaking, the policy for selecting θ_{MC} will include a fixed percentile of the SI macroblocks into the set Ω_{MC}. In theory, for the detection of macroblocks with irregular motions, a sequence dependent fixed-value threshold, instead of a fixed-percentile threshold, should be used. However, our experiments show that Eq.(2.2.4) works quite well for video scenes with distinctive regions of interest, for example, for sensor network-based surveillance videos or head-and-shoulder videos for mobile social networks, etc.



2.2.2.3 Texture Structure Consistency (TSC) Analysis

For motion-projection algorithms, the texture structure consistency between the matching blocks in key frames is also an indication of visual quality level of the corresponding SI macroblock. Higher structure consistency could imply better visual quality, even if the true error is high due to uniform shifting of object pixels. For the SI frame at time t, we calculate the correlation coefficient between the edge strength of the forward and backward motion compensated predictor images (i.e., the two hypotheses of SI) using Eq.(2.2.5):

$$\rho_t(i) = \frac{\frac{1}{256}\sum_{\mathbf{p}\in\Omega_{MB(i)}}\left[\left(s_{f,t}(\mathbf{p}) - \mu_{f,t}(i)\right)\times\left(s_{b,t}(\mathbf{p}) - \mu_{b,t}(i)\right)\right]}{\sqrt{\sigma_{f,t}^2(i)\times\sigma_{b,t}^2(i)}}. \tag{2.2.5}$$

Note that $s_{f,t}(\cdot)$ and $s_{b,t}(\cdot)$ are the edge strength images of the forward and backward

hypotheses at time t, respectively. The edge strength is computed by the sum of the magnitudes of horizontal and vertical Sobel edge strength at each pixel position. $\Omega_{MB(i)}$ is the set of pixels of macroblock i. $\mu_{f,t}(i)$ and $\sigma^2_{f,t}(i)$ are the mean and variance of $s_{f,t}(\mathbf{p})$, $\mathbf{p}\in\Omega_{MB(i)}$, and $\mu_{b,t}(i)$ and $\sigma^2_{b,t}(i)$ are the mean and variance of $s_{b,t}(\mathbf{p})$, $\mathbf{p}\in\Omega_{MB(i)}$.

SI macroblocks whose $\rho_t(i)$'s are larger than a threshold θ_{TSC} belong to the set Ω_{TSC} of macroblocks with high texture structure consistency. The threshold θ_{TSC} is adaptively calculated by Eq.(2.2.6) :

$$\theta_{TSC}(t) = \mu_{\rho_t} - \sigma_{\rho_t}, \tag{2.2.6}$$

where μ_{ρ_t} and σ_{ρ_t} are mean and standard deviation of the structure consistency measure, respectively. Note that if a pair of macroblocks do not contain any structures, comparing their structure correlation is meaningless. Therefore, macroblocks whose average edge strengths are below half of the average edge strength of current frame would be directly included into the set Ω_{TSC}.

Since the formulation of TSC is similar to SSIM [51], it might be possible to use SSIM to replace TSC and achieve similar effects. However, there are two key differences between TSC and SSIM. First, TSC is computed using the edge images, not the original pixels (as in SSIM). We have observed that most visual errors in reconstructed SI frames happen around edge pixels. In other words, TSC is a variant of SSIM that is fine-tuned to capture "texture similarity" around edge pixels (which makes the threshold θ_{TSC} less sensitive to lighting differences between key frames). The second key difference is about computational complexity. For SSIM, the computed variance, covariance, and mean images (a total of five images per key frame) are filtered by an 11×11 Gaussian filter. The complexity is quite high for our purposes. For TSC, we use only two 3×3 Sobel filters (horizontal and vertical) per key frame to compute structure correlation.

2.2.2.4 Valid Motion Projection (VMP) Analysis

Motion-projection algorithms use neighboring key frames to predict SI frames. For boundary macroblocks, the projected motion vectors are often extrapolated from outside the frame boundaries, which can cause large visual distortions in SI frames. Therefore, we use the error $\varepsilon_t(\mathbf{p})$ between the matching key frame pixels that are

projected to pixel **p** of SI at time *t* to determine whether a boundary macroblock *i* of SI have large errors,

$$\varepsilon_t(\mathbf{p}) = \left| I_{t-1}(\mathbf{p} - \mathbf{v}_f(\mathbf{p})) - I_{t+1}(\mathbf{p} - \mathbf{v}_b(\mathbf{p})) \right|, \tag{2.2.7}$$

where $I_t(\cdot)$ is the image function at time *t*, $\mathbf{v}_f(\mathbf{p})$ and $\mathbf{v}_b(\mathbf{p})$ are the matching forward and backward motion vectors from the key frames to the SI pixel **p**, respectively. A threshold, θ_ε, is calculated as:

$$\theta_\varepsilon(t) = \mu_{\varepsilon_t} + \sigma_{\varepsilon_t}, \tag{2.2.8}$$

where μ_{ε_t} and σ_{ε_t} are the mean and standard deviation of $\varepsilon_t(\cdot)$ of all SI pixels at time *t*, respectively. In other words, an invalid motion projection pixel is defined as a pixel **p** whose motion vectors projected from key frame falls outside the frame boundary, and its error measure $\varepsilon_t(\mathbf{p})$ is larger than θ_ε. The set Ω_{VMP} of valid motion projection is defined as all SI macroblocks that contain less than θ_{VMP} = 10% invalid motion projection pixels. Note that, most interior macroblocks belong to the set Ω_{VMP}, regardless of the magnitude of their error measure $\varepsilon_t(\cdot)$. The selection of θ_{VMP} is based on empirical analysis. We have computed the percentage of invalid motion projection pixels of all SI macroblock of the test sequences. Setting θ_{VMP} to 10% is a strict threshold that eliminates all visual errors due to wrong boundary motion projection in all the test sequences. In fact, any value of θ_{VMP} below 15% should work fine for all the test sequences. However, as the threshold gets smaller, the coding efficiency may drop accordingly.

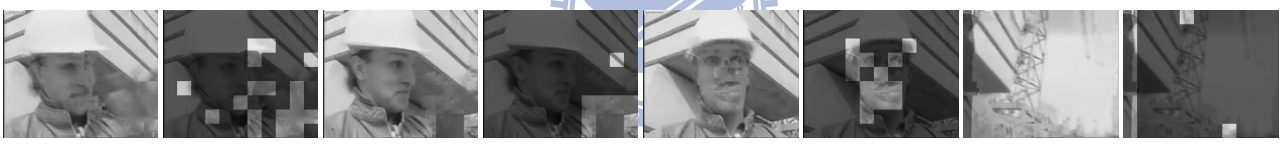
2.2.2.5 Determination of Regions of Interest (ROI)

The set ΩROI of macroblocks is composed of the macroblocks that have noticeable visual distortions. If a macroblock belongs to the intersection of the four sets ΩTDS, ΩMC, ΩTSC, and ΩVMP, we can consider this block as a macroblock that has little visual distortion and it does not require WZ reconstruction. Therefore, the initial set Ω′ROI of visually distorted macroblocks is defined as in Eq.(2.2.9):

$$\Omega'_{ROI} = (\Omega_{TDS} \cap \Omega_{MC} \cap \Omega_{TSC} \cap \Omega_{VMP})^C, \tag{2.2.9}$$

where the superscript C denotes the complementary set.

Fig. 11 and Fig. 12 illustrate examples of the complementary sets of several frames in the Foreman sequence. Fig. 11(a)~(d) show Ω_{TDS}^{C}, Ω_{MC}^{C}, Ω_{TSC}^{C}, and Ω_{VMP}^{C} for different frames of Foreman where each metric capture unique visually distorted blocks. Fig. 12 shows how these metrics complement each other in a specific frame of the Foreman sequence. Note that VMP is not particularly useful in Fig. 12. It is designed to detect artifacts at boundary blocks. Thus, it is useful when there are camera-panning motions (as in Fig. 11 Examples of SI frames (left) and detected visually distorted macroblocks (right).(d)). The unions of these sets can capture almost all SI areas with noticeable visual distortions for video scenes with distinctive regions of interest used in our experiments.



(a) Ω_{TDS}^{C} **(b)** Ω_{MC}^{C} **(c)** Ω_{TSC}^{C} **(d)** Ω_{VMP}^{C}

Fig. 11 Examples of SI frames (left) and detected visually distorted macroblocks (right).

(a) SI **(b)** Ω_{TDS}^{C} **(c)** Ω_{MC}^{C} **(d)** Ω_{TSC}^{C} **(e)** Ω_{VMP}^{C}

Fig. 12 An example of the 72nd SI frame and the detected visually distorted macroblocks.

Once the initial Ω′ROI is obtained using Eq.(2.2.9), we further refine Ω′ROI by removing isolated macroblocks and filling ROI holes. If a macroblock belongs to

Ω′ROI, but none of its neighbors belongs to Ω′ROI, it is probably an isolated outlier (unless it is located at the frame boundary). On the other hand, if a macroblock is not in Ω′ROI while the majority of its neighbors are, it probably should belong to Ω′ROI too. To obtain the refined ΩROI, we first remove isolated non-boundary macroblocks in Ω′ROI and then iteratively include a non-ROI macroblock into Ω′ROI if the majority of its neighbors are in Ω′ROI. The neighborhood structure is a diamond shape area around the macroblock under consideration as shown in Fig. 13 Neighborhood structure of the ROI refinement process. The squares are the macroblocks under consideration and the circles are their neighbors.. The iterative process continues until it converges. In the worse case, all macroblocks will be included into ΩROI, which means reliable detection of ROI is not possible and the codec falls back to full-frame WZ reconstruction. However, this situation never happens in our experiments.

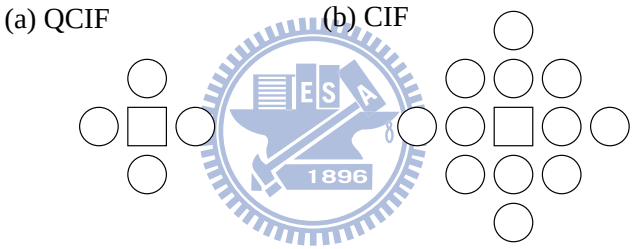


Fig. 13 Neighborhood structure of the ROI refinement process. The squares are the macroblocks under consideration and the circles are their neighbors.

Another observation is that, typically, 5~10% of macroblocks belong to the set Ω^{C}_{TDS}. However, at a scene change frame, the number of macroblocks in Ω^{C}_{TDS} would suddenly become large. Therefore, if the size of Ω^{C}_{TDS} is larger than a threshold θ_{SC} = 20% in a frame, full-frame WZ reconstruction would be used. Similar edge-based scene change detection methods have been proposed in [52][53]. Any value of θ_{SC} from 20% to 40% produces similar results in all the test sequences. If we occasionally mis-detect one of the frame as a scene change frame because the threshold is too low, there will not be any visual distortion. We simply suffer slightly on the coding gain. However, if we set the threshold too high, we may fail to detect some scene change frames and causes some visual artifacts. Thus, we set θ_{SC} to 20%. Fig. 14 shows some examples of macroblocks that belong to ROI.

Fig. 14 Examples of ROI detection results. Macroblocks with normal gray levels are in Ω_{ROI}.

2.2.2.6 Impact of large GOP sizes on the proposed perceptual metrics

We have been using the DVC coding structure with GOP size equals two due to the constraint of the SI generation algorithm used in this paper. The motion-projection SI generation algorithm assumes that the motions between two key frames are constant velocity translational motion. For most of the macroblocks, this assumption is valid when GOP size is small. However, as GOP size becomes large, most of the macroblocks will violate the constant velocity translational motion model. As a result, in addition to high SI prediction errors, there will be larger discrepancy between the predicted motion vectors and true motion vectors.

If the SI prediction error is high, the TDS metrics will include a large portion of macroblocks into the ROI since the original WZ and the predicted SI will have very different texture structure (even if the visual quality of the SI is good). Similar situation may happen to the TSC and VMP metrics because a larger portion of SI macroblocks will now be compensated from mismatched blocks due to translational motion constraint. Furthermore, because we try to use constant velocity translational motion to approximate nonlinear motions across a large period of time (i.e., large GOP), the estimated motion fields would become less regular. The proposed MC metric would also include more macroblocks into the ROI. In TABLE. II, we list the average percentage of macroblocks captured by each metric to illustrate the effect of GOP size increase on the Foreman sequence to demonstrate the impact of large GOP size on the proposed framework. The Foreman sequence has the most complex motion among four test sequences.

To solve the issue of larger GOP sizes, we have to adopt a more complex SI generation algorithm. For example, in[54], the initial SI generated using a first-order motion projection algorithm is used only for decoding of most significant bands (e.g. DC bands). The partially reconstructed WZ frame will then be used to help the second phase SI generation. Alternatively, one can use a higher-order motion-projection algorithm that takes into account more reconstructed frames (instead of simply two key frames) and tracks object trajectories for SI generation.

TABLE. II Impact of gop size on the proposed perceptual metrics.

	GOP Size	TDS^c (%)	MC^c (%)	TSC^c (%)	VMP^c (%)	ROI (%)
Foreman	2	6.3	14.0	14.0	3.4	36.0
	4	9.4	30.1	45.0	5.0	68.8
	8	15.1	36.5	50.2	6.5	77.7

2.2.2.7 Complexity analysis of the proposed perceptual DVC codec

The proposed DVC framework has to perform extra computations in both the encoder and the decoder for perceptual analysis. Nevertheless, the overall complexity of the proposed decoder is often less than the complexity of the traditional DVC codecs. Note that the most time-consuming module in a DVC codec is the channel decoder (e.g., the LDPCA decoder in this paper) and the SI generation algorithm. The proposed perceptual analysis technique allows us to perform only partial LDPCA decoding. This scheme reduces the decoder complexity significantly when the ROI is small. On the encoder side, although the complexity does increase slightly, it is negligible compared to the baseline implementation.

To quantify the computational complexity of the encoder and the decoder, we have tested the proposed DVC codec, the DISCOVER codec, the AVC intra codec, and the AVC zero-motion inter codec on an Intel Core2 3GHz CPU with 4GB RAM. The AVC codec used is JM 17.2 and the coding structure of the AVC zero-motion codec has GOP size 2 with a B frame between two I frames. The video sequence used is the FOREMAN sequence at 15 frames per second (a total of 149 frames). TABLE.

II shows the encoding time comparison while TABLE. III shows the decoding time comparison.

As one can see from TABLE. III and TABLE. IV, although the complexity of the proposed encoder is slightly higher than the complexity of the DISCOVER encoder (about 0.76% higher on average), the decoder complexity of the proposed codec is less than that of the DISCOVER codec (about 30.0% lower on average). The breakdown numbers of the execution time per frame of each module of the proposed codec for the same experimental setup are shown in

TABLE. V and TABLE. VI.

TABLE. III encoding time comparison for foreman, qcif@15fps.

DISCOVER		Proposed DVC	
(QP, QM)	Encoding time (msec)	(QP, QM)	Encoding time (msec)
(34, 1)	3629	(34, 1)	3690
(34, 5)	3735	(34, 5)	3741
(34, 8)	3777	(34, 8)	3793
AVC Intra		AVC Zero-Motion	
QP	Encoding time (msec)	QP	Encoding time (msec)
34	7249	34	10236

TABLE. IV decoding time comparison for foreman, qcif@15fps.

DISCOVER		Proposed DVC	
(QP, QM)	Decoding time (msec)	(QP, QM)	Decoding time (msec)
(34, 1)	1.0×10^7	(34, 1)	0.9×10^7
(34, 5)	1.6×10^7	(34, 5)	1.1×10^7
(34, 8)	3.4×10^7	(34, 8)	1.8×10^7
AVC Intra		AVC Zero-Motion	
QP	Decoding time (msec)	QP	Decoding time (msec)
34	227	34	204

TABLE. V breakdown of encoding time per frame (in msec) for the propsoed codec.

QM	Intra encoding	LDPCA encoding	Computing Sobel edge	Computing texture distribution map
1	47	1.3	0.77	5.0×10^{-2}
5		1.9		
8		2.6		

TABLE. VI breakdown of decoding time per frame (in msec) for the propsoed codec.

QM	Intra decoding	SI gen.	TDS analysis	MC analysis	TSC analysis	VMP analysis	ROI calc.	LDPCA decoding
1	1.5	1.1×10^{5}	0.67	0.25	1.4	0.23	1.4×10^{-2}	1.2×10^{4}
5								3.9×10^{4}
8								1.4×10^{5}

Although the overall complexity is usually lower for the proposed approach, its theoretical coding delay is indeed longer than that of the traditional DVC approaches. When the proposed encoder receives an original WZ frame, it has to wait until the decoder provides the ROI map before it can start LDPCA encoding. This delay is composed of two parts: the computation time of the ROI map and the transmission time of the map back to the encoder. The uncompressed ROI information is one bit per macroblock. For CIF@15fps, the time interval between two video frames is about 66 milliseconds. With a feedback channel bandwidth of 20 kbps, it would take 19.8 milliseconds to transmit 396 bits per frame back to the encoder. Such feedback bandwidth is not difficult for today's wireless access technology.

As for the coding delay caused by the computation of the ROI map, it includes the SI generation time plus the proposed perceptual analysis time. From

TABLE. V, it is obvious that the SI generation time requires hardware acceleration in the decoder in order to fulfill real-time requirement. However, since

the SI generation algorithm is very similar to the motion estimation algorithm of traditional video encoders, there are many hardware solutions available.

2.2.3 Experimental Results

This section presents experiments to demonstrate the efficiency of the proposed framework. The experiments are composed of two parts. The first part compares the R-D performance of the proposed framework with the DISCOVER codec [12], AVC/H.264 intra codec, and AVC/H.264 zero-motion inter codec. The GOP size of the DVC codecs and the AVC/H.264 zero-motion inter codec is two. The coding structure of the AVC/H.264 zero-motion inter codec is IBI. The DISCOVER codec is obtained from the DISCOVER website. The second part of experiments provides video snapshots of consecutive frames for visual quality evaluations. Visual results of the proposed DVC codec and the DISCOVER codec are shown side-by-side for comparisons. The 15 Hz, QCIF version of four standard test sequences, (Foreman, Hall Monitor, Coastguard, and Car Phone), and the 30 Hz, CIF version of four standard test sequences, (Foreman, Hall Monitor, Coastguard, and News), are used in the experiments. The key frames are coded using an AVC/H.264 main profile intra coder (JM 17.2). Note that the proposed codec requires transmission of texture distribution bitmap and ROI bitmap in addition to the WZ bits. Both maps are represented using an uncompressed bitmap of 1 bit per macroblock (i.e. 99 bits per map for QCIF images). Although we can use Huffman codes to compress the maps by 68% on average, we do not think it is critical to do so for the proposed framework. All R-D curves of the proposed codec in this section include the data rates required for transmission of the extra information.

2.2.3.1 R-D Performance Evaluation

Fig. 15 and Fig. 16 show the R-D performance of different codecs using both PSNR metric and the perceptual metric SSIM [51]. For DVC codecs, each rate point assumes a constant key frame QP and a constant WZ frame quantization matrix QM. For the proposed codec, we have adopted the same (QP, QM) setting as in the DISCOVER codec, obtained by minimizing the PSNR variance of the sequence. The (QP, QM) settings in the experiments are listed in

TABLE. VII. For the AVC/H.264 intra codec and the AVC/H.264 zero-motion inter codec, we choose the QP parameters from 29 to 43. For QCIF version of the sequences, the proposed DVC codec has better performance in all sequences comparing to the AVC-intra and the DISCOVER codecs. When compared against the AVC zero-motion inter codec, the proposed codec has better performance for Coastguard, and worse for Carphone, Foreman, and Hall Monitor. The main reason that the proposed perceptual-based codec out-performs DISCOVER in objective evaluation is that, with the proposed perceptual coding model, only a small portion of WZ bits is required to maintain consistent visual quality across the sequence. In other words, more data bits can be allocated to key frames to improve overall PSNR in the proposed scheme. As a result, the BD PSNR gain [55] over DISCOVER is 0.71 dB on average, and the SSIM gain is 0.016 on average.

For CIF version of the sequences, the result is similar. When compared against the AVC zero-motion inter codec, the proposed codec has better performance for Coastguard, slightly worse for Foreman, but worse for News and Hall Monitor. The overall PSNR gain over DISCOVER is 0.41 dB on average, and the SSIM gain is 0.010 on average. The BD Rate and BD PSNR results are listed in TABLE. VIII.

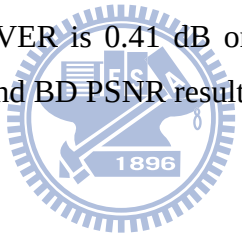


TABLE. VII Quantization Setting (QP,QM) of DVC codecs in the experiments.

QCIF Sequences							
Carphone	(40,1)	(40,2)	(39,3)	(36,4)	(36,5)	(34,6)	(32,7)
Coastguard	(38,1)	(37,2)	(37,3)	(34,4)	(33,5)	(31,6)	(30,7)
Foreman	(40,1)	(39,2)	(38,3)	(34,4)	(34,5)	(32,6)	(29,7)
Hall	(37,1)	(36,2)	(36,3)	(33,4)	(33,5)	(31,6)	(29,7)
CIF Sequences							
Coastguard	(37,1)	(36,2)	(36,3)	(34,4)	(34,5)	(33,6)	(30,7)
Foreman	(39,1)	(37,2)	(37,3)	(35,4)	(35,5)	(33,6)	(31,7)
Hall	(35,1)	(35,2)	(34,3)	(33,4)	(33,5)	(31,6)	(30,7)
News	(38,1)	(37,2)	(36,3)	(34,4)	(34,5)	(32,6)	(30,7)

TABLE. VIII BD results of the test sequences.

QCIF				CIF			
Sequence	ΔRate	ΔPSNR	ΔSSIM	Sequence	ΔRate	ΔPSNR	ΔSSIM
Carphone	-18.6%	0.93	0.021	Coastguard	-5.5%	0.23	0.014
Coastguard	-10.5%	0.47	0.019	Foreman	-7.8%	0.28	0.011
Foreman	-11.3%	0.59	0.016	Hall	-6.6%	0.31	0.005

Hall	-10.6%	0.83	0.008	News	-12.9%	0.81	0.009

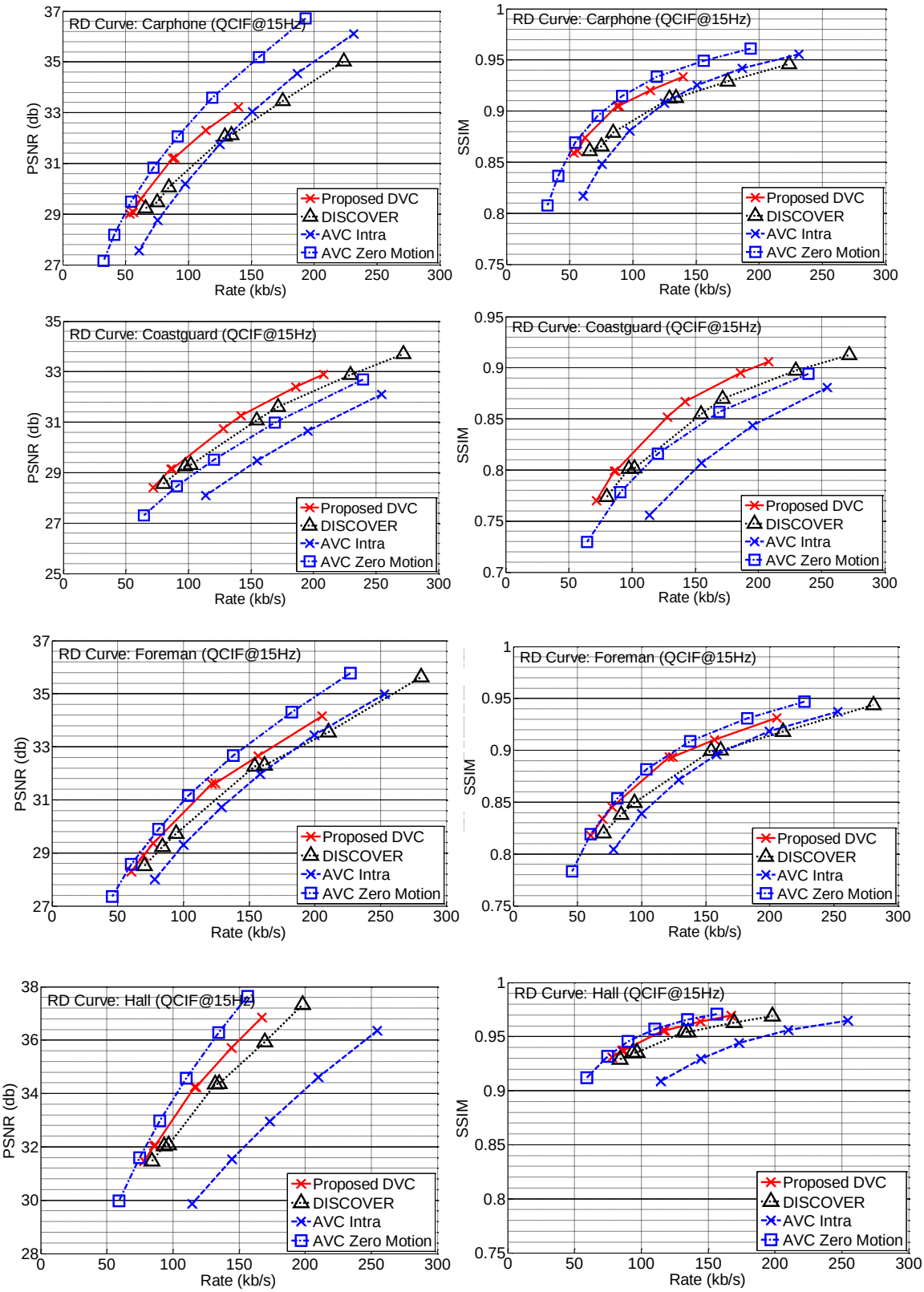


Fig. 15 QCIF sequences R-D performance comparisons using PSNR and SSIM. The average BD PSNR gain over DISCOVER is 0.71dB, and the average SSIM gain is 0.016.

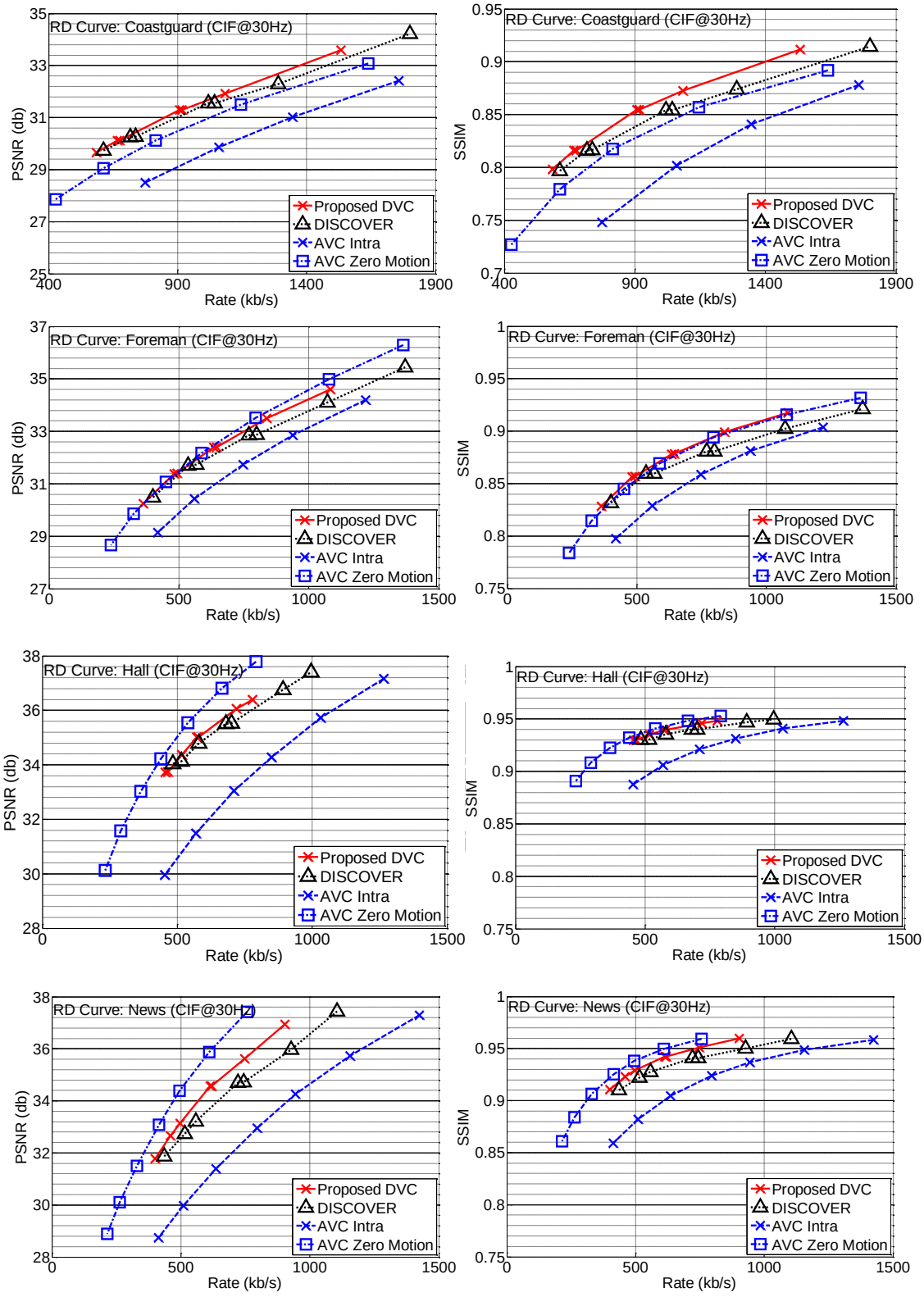


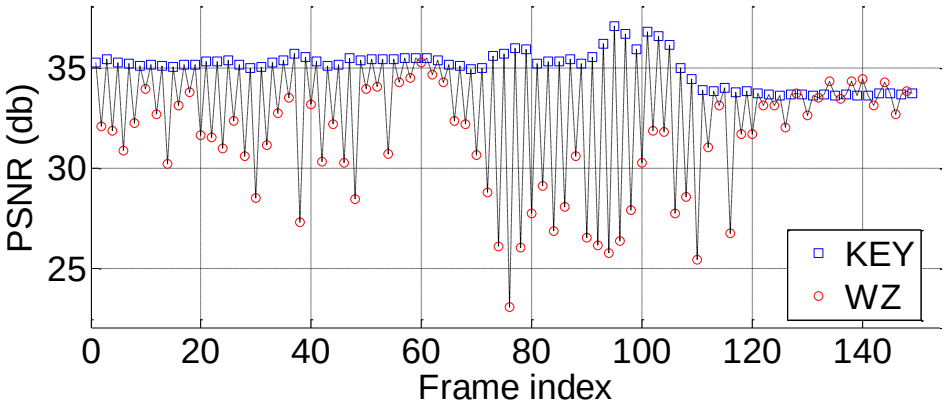
Fig. 16 R-D performance comparisons using PSNR and SSIM. The average PSNR BD gain over DISCOVER is 0.41dB, and the average SSIM gain is 0.010.

2.2.3.2 Visual Quality Comparisons

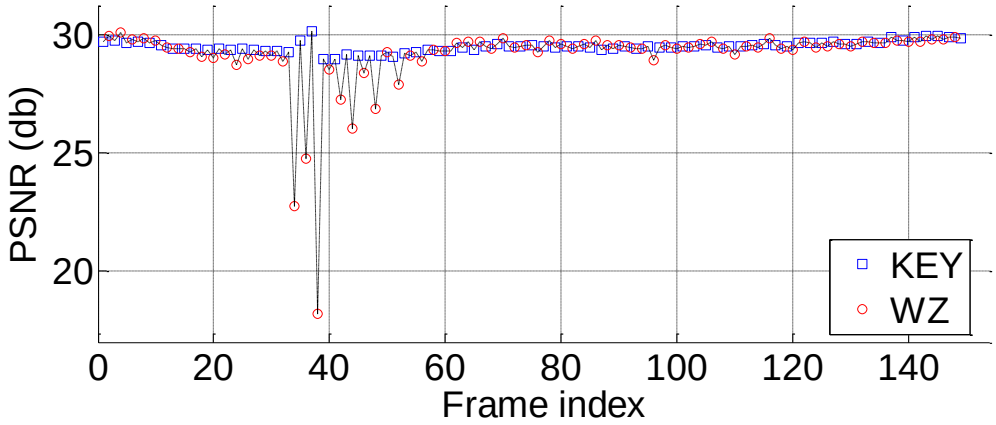
Since objective measures, such as the PSNR, do not fully reflect the visual quality of video sequences, subjective evaluation is often required for practical purposes. Therefore, in this section, we show some reconstructed frames from the proposed codec and the DISCOVER codec for visual quality evaluation. Since we have to match the target bitrate of a reconstructed sequence by the DISCOVER codec, we cannot use the same QP/QM selected using DISCOVER's algorithm. The reason is that, given same QP/QM, the encoded bit rate using the proposed codec would be much lower than that of DISCOVER's. Therefore, for each video sequence, we will use the rate ratio obtained using DISCOVER's QP/QM, and then find a set of finer QP/QM quantizers that maintains this rate ratio and produces an initial rate that is close to the target bitrate of DISCOVER's. To match the rate exactly, for each sequence, the following rate-allocation policy is applied. The bitrate of key frames are deducted from the target bitrate of DISCOVER's. The remaining bits are allocated to WZ frames. The target bit budget for each WZ frame is linearly proportional to the total sum of the errors $\varepsilon t(p)$ in its ROI (see Eq.(2.2.7)). Note that such process is not a general policy for rate control of the proposed codec. Rate control of DVC codecs is a difficult problem [56]-[59]. We simply use the aforementioned process to match DISCOVER's bitrates for visual comparisons. To evaluate visual quality, in Fig. 18 and Fig. 20, we show the snapshots of consecutive frames where the proposed codec produces largest PSNR differences between key frames and the in-between WZ frame for the Foreman and Coastguard sequences. The PSNR values across frames of Foreman and Coastguard are shown in 0. It is quite evident from Fig. 18 and Fig. 20 that PSNR, as well as SSIM [51] and FSIM [60], do not precisely reflect visual quality. For Hall Monitor and Carphone, we show the snapshots where there are noticeable visual improvements between the proposed method and the DISCOVER codec in Fig. 22 and Fig. 24. When the sequences are played back in real time, all four test sequences reconstructed by the proposed codec look sharper and have better visual quality than those reconstructed by the DISCOVER codec.

Since key frame quality has direct impact on the visual quality of the reconstructed DVC video, we also conduct another experiment where the same key frames are used for both the DISCOVER codec and the proposed codec. The key

frames are encoded using the DISCOVER codec. In addition, same amount of WZ bits are used to correct WZ frames for each method. However, in the proposed codec, WZ bits are used to reconstruct only the ROI region. Some reconstructed frames are shown in Fig. 26. In Fig. 26, snapshots are chosen at frame positions with poorest SI quality. It is clear that the visual quality of the ROI-only decoding is much better than that of the full-frame decoding method because the WZ bits are devoted to error corrections of the ROI areas where the visual errors are estimated to be large.



(a)Foreman



(b)Coastguard

Fig. 17 The PSNRs of reconstructed frames of the Foreman (160 kbps) and Coastguard (100 kbps) sequences using the proposed DVC codec.

Proposed: 35.7 dB
14.1 kbits
(SSIM=9.4, FSIM=9.6)

Proposed: 23.0 dB
2.1 kbits
(SSIM=7.4, FSIM=7.9)

Proposed: 36.0 dB
13.2 kbits
(SSIM=9.5, FSIM=9.6)

DISCOVER: 33.4 dB
10.4 kbits
(SSIM=9.2, FSIM=9.4)

DISCOVER: 29.5 dB
16.4 kbits
(SSIM=8.6, FSIM=9.0)

DISCOVER: 33.7 dB
9.4 kbits
(SSIM=9.2, FSIM=9.4)

Foreman, left-to-right: 75th (key), 76th (WZ), and 77th (key) frames

Fig. 18 Visual comparisons between the proposed codec (top row) and the DISCOVER codec (bottom row) at frame positions with highest PSNR variations. The bitrate of the proposed codec is 160.0 kbps, and the bitrate of DISCOVER is 161.6 kbps

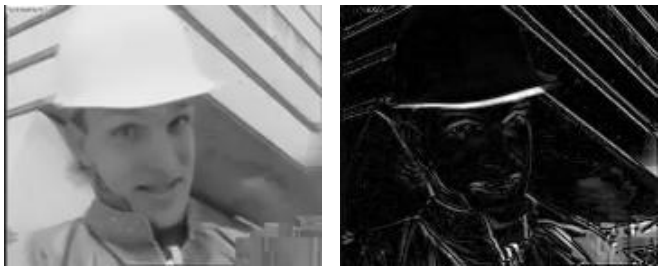


Fig. 19 The SI frame of the proposed codec (left) used in 0, and its error image (right). The SI frame of the DISCOVER codec is not available.

Proposed: 30.2 dB
9.3 kbits
(SSIM=8.4, FSIM=9.0)

Proposed: 18.2 dB
7.7 kbits
(SSIM=3.6, FSIM=7.3)

Proposed: 29.0 dB
12.7 kbits
(SSIM=8.2, FSIM=9.1)

DISCOVER: 30.1 dB
9.2 kbits
(SSIM=8.3, FSIM=9.0)

DISCOVER: 21.8 dB
12.0 kbits
(SSIM=5.4, FSIM=7.7)

DISCOVER: 28.9 dB
12.6 kbits
(SSIM=8.2, FSIM=9.1)

Coastguard, left-to-right: 37th (key), 38th (WZ), and 39th (key) frames

Fig. 20 Visual comparisons between the proposed codec (top row) and the DISCOVER codec (bottom row) at frame positions with highest PSNR variations. The bitrate of the proposed codec is 99.4 kbps, and the bitrate of the DISCOVER codec is 101.4 kbps.

Fig. 21 The SI frame of the proposed codec (left) used in 0, and its error image (right). The SI frame of the DISCOVER codec is not available.

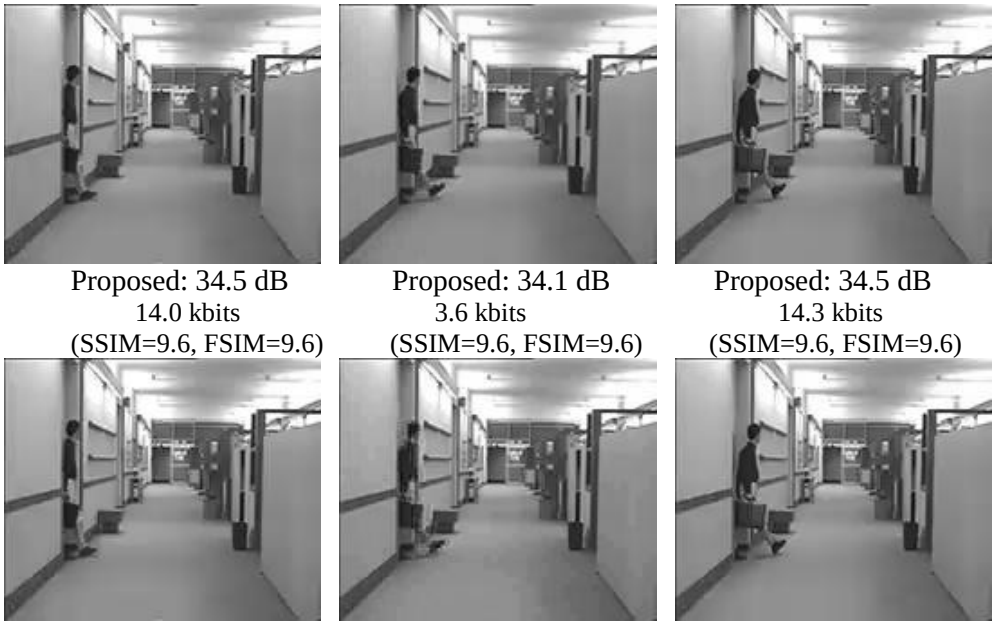


Fig. 22 Visual comparisons between the proposed codec (top row) and the DISCOVER codec (bottom row) at frame positions with noticable visual improvements. The bitrate of the proposed codec is 127.8 kbps, and the bitrate of the DISCOVER codec is 131.5 kbps.

Fig. 23 The SI frame of the proposed codec (left) used in 0, and its error image (right). The SI frame of the DISCOVER codec is not available.

Fig. 24 Visual comparisons between the proposed codec (top row) and the DISCOVER codec (bottom row) at frame positions with noticeable visual improvements. The bitrate of the proposed codec is 134.1 kbps, and the bitrate of the DISCOVER codec is 134.4 kbps.

Fig. 25 The SI frame of the proposed codec (left) used in 0, and its error image (right). The SI frame of the DISCOVER codec is not available.

Proposed: 29.0 dB 8.7 kbits (SSIM=8.9, FSIM=9.2)

Proposed: 25.9 dB 11.9 kbits (SSIM=7.9, FSIM=8.7)

Proposed: 28.5 dB 13.7 kbits (SSIM=7.6, FSIM=8.8)

DISCOVER: 30.9 dB 13.3 kbits (SSIM=8.8, FSIM=9.2)

DISCOVER: 30.8 dB 15.8 kbits (SSIM=8.6, FSIM=9.2)

DISCOVER: 30.5 dB 15.7 kbits (SSIM=8.2, FSIM=9.1)

Foreman, left-to-right: 48th (WZ), 94th (WZ), and 106th (WZ) frames

Fig. 26 Visual comparisons between the proposed codec (top row) and the DISCOVER codec (bottom row) at frame positions with poorest SI quality. The key frames for both codecs are the same. The WZ rate for the corresponding frame are the same too. The bitrate of the proposed codec is 150.4 kbps, and the bitrate of the proposed codec is 161.6 kbps.

Fig. 27 The SI frames of the proposed codec (top row) used in 0, and its error image (bottom rows).

Chapter 3 Robust Video Coding

During the stage of transmission through the error-prone environment, packet loss might occur due to signal degradation, oversaturated bandwidth, or routing issues. Moreover, the data may arrive too late to be used in real-time applications. In the case of transmission of compressed video sequences, this loss may result in a completely damaged stream at the decoder side. Error resilience (ER) and error concealment (EC) techniques are required for displaying a pleasant video signal despite the errors and for reducing distortion introduced by error propagation.

In recent years, several ER methods have been developed, such as forward error correction (FEC) [5], intra/inter coding mode selection [6] , temporal error concealment [7], and multiple description coding (MDC)[8][9]. In this chapter, we proposed three different approaches.

In section 3.1, a MRF-MCP based error resilient scheme is proposed, which employs the nearest *error-resilient frame (ER-frame)* as one of the reference frames and adopts error-resilient RDO (ER-RDO) for optimal reference block selection. The ER-frame in our approach is a frame capable of suppressing error propagation, which can be an intra-coded frame, or an inter-coded frame with high ratio of intra-coded macroblocks. Incorporating ER-RDO in our approach is for the purpose of making the choice of the *number* and *location* of the macroblocks referring to ER-frames to be decided adaptively by using rate-distortion technique. Significant performance gains in the experiments confirm that our approach has substantial improvement over competed schemes in providing error resilience using MRF-MCP. Besides, some techniques based on our error resilient scheme are further proposed to reduce the computational cost. These techniques include moving ER-RDO from motion vector to reference frame selection, skipping unnecessary reference frames, and predicting precise motion search centers.

In section 3.2, an error resilient coding based on hierarchical B pictures is proposed. In this approach, a new hierarchical coding structure which combines two conventional hierarchical coding structures is employed to reduce the distance between a lost picture and its recovering pictures. In addition, based on the new structure, an improved estimation method is also proposed to further increase the accuracy of recovering motion.

In section 3.3, a rate-distortion optimization framework for MDC systems. With the proposed framework, the encoder can dynamically adjust coding strategy according to both video contents and channel conditions.

3.1 Error-Resilient Video Coding Using Multiple Reference Frames

In this section, a MRF-MCP based error resilient scheme is proposed, which employs the nearest *error-resilient frame (ER-frame)* as one of the reference frames and adopts error-resilient RDO (ER-RDO) for optimal reference block selection. Ths section is organized as follows. In section 3.1.1, we describe JM RDO and related works in end-to-end distortion estimation. The proposed error-resilient MRF-MCP scheme and the computational time reduction techniques are presented in sections 3.1.2 and 3.1.3, respectively. The experimental results are shown in Section 3.1.4.

3.1.1 Related Works

This section presents the RDO technique used in JM [47] which is the reference software of H.264/AVC, and describes some related works about end-to-end distortion estimation.

3.1.1.1 Rate-Distortion Optimization in JM

JM provides a Lagrangian method which optimizes the tradeoff between video quality and bit rate to determine coding parameters. The Lagrangian method is applied at two stages, *motion estimation* and *mode decision*. In the stage of motion estimation, it is applied to determine the best motion vector (MV) and the reference frame; while in the stage of mode decision, it is to decide the best coding mode.

The Lagrangian formulation for motion estimation stage is written as follows:

$$J(mv) = D_{src} + \lambda_{motion} R(mv, ref) \quad (3.1.1)$$

where the source distortion, D_{src}, denotes the block-level prediction error between the current and the reference blocks. It is usually measured as the *sum of absolute*

difference (SAD); *R*(*mv*,*ref*) is the estimate of bitrate for the specified motion vector and reference frame index; and λ_{motion} is the Lagrange multiplier to control the weight of the bitrate cost.

The Lagrangian formulation for mode decision stage is written as follows:

$$J(mode) = D_{src} + \lambda_{mode} R(mode) \tag{3.1.2}$$

where the source distortion, D_{src}, denotes the macroblock-level difference between reconstructed macroblock and the original one. It is usually measured as *the sum of squared difference* (SSD); λ_{mode} is the Lagrange multiplier for mode decision; and *R*(*mode*) denotes the estimated coding rate for the specified mode (reference frame, coding mode, residue, *etc*.).

3.1.1.2 Expected End-to-End Distortion Model

Commonly, the expected end-to-end distortion is defined using sum of squared differences (SSD). That is

$$d_n^i = E\left\{\left(f_n^i - \tilde{f}_n^i\right)^2\right\} \tag{3.1.3}$$

where f_n^i and $\tilde{f}_n^i$ denote the original value and the decoder reconstructed value, respectively, for pixel *i* in frame *n*. To effectively calculate the distortion, the decoder reconstructed value $\tilde{f}_n^i$ which is unknown in the encoder needs to be derived further. The authors in [11] have derived $\tilde{f}_n^i$ in a way such that $d_n{}^i$ can be recursively calculated at the encoder. We summarize their approach here. Let $\hat{f}_n^i$ and $\hat{r}_n^i$ be the reconstructed value and the reconstructed residue in the encoder, respectively. With a motion vector *mv* predicted from reference frame *ref*, $\hat{f}_n^i$ can be represented as $\hat{f}_n^i = \hat{f}_{ref}^{i+mv} + \hat{r}_n^i$. Suppose the transmission error rate is known as *p* and frame copy is adopted as the error concealment policy. Then, the decoder reconstructed value $\tilde{f}_n^i$ can be represented as

$$\tilde{f}_n^i = \begin{cases} \tilde{f}_{ref}^{i+mv} + \hat{r}_n^i & w.p.\ \ 1-p \\ \tilde{f}_{n-1}^i & w.p.\ \ \ p \end{cases} \tag{3.1.4}$$

Hence, the expected end-to-end distortion d_n^i for inter-coded pixel i in frame n was derived as

$$d_n^i = E\left\{\left(f_n^i - \tilde{f}_n^i\right)^2\right\}$$
$$= (1-p)E\left\{\left(f_n^i - (\tilde{f}_{ref}^{i+mv} + \hat{r}_n^i)\right)^2\right\} + pE\left\{\left(f_n^i - \tilde{f}_{n-1}^i\right)^2\right\}$$
$$= (1-p)d_{s_n}^i + (1-p)d_{ep_{ref}}^{i+mv} + p(E\left\{\left(f_n^i - \hat{f}_{n-1}^i\right)^2\right\} + d_{ep_{n-1}}^i) \qquad (3.1.5)$$

where ds denotes the source distortion and dep the error- propagated distortion. The detailed derivation can be found in [66] and thus is omitted here. Since $d_{s_n}^i$, f_n^i, and $\hat{f}_{n-1}^i$ are known at the encoder, the estimation of d_n^i mainly relies on the calculation of $d_{ep_{ref}}^{i+mv}$ and $d_{ep_{n-1}}^i$. Since $d_{ep_{ref}}^{i+mv}$ and $d_{ep_{n-1}}^i$ are in a similar style, they are derived in a generalized formula as

$$d_{ep_n}^i = (1-p)d_{ep_{ref}}^{i+mv} + p(E\left\{\left(\hat{f}_n^i - \hat{f}_{n-1}^i\right)^2\right\} + d_{ep_{n-1}}^i) \qquad (3.1.6)$$

It is observed that the error-propagated distortion of current frame (*i.e.*, $d_{ep_n}^i$) can be recursively calculated by the error-propagated distortion values from previous frames. As a consequence, the expected end-to-end distortion d_n^i can be estimated at the encoder.

3.1.2 The proposed method

This section presents a novel scheme which exploits MRF-MCP taking into account channel conditions and content characteristics to achieve the goal of error resilience.

3.1.2.1 Candidate Reference Frames

In traditional MRF-MCP techniques, the current coding frame uses immediately preceding K frames as reference frames. Due to high co-relation between these frames and the current coding frames, a good coding efficiency can be achieved. However, since these frames are located closely in the sequence, they have similar

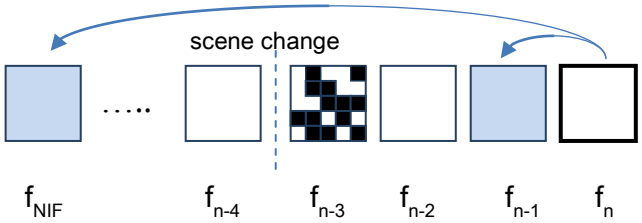
characteristics in regard to error propagation length. Predicting from any of them shall obtain similar low error-resilient capabilities. However, in some circumstances such as high packet-loss rates, obtaining better error resilience is preferred even with some loss in coding efficiency. Hence, we include high error-resilient frames as part of reference frames. The nearest intra-frame (NIF) is a good candidate because of its high error-resilience. Assume the number of reference frames is 2. In Fig. 28(a), for instance, the candidate reference frames of the current coding frame f_n will consist of f_{NIF} as well as $f_{n-1.}$

When scene changes occur, however, the difference between the two consecutive frames right before and after a scene cut becomes large. In Fig. 28(b), assume a scene change happened between frames f_{n-4} and f_{n-3}. For those frames (*e.g.*, f_n) locating after the scene cut and before the next I-frame, predicting from the NIF, f_{NIF}, may suffer from a large prediction error and dramatically decrease the coding efficiency. In this case, it would be no longer beneficial for f_n to predict from the NIF as the gain from error propagation suppression may not be able to compensate the considerable loss in coding efficiency. As a consequence, f_n will choose to predict from f_{n-1}, suffering from the error propagation from f_{NIF} to f_{n-1}. We also notice that the frame right after the scene cut (called *scene-change frame*) often has a high ratio of intra-coded blocks, which provide a certain ability to alleviate error propagation. Compared with NIF, referencing to this scene-change frame shall obtain a much better coding efficiency. Therefore, for those frames after the scene cut and before the next I-frame, we employ the scene change frame, instead of NIF, as one of the reference frames. As shown in Fig. 28(b), the candidate reference frames of f_n would become f_{n-1} and f_{n-3} if f_{n-3} is a scene-change frame. To be more general, for a frame with K reference frames, we propose that its reference frames include $(K-1)$ immediately preceding frames as well as the nearest *ER-frame*, which can be either

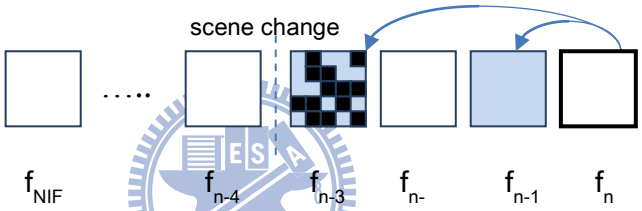
- an intra-fame (I frame), or
- an inter-frame (P or B frame) with high-ratio of intra- coded macroblocks (e.g., scene-change frames)

While encoding frame n, the encoder maintains K frames (K-1 short-term and one long-term reference frames) in buffer. The short-term reference are frames n-1, n-2, …, and (n-K+1). The long-term reference is the nearest ER-frame. When the encoder moves on to encode frame n+1, if frame n is not an ER-frame, then the

long-term reference frame remains the same and the short-term reference slide forward by one to frames *n*, *n*-1, …, *n*-*K*+2; otherwise, the long-term reference jumps forward to frame *n* and the short-term reference frames are all removed from the encoder buffer. In the latter case, when the encoder moves on to encode succeeding non-ER frames, the long-term reference remains static and the number of short-term reference frames increases one at a time till *k*-1 frames are maintained in buffer and then the *k*-1 frames slide forward by one again as described above.



(a) The nearest I-frame as the ER-frame



(b) The scene change frame as the ER-frame

Fig. 28 ER-frames as part of reference frames.

3.1.2.2 Error Resilient RDO

Even though the periodic-macroblock method in [67] and the robust-macroblock method in [68] have shown that error propagation can be suppressed by using long-term reference frames, they used predefined constant for the number of the macroblocks predicting from long-term reference frames and selected these blocks from the ones having maximum estimated distortion, without considering the bit-rate increased. Therefore, these methods are not adaptive to various network conditions and content characteristics. In this paper, we use rate-distortion optimization (RDO) technique to dynamically choose the number and locations of the blocks that refer to ER-frames.

As have described, JM exploits RDO technique at the stages of motion estimation and mode decision (using formulae (3.1.1) and (3.1.2), respectively) for coding

parameter selection. However, the RDO is designed for error-free environments. In order to provide error resilience, we incorporate the expected end-to-end distortion as computed in the previous section within JM RDO framework. Toward this goal, the only change in formula (3.1.1) is the replacement of source distortion D_{src} with the expected end-to-end distortion $E(D_{ee})$, where the $E(D_{ee})$ is the sum of the distortion contribution of the individual pixel in the block currently under motion estimation, *i.e.*, $E(D_{ee}) = \sum_{i \in block} d_n^i$ for the d_n^i calculated by formula (3.1.5)$\sum_{i \in block} d_n^i$. Hence, the resulting error-resilient RDO (ER-RDO) is as follows:

$$J(mv, ref) = E\{D_{ee}(mv, ref)\} + \lambda_{err} R_{block}(mv, ref) \tag{3.1.7}$$

where λ_{err} is equal to $(1-p)\lambda$ according to [66], in which the λ denotes the Lagrange multiplier in error-free environment. Rather than applying RDO formula (3.1.1) for motion vector and reference frame selection as that in JM, our approach adopts ER-RDO formula (3.1.7) such that the number and locations of blocks referring to ER-frames are determined by taking into account for channel conditions. Besides, we also incorporate error resilience for mode decision. By replacing the source distortion D_{src} in formula (3.1.2) with the expected end-to-end distortion $E(D_{ee})$, the ER-RDO for mode decision is defined as

$$J(mode) = E\{D_{ee}(mode)\} + \lambda_{err} R_{MB}(mode) \tag{3.1.8}$$

where $\sum_{i \in MB} d_n^i$the Lagrange multiplier λ_{err} is identical to that used in formula (3.1.7). Since optimal coding parameters (including motion vectors, reference frame indices and coding modes) are selected according to ER-RDO, the number and locations of blocks that refer to ER-frames are determined adaptively to varying channel conditions and various content characteristics.

3.1.3 Computational Cost Reduction

Even though the proposed scheme may improve the error- resilience of the coded video, it suffers from increased cost in computation. To reduce the computational cost of our error- resilient scheme, three techniques are further proposed. They are

reference-frame selection, *reference frame skipping*, and *long-term motion search center prediction*.

3.1.3.1 Reference Frame Selection

Motion vector selection taking end-to-end distortion into account often incurs impractical complexity because the distortion needs to be calculated for each candidate reference block in the search window to find the best one. To reduce the computational cost, we proposed to adopt JM RDO for motion vector selection, and utilize ER-RDO only for reference frame selection. Namely, end-to-end distortion is only calculated in reference frame selection. The reason is that blocks in the same reference frame have similar lengths of error propagation paths and thus, JM RDO considering source distortion at low cost in computation should be good enough to choose the best one among them. However, since blocks in different reference frames have different error propagation lengths, it is worth to take error propagated distortion into account, even with more computational cost. So, ER-RDO which considers end-to-end distortion is adopted in selecting reference frames. In our approach, we refer to JM RDO formula (3.1.1) as RDO_{mv} for motion vector selection, the ER-RDO formula (3.1.7) as $ER\text{-}RDO_{ref}$ for reference frame choice, and the ER-RDO formula (3.1.8) as $ER\text{-}RDO_{mode}$ for mode decision. As an example, for a block on current frame f_n, its motion search is performed on five reference frames, f_{n-1}, f_{n-2}, f_{n-3}, f_{n-4}, and f_{ER}, each of which has its best motion vector decided by RDO_{mv}. Among the five motion vectors, the choice is made according to $ER\text{-}RDO_{ref}$, and thus, the reference frame under current block mode is determined. The above process goes for each block mode, and finally the best mode is decided by $ER\text{-}RDO_{mode.}$

3.1.3.2 Reference Frame Skipping

This subsection presents how to skip unnecessary reference frames to reduce time complexity in motion estimation. As have described, even though predicting from near frames may have better coding efficiency, it suffers from longer error propagation. Predicting from ER-frames, however, alleviates the propagation at the expense of higher bit-rates since the correlation between the ER-frame and current frame becomes weaker in general as they are more widely separated. Hence, it is

worth to determine the dominant cost between bit-rate and distortion. Toward this goal, we refer to both ER-frame and the immediately previous frame as *dominant frames* and use these two frames to decide whether it is necessary to go on for other reference frames. Assume current coding frame is *n*. For its two dominant frames, if frame *n −1* is selected, motion estimation will go on for reference frames *n −2*, *n −3*, and *n −4*; otherwise, the motion estimation for current block is early terminated (*i.e.*, only two reference frames are used). The idea behind this is that if frame *n −1* is selected according to ER-RDO, it indicates that coding efficiency dominates the cost function and thus, motion estimation shall continue with reference frames *n −2*, *n −3*, and *n −4* which have similar characteristics with frame *n −1*. This could be the case that the channel is at a low packet loss rate or the video is with high-motion content. On the contrary, selecting ER-frame indicates that error propagation distortion dominates the cost function and thus, it is unnecessary to check remaining three frames because they cannot provide better error resilience.

3.1.3.3 Long Term Motion Search Center Prediction

Since ER-frames can be far away from current frame, the conventional way of using MBs in current frame to predict motion search center may not be adequate, especially for moving objects. A better search center across multiple frames is necessary. Motion vector composition is a good technique for this purpose. Considering a simple example that an object with motion $MV_{n\rightarrow(n-1)}$ between frames f_n and f_{n-1}, and $MV_{(n-1)\rightarrow(n-2)}$ between f_{n-1} and f_{n-2}., the most common way to represent the corresponding motion between f_n and f_{n-2} would be $MV_{n\rightarrow(n-1)}+ MV_{(n-1)\rightarrow(n-2)}$. MV composition can be extended to across more frames. Assuming that current frame is f_n and ER-frame is f_0, the predicted MV (called *PMV*) from f_n to f_0 can be represented as

$$PMV_{n,0} = \sum_{i=1}^{n} MV_{i,i-1} \qquad (3.1.9)$$

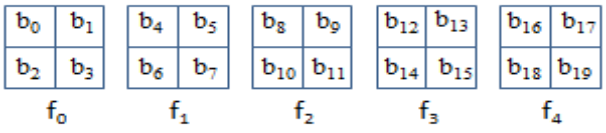
In block-based motion estimation, however, the area pointed by a MV may not align on block boundary. Several methods [69][70] have been proposed to choose proper MVs for composition. Without loss of generality, we adopt FDVS method proposed in [70], where the MV associated with the block having the largest

overlapping area out of overlapped MBs is selected. Let b_0~b_{19} in Fig. 29(a) denote aligned blocks in frames f_0~f_4. To calculate $PMV_{4\to0}$ for block b_{16}, since the area pointed by $MV_{4\to3}(b_{16})$ is overlapped with four blocks as shown in Fig. 29(b); the MV associated with b_{15} which has the largest overlapping area is selected for composition. Similarly, the MVs associated with b_{10} in f_2 and b_4 in f_1 are selected. As a result, $PMV_{4\to0}$ of b_{16} is obtained by $MV_{4\to3}(b_{16}) + MV_{3\to2}(b_{15}) + MV_{2\to1}(b_{10}) + MV_{1\to0}(b_4)$ according to FDVS.

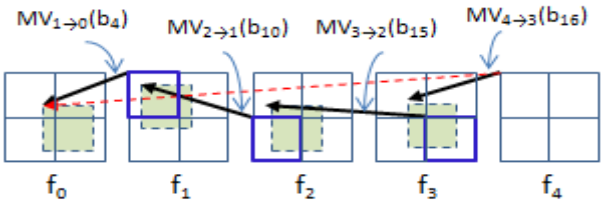
It can be seen that, the PMV calculation using formula (3.1.9) relies on the availability of all MVs between f_n and f_0, but these MVs may not be all available in the encoding buffer. To save memory space usage, we use an Accumulated Motion Vector (AMV) to represent the latest composite MVs. Let $AMV_{i\to0}$ denote the composite MV pointing to ER-frame (say f_0) by a block in f_i. Obviously, $AMV_{i\to0}$ can be recursively derived from $AMV_{(i-1)\to0}$ using composition formula (3.1.10), where the initial term $AMV_{0\to0}$ is (0,0) because f_0 is an ER-frame.

$$AMV_{i\to0} = MV_{i\to(i-1)} + AMV_{(i-1)\to0} \quad \text{for } i>1 \qquad (3.1.10)$$

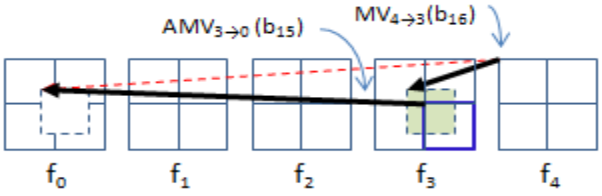
As an example in Fig. 29(c), assuming that AMVs of the blocks in f_3 are available, $AMV_{4\to0}(b_{16})$ can be derived from $MV_{4\to3}(b_{16}) + AMV_{3\to0}(b_{15})$. It is interesting to note that, with AMVs of f_3, the MVs of f_3, f_2, and f_1 are no longer required in $PMV_{4\to0}$ composition and thus can be removed from the encoder buffer. Compared with FDVS in Fig. 29(b) which requires MVs of all the frames along current coding frame to the ER-frame, using AMV can save plenty of memory space. Once PMVs for all the blocks in f_4 have been conducted, these PMVs become AMVs of f_4 and can be used in PMV calculation for f_5. By updating AMVs frame by frame, only AMVs of one frame need to be maintained for PMV composition. With proper PMVs, motion search range on ER-frames can be reduced to save computational cost. In our experiments, the search windows on ER-frames are set to 4x4; while on other reference frames, they are set to 32x32.



(a) Aligned blocks



(b) FDVS



(c) Aaccumulated FDVS

Fig. 29 Motion vector composition using FDVS and accumulated FDVS

3.1.3.4 Summarization of the Proposed Method

The flow chart of the proposed method is shown in Fig. 30, where the frame under processing is f_n. For each MB in f_n, first perform motion estimation on dominant frames, f_{n-1} and f_{ER}, and then select reference frame between them by using ER-RDO$_{ref}$. Perform motion estimation using JM RDO on f_{n-2}, f_{n-3} and f_{n-4} if f_{ER} is not selected; otherwise, choose the next block mode and repeat the above process until all modes of current MB are done. Then, perform mode decision using ER-RDO$_{mode}$ to determine the best mode for current MB. After all MBs in f_n have been processed, scene-change detection is performed. Replace ER-frame with f_n if f_n is a scene-change frame; otherwise, only AMV needs to be updated. Since the proposed algorithm is independent of scene change detection methods, any method that can correctly detect scene changes can be incorporated into our approach. In later experiments, a frame is regarded as a scene-change frame if the ratio of its intra-coded MBs is higher than 50% because such a frame can provide high ability to alleviate error propagation.

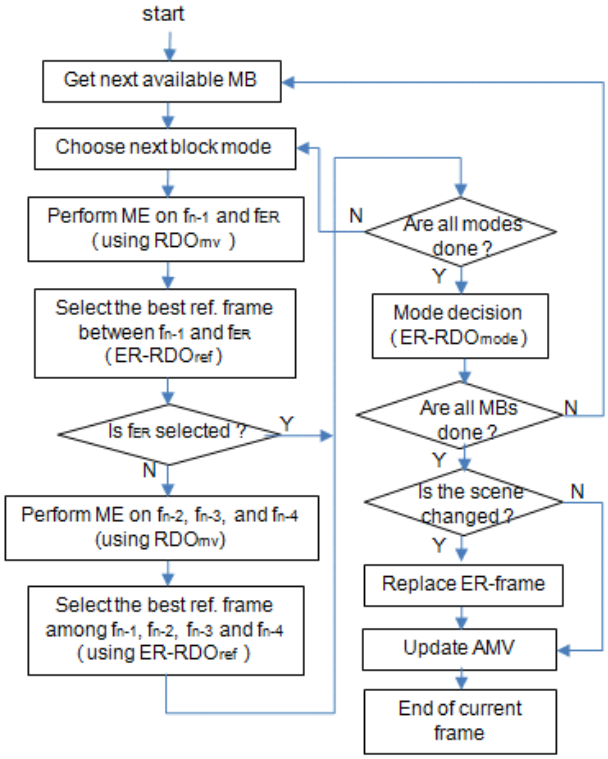


Fig. 30 The flow chart of the proposed MRF-MCP with fast motion estimation

3.1.4 Experimental Results

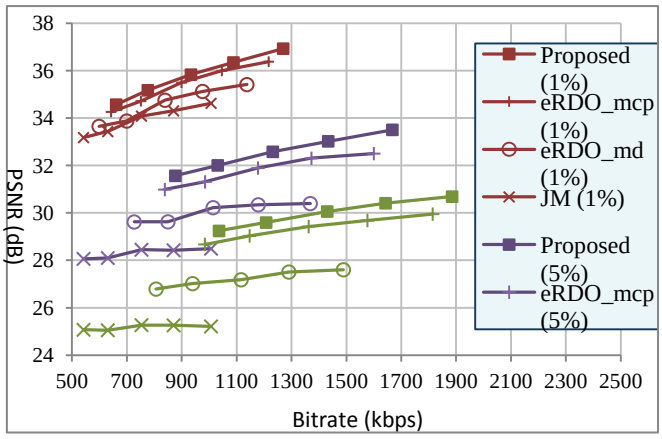
The performance of the proposed error resilient scheme is evaluated with respect to rate-distortion performance, mismatch of packet loss rates, and the effects of ER-frames.

3.1.4.1 Overall Rate Distortion Performance

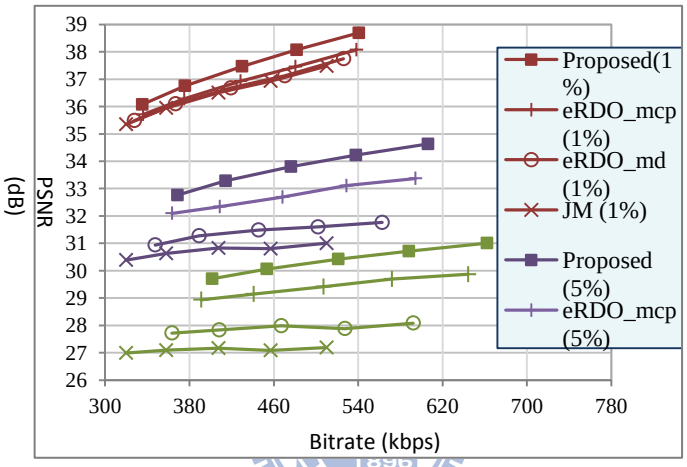
The rate-distortion performance of the proposed method was examined with three different packet loss rates (PLR=1%, 5%, and 10%) on three CIF sequences: *Foreman*, *Football* and *News*. 200 frames are encoded for each sequence and GOP structure is IPPP. The methods adopted include: *Proposed*, *JM*, *eRDO_md [66]*, and *eRDO_mcp [71]*. All these methods are implemented based on JVT reference software, JM [47], with rate control disabled. The *Proposed* is the full version of our error-resilient scheme with computational time reduction techniques included; the *JM* is JM software; the *eRDO_md* is the approach in *[66]*, where ER-RDO was applied in mode decision; the *eRDO_mcp* is the method in *[71]*, where ER-RDO is not only applied on mode decision, but also on motion estimation. All these methods use five reference

frames and the performance was examined in a simulated Bernoulli channel which assumes that packets are lost independently of each other.

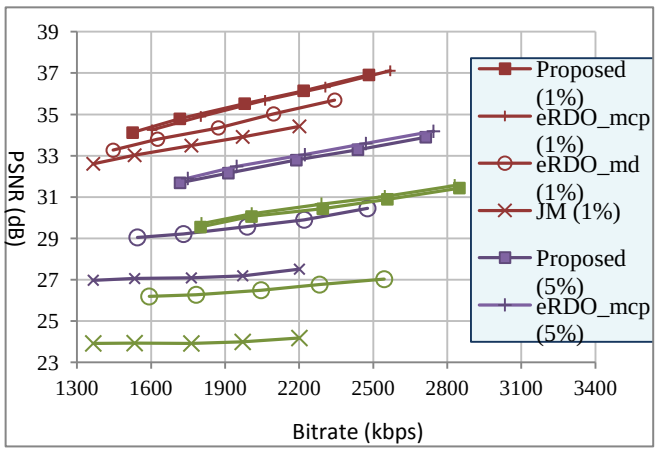
The performance result is shown in Fig. 31, where the average PSNR values as a function of bitrates for *Foreman*, *News*, and *Football*, respectively, are presented. The average PSNR values are obtained from 300 runs of the experiment, each with different seeds of the random number generator for packet loss patterns. From Fig. 31, it is observed that the JM scheme has the worst performance as expected in most of the cases. This is due to the fact that channel conditions are not considered in JM RDO which takes into account source distortion only. eRDO_mcp performed better than eRDO_md for all the cases. Their performance gaps become larger as the loss rate increases, indicating that error-resilient capability can be enhanced by accounting for overall distortion at the stage of motion vector selection. Compared with eRDO_mcp, the proposed method performed even better, especially for sequences: *Foreman* and *News* at high packet loss rates. This implies that, by considering end-to-end distortion in both motion-vector selection and mode decision, eRDO_mcp scheme still cannot have sufficient error robustness because it utilizes only near reference frames. By adopting long-term ER-frames as reference, the proposed method can suppress error propagation effectively and thus improve the error resilience. As for *football* sequence, since it is a high-motion video, predicting from long-term reference frames will suffer from low coding efficiency and thus very few blocks will choose to predict from ER-frames according to RDO technique. This issue will be further discussed in Section V.D. Since not many benefits can be obtained from the use of ER-frames, the proposed method performed almost equally to eRDO_mcp for *Football* sequence. To summarize, the overall results in Fig. 31 show that compared with other methods, the proposed approach is more robust to packet loss.



(a) Foreman sequence (PLR = 1%, 5%, 10%)



(b) *News* sequence (PLR = 1%, 5%, 10%)

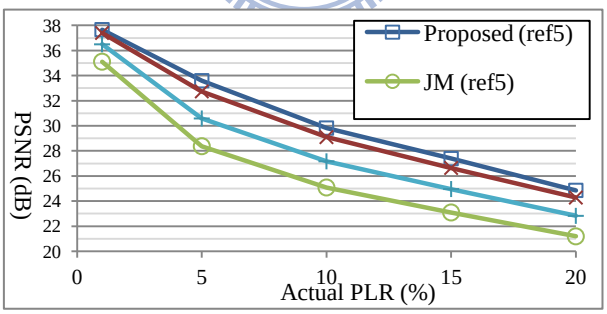


(c) *Football* sequence (PLR = 1%, 5%, 10%)

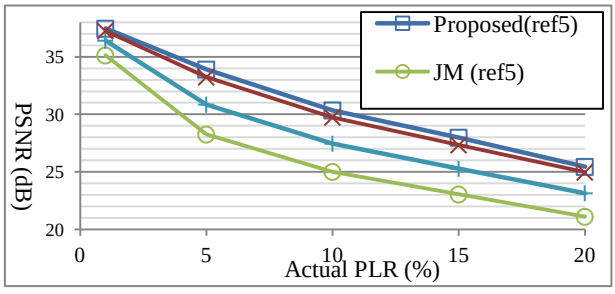
Fig. 31 R-D Performance comparison using five reference frames

3.1.4.2 Mismatch of Packet Loss Rate

To see the effects of a mismatch in the packet-loss-rate values (PLR), experiments were conducted for the mismatch between the assumed PLR at the encoder and the actual PLRs in the network. To this end, the video sequences were coded assuming a certain PLR and transmitted over channels with various packet loss rates. The assumed PLRs of 5% and 10% are considered for *Foreman* sequence. Fig. 32 presents the results for the methods: *Proposed*, *eRDO_md*, *eRDO_mcp*, and *JM* (all of them use five reference frames as described in Section 3.1.4.1). From these figures, it can be seen that in all scenarios the proposed method offers the best performance exhibiting improved robustness to the mismatch. Furthermore, if PLR= 10% is assumed, better performance is observed by all methods (except JM) in comparison to the case where PLR=5% is assumed for the same actual packet loss rates. The JM method does not take packet loss rates into account during encoding and therefore, its performance is not affected by the assumed PLRs. According to these empirically observed effects, it is advisable to assume worse network conditions at the encoder. This will guarantee better overall performance when the actual packet loss rate is not as severe as assumed.



(a) Assumed PLR = 5% (*Foreman*, CIF, 1614kbps, five reference frames)



(b) Assumed PLR=10% (*Foreman*, CIF, 1614kbps, five reference frames)

Fig. 32 Performance for mismatch with an assumed PLR of (a) 5%; (b) 10%

3.1.4.3 Effect of ER-frames

To examine the effects of incorporating ER-frames into reference frames, Fig. 33 presents the percentage distribution of intra-coded blocks, inter-coded blocks predicting from ER- frames and inter-coded blocks predicting from four previous frames for the Bernoulli channel at different packet loss rates. From Fig. 33, it is observed that the percentage of blocks predicting from ER-frames can be up to 18% for *Foreman* sequence. Since the result is the average of all frames in the sequence, this percentage can be even higher for some individual frames, indicating that ER-frames did serve as an important role in *Foreman* sequence. For *Mobile* sequence, since it is a high-motion video, predicting from ER-frames will suffer from too much increase in bit-rate and therefore most of blocks chose intra-coding to stop error propagation. The percentage of blocks predicting from ER-frames ranges from 5% to 7% only. As for *News* sequence, since it is a low-motion video, lost data can be concealed well without much error propagation. As a result, the ER-frames which are mainly utilized for suppressing error propagation cannot provide much help in *News* sequence. The percentage of blocks predicting from ER-frames is about 9% for three loss rates. However, even with low ER-frame reference ratio in *News* sequences, the proposed method is still able to achieve the PSNR gain of up to 1dB and 2.5dB (see Fig. 32), when compared to eRDO_md and eRDO_mcp, respectively because both of them did not use long-term reference frames.

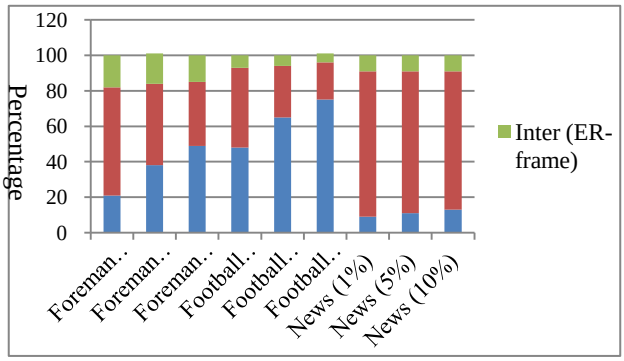


Fig. 33 Frame reference distribution

3.1.4.4 Computational Cost

To examine how the computational cost can be saved by these time-reduction methods, experiments were conducted for two versions of the proposed method: *Proposed1* which denotes the method presented in Section 3.1.2 and *Proposed_full* which denotes the Proposed1 added with computational time reduction techniques in Section 3.1.3. The two methods are implemented based on JM with full-search motion estimation and five reference frames enabled. The results are shown in Fig. 34, where the execution time relative to the time executed by JM is presented. As Fig. 34 shows, Proposed1 which adopts ER-RDO at motion estimation stage obtained an obvious increase in execution time when compared to JM. The increased time, however, can be reduced substantially by using the proposed computational time reduction techniques. As shown in Fig. 34, Proposed_full achieves a reduction of up to 40% in execution time, when compared to Proposed1. The time reduction increases as the packet loss rate increases. This is mainly due to that the probability of selecting ER-frame as reference frame increases as packet loss rate increases, resulting in more short-term reference frames to be skipped. With such time reduction, Proposed_full can run even faster than original JM which does not support error resilience in its MRF-MCP.

To examine how the performance might be affected by these computational time reduction methods, Fig. 35 shows the R-D performance of Proposed1 and Proposed_full for different video sequences at different packet loss rates. We observed that the performance gaps between the two methods are all within 0.5dB, meaning that, by moving ER-RDO from motion- estimation stage to reference-frame selection and skipping unnecessary reference frames, Proposed_full can save a lot of computational cost without causing much loss in performance. In some cases, Proposed_full even outperformed Proposed slightly. This stems from the fact that Proposed_full uses MV composition to locate the motion search center on ER-frame and thus, improve coding efficiency and overall performance.

To summarize, the overall results in Fig. 34 and Fig. 35 show the superiority of proposed computational time reduction methods in regard to time saving with neglectable loss in performance. Moreover, since Proposed_full applies ER-RDO at the stage of reference frame selection which is independent of motion estimation, any

block-matching search algorithm or early termination method that is used to speed up motion estimation process can be incorporated into our approach. According to our experiments, up to 80% and 76% of the execution time in Fig. 34 can be further reduced by simply changing the motion estimation algorithm used in Proposed_full from full-search to EPZS [72] and Multi-Hexagon-Grid-Search (UMHexagonS) [73] in JM, respectively. Since it is easy to adapt the proposed method to different motion estimation algorithms, it can be applied to advanced motion-estimation approaches to provide error robustness for time-sensitive applications. This paper focuses on providing a cost-effective error-resilient scheme. Selecting the best motion estimation method to be integrated is beyond the scope of this paper.

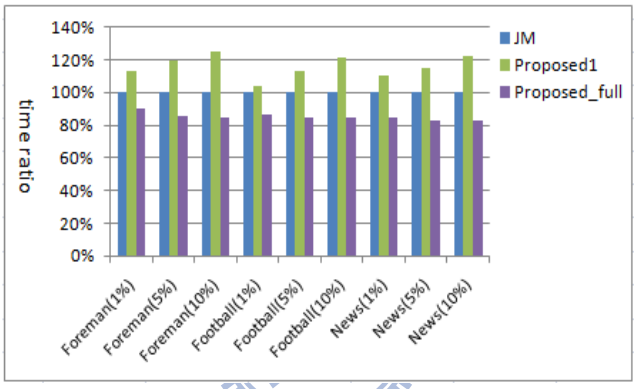
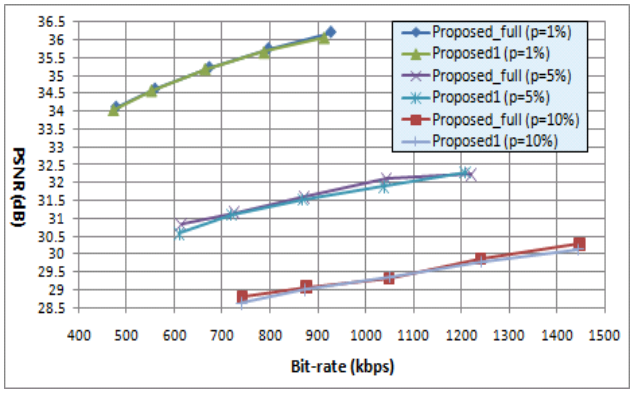
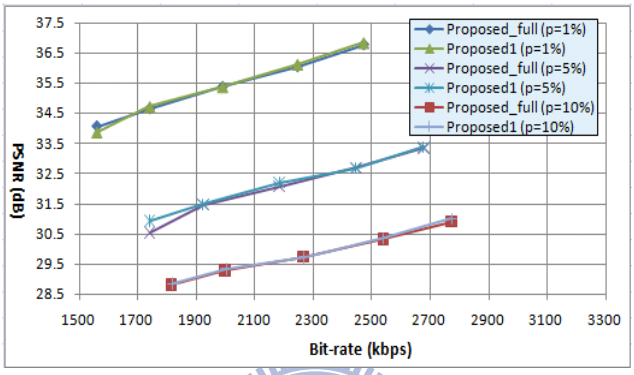


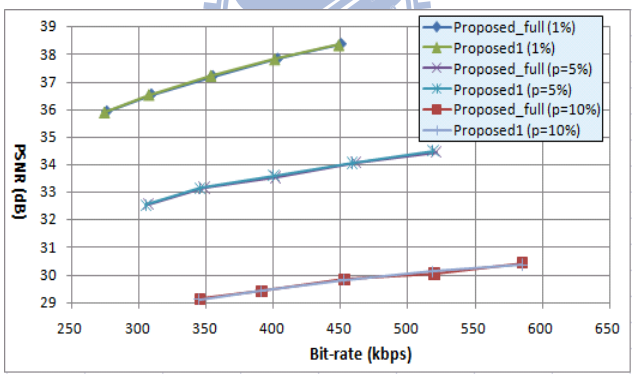
Fig. 34 Execution time ratio of different methods



(a) *Foreman* sequence



(b) *Football* sequence



(c) *News* sequence

Fig. 35 Performance with and without computational time reduction techniques.

3.2 Error Resilient Video Coding Based on Hierarchical B Pictures

3.2.1 Introduction

A hierarchical B-picture coding structure has demonstrated superior compression performance than a conventional one[74], in this section, we proposed an error resilient coding based on hierarchical B pictures. In our approach, a new hierarchical coding structure which combines two conventional hierarchical coding structures is employed to reduce the distance between a lost picture and its recovering pictures. In addition, based on the new structure, an improved estimation method is also proposed to further increase the accuracy of recovering motion.

3.2.2 Motivation

A typical hierarchical prediction framework with 4 dyadic hierarchy stages is illustrated in Fig. 36, where the key frames (which can be I or P frames) are coded in regular intervals. A key frame and all frames that are temporally located between the key frame and the previous key frame form a group of pictures (GOP). The remaining B frames are hierarchically predicted using two reference frames from the nearest neighboring frames of the previous temporal level as shown in Fig. 36. For optimized encoding, it is better to set smaller QPs for the frames that are referenced by other frames. In the Joint Scalable Video Model 11 (JSVM11) [75], QPs of the B frames at level-1 equal to the QPs of the I/P frames plus 4, and the QPs increase by 1 from one hierarchical level to the next level.

We refer to the I/P frames at the lowest hierarchical level as key frames; the B frames at intermediate levels as reference B frames (RB frames) because they are used as reference; and the B frames at the highest level as non-reference B frames (NRB frames) because they are not used as reference. Hierarchical B-frame structure has the characteristic that the frames at different levels have different reference distances (which means the temporal distance between a frame and its reference frame). Among the three types of frames, key frames have the longest reference distance, RB frames the medium, and NRB frames the shortest.

Temporal error concealment in hierarchical B-picture structure includes lost-motion and lost-pixel recovery. The lost information is estimated by referring other valid frames. The lost-motion would be estimated by interpolating, extrapolating, or compositing the motion vectors of the blocks in the *motion prediction frame* (MF). Then, the lost-pixel could be recovered from pixels of the *data prediction frame* (DF) according to estimated motions. DF can be different from MF. To have better error concealment, it is important to select appropriate MFs and DFs. Because the correlation of the lost frame and the referred frame increases when the frame distance decreases, frames within smaller distance could provide more reliable recovering information and thus better concealment performance. To explore the relation between error concealment performance and recovery distance, experiments were conducted for Foreman sequence (CIF), where assume a four-level hierarchical B-frame structure is adopted and frame-loss occurs on every level-1 frame. To recover these lost frames, temporal concealment is applied with various selections of DFs and MFs. As illustrated in Fig. 37 Experimental setting for different combinations of motion frames (DF1, DF2, and DF3) and data frames (MF1, MF2, and MF3)., to recover frame n+4, both MFs and DFs were chosen from frames n+5, n+6, or n+8, denoted by MF3/DF3, MF2/DF2, and MF0/DF0, respectively, because they are on levels 3, 2, and 0, respectively. We conducted experiments for all the possible combinations of DFi and MFi, where i=0, 2, 3, and the results are shown in TABLE. IX. Note that some combinations in Table I may not be realistic because the DF frames are unavailable (not yet been decoded) when performing error concealment, e.g., DF2 and DF3 are not available when level-1 frames are under error concealment. We still simulate these cases for illustrating the rationale behind the proposed method. In the TABLE. IX, the cell (MF0, DF0) means to use level-0 frames (i.e., frame n+8) as both DF and MF for recovery. Note that frame n+8 is the reference frame of the lost frame in this example, choosing MF and DF in this way is known as temporal direct mode (TDM) [76] of H.264/AVC.

Instead of using reference frames as both MFs and DFs, WTDM [7] chooses MFs from the frames on the next level of the lost frame to reduce motion recovering distance. In our example, the corresponding performance is the case shown in cell (MF2, DF0). Compared with the one in (MF0, DF0), the performance is improved because motion recovering distance becomes shorter. From this result we might

expect that choosing MF3 as the motion prediction frame should produce the best error concealment quality because MF3 has the shortest motion recovering distance. However, it is not as expected when DF0 is adopted as the data prediction frame, as can be seen in Table I where (MF3, DF0) performs worse than (MF2, DF0). The reason might be that even though MF3 is located close to the lost frame, it is far away from DF0. Therefore, the motion vectors in MF3 need to be greatly extrapolated to reach DF0, resulting in the decrease in motion accuracy and hence the degradation in error concealment quality. The result implies that MF and DF should not be determined independently.

Although many methods have been proposed to select proper MFs to reduce motion recovering distance, how to reduce data recovering distance is seldom discussed. Most of studies use pixels on the reference frames to recover missing pixels. Selecting DFs in this way may result in long data recovering distance. Take level-1 frame loss as an example, reference frame of the frame n+4 in Fig. 36 are frames n and n+8, both of them are four frames away from frame n+4 in display order; namely, the data recovering distance will be 4 if frame n+4 is lost. TABLE. X shows data recovering distances for frame loss in different hierarchical levels, respectively, assuming that their reference frames are used for recovery. It can be seen that data recovering distances is large, especially for the cases of frame loss in lower hierarchical levels. However, long data recovering distance may result in severe quality degradation, as it can be seen in Table I where the performance with DF1 is always the worst, while that with DF3 is always the best, if the same MFs are adopted. This implies that if data recovering distance can be reduced, it is very promising that error concealment performance can be improved. However, with hierarchical coding structure, it is hard to take advantage of those frames with recovering distances shorter than reference frames because these frames have not yet been decoded when the lost frame is under recovery. To solve this problem, we propose a variation of hierarchical B structure to reduce data recovering distance.

In summary, both motion and data recovering distances influence error concealment performance significantly, in this paper, an approach based on hierarchical B-picture structure is proposed, which is aimed at jointly determining MFs and DFs to reduce both motion and data recovering distances.

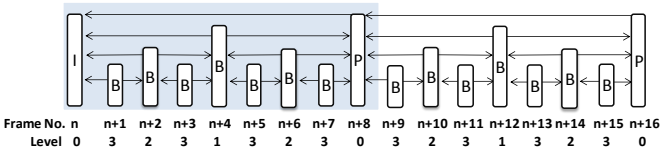


Fig. 36 Hierarchical B-picture prediction structure

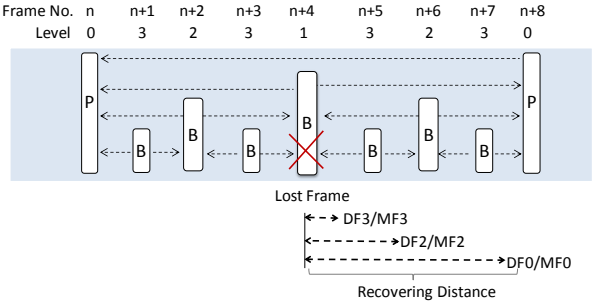


Fig. 37 Experimental setting for different combinations of motion frames (DF1, DF2, and DF3) and data frames (MF1, MF2, and MF3).

TABLE. IX Experimental result for all combinations of motion frames and data frames.

Concealment performance (db)			
Motion Frame / Data Frame	MF0	MF1	MF2
DF0	20.4[13]	23.2[14]	21.9
DF2 (not valid)	23.7	26.6	27.2
DF3 (not valid)	26.2	29.5	29.8

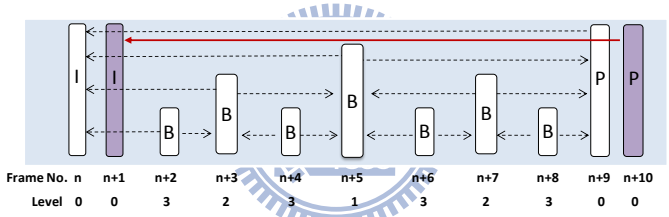
TABLE. X Minimal pixel recovering distance for lost frames at different hierarchical levels

Hierarchical level	Lost frame	Recovering frame	Recovering distance
Level 0	8	0	8
Level 1	4	0, 8	4
Level 2	2	0, 4	2
	6	4, 8	2
Level 3	1	0, 2	1
	3	2, 4	1
	5	4, 6	1
	7	6, 8	1

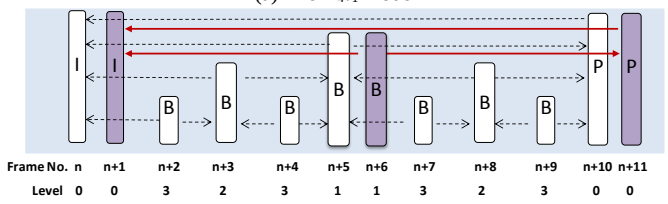
3.2.3 Proposed Method

Here a variation of hierarchical B structure is proposed to reduce quality degradation. As mentioned above, key frames have the longest reference distance,

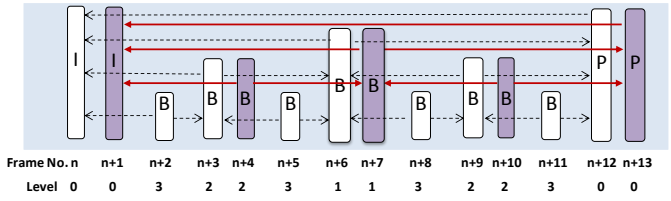
resulting in the worst error concealment performance when it is lost. To improve the performance, a hybrid model called H_{N+1} is proposed, which combines an N-level with a one- level hierarchical B-picture structure. As an example in 0(a) where N=4, by combining a 4-level and a 1-level hierarchical B structures, each key frame in the resulting sequence has a neighboring frame located at the same level and predicted from the same reference frames. Rather than encoding frame n+10 as a level-2 RB frame in the conventional hierarchical B structure shown in 0 (a), the proposed model will encode frame n+10 as a key frame (P frame). We call a key frame and its neighboring key frame as the *buddy frames* which are a pair of frames used to recover each other when there is a loss. In 0 (a), frame n+9 and frame n+10 are buddy frames. If frame n+9 is lost, instead of using its reference frame (frame n) for missing pixel recovery, its buddy frame n+10 is used. Compared with WTDM[7] described in the previous section, the proposed H_{4+1} reduces the recovering distance of key frames from eight to one frame. By employing buddy frames in this way, error concealment performance of key-frame loss can be improved significantly.



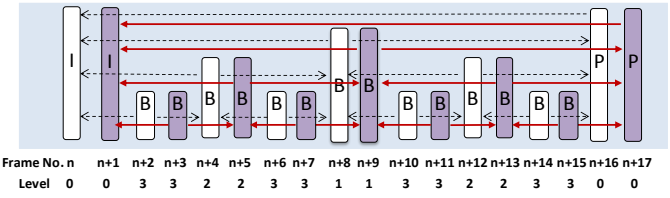
(a) The H_{4+1} model



(b) The H_{4+2} model



(c) The H_{4+3} model



(d) The H_{4+4} model

Fig. 38 The proposed hybrid model based on hierarchical B structure

In addition to key frames, RB frames also suffer from the problem of long recovering distance. The proposed buddy frames can also be applied to RB frames to reduce the recovering distance. The hybrid model H_{N+2} which is a variation of H_{N+1} is proposed for this. It combines an N-level hierarchical B structure with a 2-level hierarchical B structure as an example in 0 (b) where N=4. By combining a 4-level and a 2-level hierarchical B structures, not only each key frame but also each level-1 RB-frames such as frame n+5 in the resulting sequence have buddy frames located at the same level and predicted from the same reference frames. In 0 (b), if RB-frame 4 is lost, instead of using its reference frames (frames n and n+8 in Fig.1) for missing pixel recovery, its buddy frame (frame n+6 in 0 (b)) will be used. Compared with WTDM[7] where reference frames are used for recovery, H_{4+2} model reduces the recovering distance of RB-frame n+5 from four frames (the distance between frame n+4 and its reference frames shown in Fig.1) to one frame only (the distance between frame n+5 and its buddy frames in 0 (b)).

Similarly, the proposed buddy frames can also be applied to level-2 RB-frames and level-3 NRB-frames to reduce their recovering distances. Two variations of hybrid model, H_{N+3},and H_{N+4} , are shown in 0 (c) and (d), respectively. The H_{N+3} model in 0 (c) combines a 4-level and a 3-level hierarchical B structures; while the H_{N+4} model in 0(d) combines a 4-level and a 4-level hierarchical B structures. As observed in these figures, H_{4+3} model reduces the recovering distance of level-2 RB-frame (e.g., frame n+3) from two to one frame and H_{4+4} model keeps the recovering distance of level-3 NRB frame (e.g., frame n+2) as one frame.

The proposed various hybrid models can be generalized as a H_{N+M} model which means that the resulting sequence is the combination of an N-level hierarchical B-picture structure and an M-level one. The encoder architecture of the H_{N+M} model is depicted in Fig. 4(a). As the figure shows, the frames in the sequence are split into two groups: *normal frames* and *buddy frames* first, and then each group will go through a standard hierarchical B picture encoder to perform motion estimation, transform, quantization and entropy coding. The normal frames are encoded as an N-level hierarchical structure and the buddy frames as an M-level structure, resulting in a H_{N+M} sequence.

Different hybrid models are made up by different normal frames and buddy frames. For example in H_{4+1} model, the buddy frames consist of frames 1, 10, 19,

28, …, etc., while in H_{4+2} model, they are frames 1, 6, 11, 16, …, etc. For N=4, the buddy frames of the four variations are summarized as follows, where *m* is an integer.

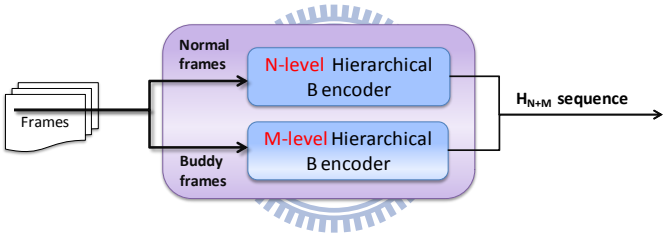
H_{4+1} : *frames 1, 10, 19, 28, …, 9m+1*

H_{4+2} : *frames 1, 6, 11, 16, …, 5m+1*

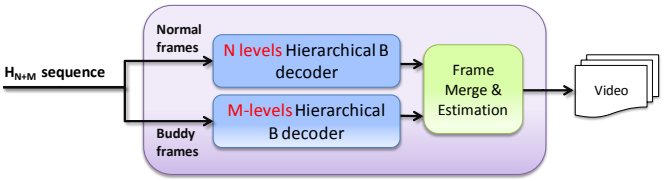
H_{4+3} : *frames 1, 4, 7, 10, …, 3m+1*

H_{4+4} : *frames 1, 3, 5, 7, …, 2m+1*

The decoder architecture of the proposed hybrid model H_{N+M} is depicted in 0(b), where the received frames are first split into two groups, normal frames and buddy frames. Then each group will go through a standard hierarchical B decoder for entropy decoded, de-quantized, and inversely transformed. Normal frames are decoded with an N-level hierarchical structure; while buddy frames are decoded with an M-level one. Finally, the frame-merge and estimation procedure is used to reconstruct the order of frames for generating output sequence. If the decoder does not receive the two structures intact, the estimation procedure will be adopted to estimate the lost data. The details of estimation method will be described in the next section.



(a) Encoder architecture



(b) Decoder architecthre

Fig. 39 Architecture of the proposed hybrid model HN+M

3.2.4 Estimation of lost pictures

In the proposed method, we assume that each frame is divided into three slices in raster scan order. In case of packet loss, it will result in successive macroblock loss regardless of frame types and levels. Each lost block is recovered based on temporal correlation since the neighboring blocks are also lost. We refer to the pictures whose pixels are used to predict the missing pixels as the *data prediction frame* (DF) and the

pictures whose block motions are used to predict the motion of the missing blocks as the *motion prediction frame* (MF). In our method, DF can be different from MF.

To serve as DFs requires that these pictures are decoded earlier than the lost picture. Therefore, for hierarchical B structure, almost all the error concealment methods choose reference frames of the lost frame to serve as the DF. The DF can be in backward direction, forward direction or both. Since data correlation among pictures involved tends to considerably weaken as the temporal distances among these pictures become longer, for a lost picture, it is better to choose pictures near in the display order to serve as its DFs. Therefore, in the proposed hybrid model, we choose buddy frame of the lost frame to serve as DF because it is usually located near in temporal distance. However, not every frame has buddy frame. For example, in H_{4+1} model, only level-0 frames have buddy frames; while in H_{4+2} model, both level-0 and level-1 frames have buddy frames. If the lost frame has no buddy frame, we simply use its reference frames to serve as DF. That is, for the lost frame F_t^l with hierarchical level l at time instant t, we select its DF as

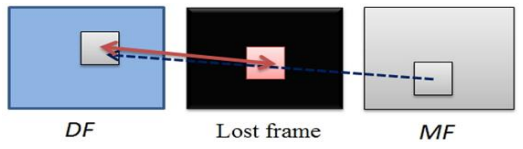
$$\mathrm{DF}(F_t^l) = \begin{cases} F_{t_{buddy}}^{l}, & \text{if has buddy frame.} \\ F_{t_{ref}}^{k}, & \text{otherwise.} \end{cases}$$

where k can be l , l-*1*, or l-*2*, depending on what level the reference frame of the lost frame is. As an example, for the H_{4+2} model in 0 (b), if frame n+8 is lost, its DFs are frame n+5 and frame n+10 because it has no buddy frame. But if frame n+5 is lost, the DF will be its buddy frame n+6.

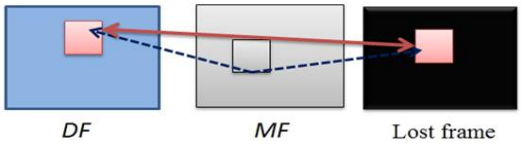
As for MFs, since we can obtain motion information of a frame even though this frame is not decoded, the MFs can be the frames later than the lost frame (in decoding order). As discussion in the section 3.2.2, how to choose MF depends on not only motion recovering distance but also pixel recovering distance. Therefore, instead of using reference pictures at lower levels or buddy frames at the same level, if the lost frame has buddy frame, we choose the nearest pictures at higher levels to serve as MFs because these pictures are temporally closer to the lost picture in display order. Otherwise, we choose the pictures at next level to serve as MFs in order to prevent motion interpolation/extrapolation. As an example in 0 (b), if the frame n+8 is lost, we will select frames n+7 and n+9 (rather than its reference frames n+5 and n+10) as

its MFs. Similarly, if frame n+5 is lost, we will select frames n+4 and n+7 (rather than its buddy frame n+6) as its MFs. This selection policy is applied to all frames except NRB frames which are at the highest level within the hierarchical structure. For NRB frames, the MFs are selected from the reference frame at the next lower level or the buddy frame at the same level. As an example, if NRB frame n+5 in 0 (c) is lost, its reference frame (frame n+3) at the next lower level is chosen as its MF because it has no buddy frame. But if NRB frame n+6 in 0(d) is lost, its buddy frame (frame n+7) at the same level will be chosen.

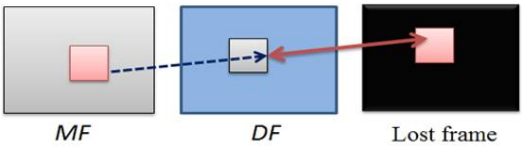
Once both DFs and MFs of the lost picture have been determined, for every block in MF, its motion vectors(s) are *composed*, *extrapolated*, or *interpolated* so that the motion vectors pointing to DF from the lost frame can be obtained. Such motion vectors are called *recovery motion vectors* (RMV). If DF and MF are on different sides of the lost frame along temporal dimension, the MV pointing to DF from MF are interpolated to obtain the RMV as illustrated in 0 (a), where the RMV is denoted using a solid arrow. If DF and MF are on the same side of the lost frame, the MV pointing to DF from MF are either extrapolated or composed to get RMV as illustrated in 0(b) and (c). Once all the RMVs have been derived, if a location on the lost picture is pointed by one or more than one RMV, its pixel is replaced by the average of these pointed pixels on the DFs. If a location on the lost picture is not pointed by any RMV, its pixel is replaced by co-located pixel on the DF.



(a) motion interpolation



(b) motion composition



(c) motion extrapolation

Fig. 40 Motion interpolation, composition, and extrapolation

3.2.5 Experimental Results

3.2.5.1 Effects of Hybrid Structures

To see the effects of the proposed hybrid model, experiment was first conducted for comparing the proposed H4+4 with a standard hierarchical B-frame structure with four levels. It is interesting to observe that, excluding the first I-frame, both structures contain two P-frames, two level-1 frames, four level-2 frames, and eight level-3 frames for every successive sixteen frames, as seen in Fig. 41 (a-b). With the same number of frames for each frame type, standard structure encodes the sixteen frames as two successive GOPs, while H4+4 encodes them as two independent GOPs with interleaved positions in display order.

Table.XI shows the resulting performance of the two structures. It can be seen that H4+4 performs worse than the standard structure for the error free case as expected because temporal prediction distance in H4+4 are much farther than that in the standard one. However, in case of packet loss, H4+4 shows superior performance. The result shows that, with the proposed hybrid structure, H4+4 did improve error resilience significantly.

3.2.5.2 Effects of Hybrid Structure Variations

There are four variations of hybrid models: H_{4+1}, H_{4+2}, H_{4+3}, H_{4+4}. This section examines how them affect error resilient capability. Since how often a key frame is encoded as an I-frame instead of a P-frame also affects the performance of the overall sequence, we adopt the same I-frame period, 32, for normal frames in the four hybrid models. As for buddy frames, three different I-frame period settings are used for comparison. The first setting, called *Equal*, is to use the same I-frame period (i.e., 32) for buddy frames in the four models. Since the amount of buddy frames are different from the amount of normal frames in some hybrid models, their I-frame positions in buddy frames and normal frames will be different. The second setting, *Sync1*, is to synchronize the positions of I-frames in normal frames and buddy frames. In order words, with Sync1, the I-frame periods of buddy frames are 4 in H_{4+1}, 8 in H_{4+2}, 16 in

H_{4+3}, and 32 in H_{4+4}. The third setting, *Sync2*, simply double the Sync1 I-frame periods for buddy frames, and keep I-frame period as 32 for normal frames.

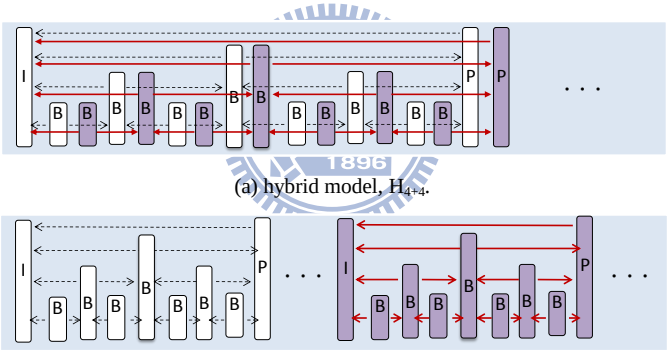
TABLE. XII shows the PSNR as a function of PLR for 12 combinations of the four hybrid models with three I-frame periods under four CIF sequences: *coastguard*, *flower*, *foreman*, and *news*. All combinations encode the same sequence using the same bit-rate for fair comparison and the results presented are the averages of 300 independent runs. It is observed that, among the three I-frame periods, Sync1 have the best performance. And, among the 12 combinations, H_{4+3} with Sync1 achieve the overall best performance for all the sequences.

3.2.5.3 Packet Loss Performance

Since H_{4+3} with Sync1 outperforms all other hybrid models, it was adopted for the comparison with other methods in a packet-loss scenario. The Bernoulli channel is adopted, which assumes that each packet is lost randomly and independently. Each frame was encoded into three slices in raster scan order. And, We assume one slide is transmitted by one packet. We compare H_{4+3} with Ji et al.'s method [77] and Zhu et al.'s method [78]. Ji et al.'s method called WTDM is a method based upon temporal direct mode (TDM) of H.264/AVC for error concealment in hierarchical B-picture prediction structure. The I-frame period is 32. Zhu et al.'s method duplicates each test sequence into two and then encodes by hierarchical B structure with staggered key frames in the two sequences. For example, if one sequence is encoded with the structure shown in Fig.1 where frames n, n+8, n+16, … are key frames, then the other one will have frames n+1, n+9, n+17, … encoded as key frames. This approach is characterized by that each frame at levels 0, 1, or 2 of one sequence will be at level 3 of the other sequence and vice versa, resulting in two fidelities of each frame. Two variations, *defaultQP* and *modifiedQP*, in their literature are adopted in our comparison. The defaultQP follows the QP assignment rules specified in JSVM11, while modifiedQP modifies the QPs of top-level frames to 51 to reduce bit-rates redundancy. The results in [78] show that rate-distortion performance of center decoder can be improved remarkably by modifiedQP in comparison to defaultQP. All these methods are implemented based on H.264 reference software, JM 16.0.

Fig.42 shows the result for four different methods with four test sequences. In Fig.42, the four methods encode the same sequence using the same bit-rate for fair

comparison and the results are the averages of 300 independent runs. It can be seen that, as PLR increases, WTDM curves drops much more quickly than others, showing its poor error resilience. By duplicating the entire sequence, defaultQP and modifiedQP achieve better error robustness than WTDM. Compared with defaultQP, the modifiedQP method shows better performance in low PLR because of its reduced bit-rate at top-level frames (NRB frames). However, such a reduction in bit-rate strongly affects its error concealment effectiveness and hence, degrades its performance dramatically at high PLR. Among all methods, the proposed H4+3 performed the best because it modifies hierarchical B coding structure by encoding more key frames and RB frames as buddy frames, resulting in reduced recovery distance and better error concealment effect, especially at high PLRs. To summarize, the overall results demonstrate that, by combining two hierarchical B-picture structures, the proposed hybrid model offers a better trade-off between bit-rate redundancy and error-resilient capability and thus, achieves the best performance among the four methods.



(a) hybrid model, H_{4+4}.

(b) original model.

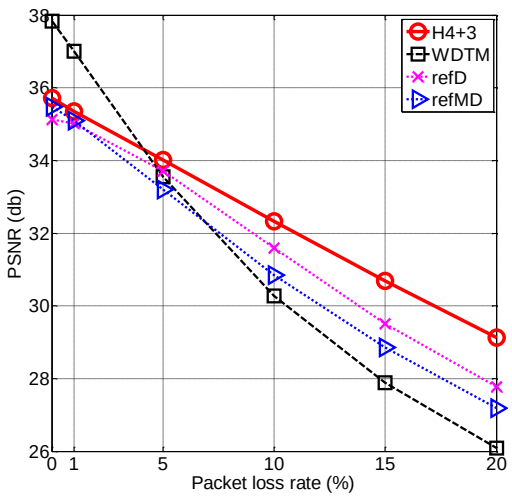
Fig. 41 Coding structures of hybrid model, H_{4+4}, and original model.

TABLE. XI Performance comparison between hybrid model, H4+4, and the original model. Both models encode Foreman sequence (CIF) at 800kbps.

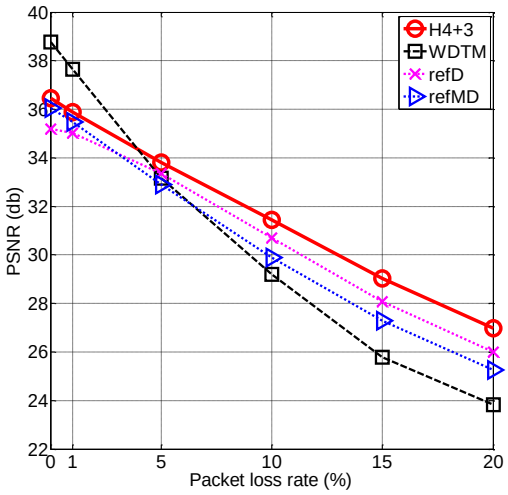
Concealment performance (db)		
Model / Loss rate	H_{4+4} hybrid model	Traditional model
0%	37.8	40.3
5%	35.2	34.5
10%	32.7	30.5
15%	30.6	27.6
20%	28.4	25.7

TABLE. XII Packet-loss performance comparison.

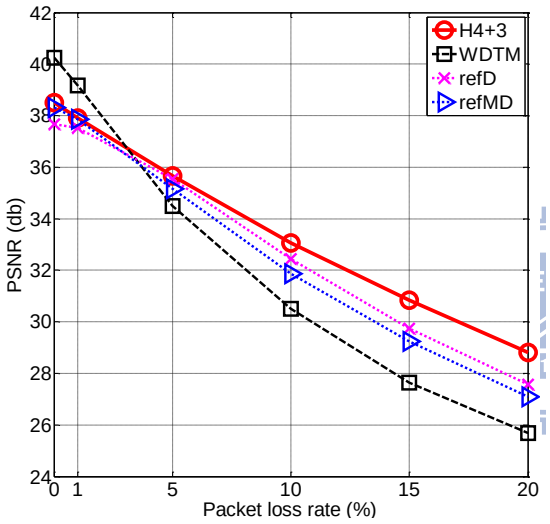
Sequence	Loss rate	Sync1				Sync2				Equal			
		H_{4+1}	H_{4+2}	H_{4+3}	H_{4+4}	H_{4+1}	H_{4+2}	H_{4+3}	H_{4+4}	H_{4+1}	H_{4+2}	H_{4+3}	H_{4+4}
Coastguard @1600kbps	5%	33.87	33.84	**34.02**	33.83	33.67	33.58	33.66	33.65	33.31	33.27	33.66	33.83
	20%	28.69	28.93	**29.12**	29.09	28.13	28.35	28.38	28.46	27.33	27.79	28.38	29.09
Flower @2000kbps	5%	33.52	33.58	**33.80**	33.20	33.26	33.27	33.28	32.82	32.72	32.94	33.28	33.20
	20%	26.38	26.86	**26.98**	26.57	25.73	26.15	26.14	25.69	24.79	25.58	26.14	26.57
Foreman @800kbps	5%	35.29	35.57	**35.67**	35.22	35.10	35.31	35.14	34.88	34.82	35.09	35.14	35.22
	20%	28.80	**29.52**	28.80	28.43	28.15	28.88	28.12	27.56	27.51	28.29	28.12	28.43
News @400kbps	5%	38.70	38.61	**38.94**	38.82	38.55	38.35	38.67	38.66	38.30	38.24	38.67	38.82
	20%	34.04	33.84	34.18	**34.20**	33.50	33.22	33.57	33.56	32.78	33.01	33.57	34.20



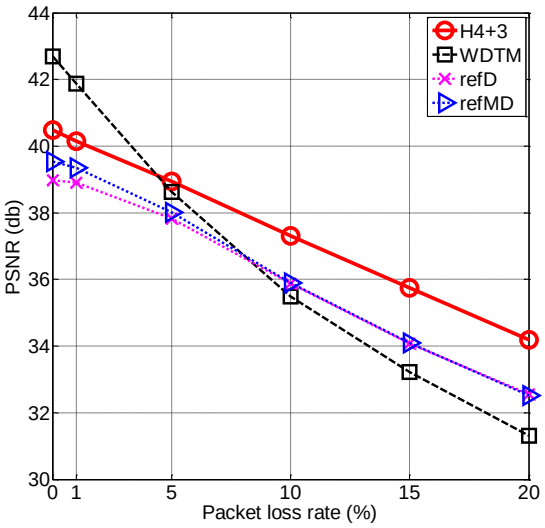
(a) Coastguard (CIF@1600kbps)



(b) Flower (CIF@2000kbps)



(c) Foreman (CIF@800kbps)



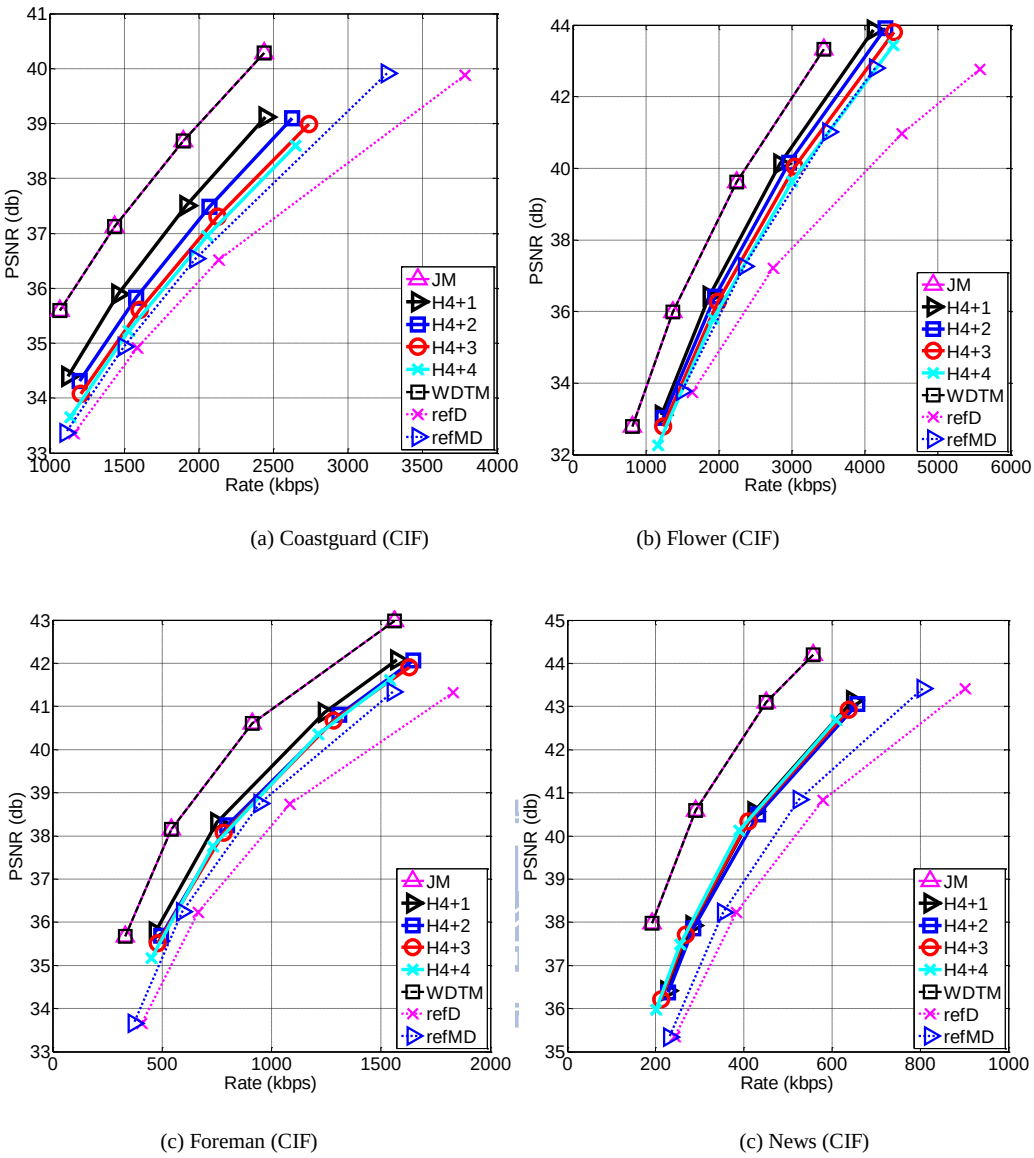
(c) News (CIF@400kbps)

Fig. 42 Packet-loss performance of four hybrid models.

3.2.5.4 Error Free Performance

This section examines the error-free performance of all the methods and the results are presented in Fig. 43. It is observed that the eight curves in Fig. 43 can be divided into three groups: WTDM and JM have the best rate-distortion performance; defaultQP and modifiedQP have the worst performance; and the four hybrid models have the performance in between them. TABLE. XIII shows the bitrate redundancy produced by each method. The bitrate redundancy is defined as the Bjontegaard delta bitrate between JM and each method, which is calculated by the method in [55]. In Fig. 43, WTDM has performance the same to JM16.0 because it focused its error

concealment approach on missing motion recovery and did not modify hierarchical B picture coding structure. Thus, it did not produce any bit-rate redundancy that may reduce rate-distortion (RD) performance in Fig. 43. Both defaultQP and modifiedQP have large bit-rate redundancy that degrades their performance in Fig.8. As shown in Table V, defaultQP produces the redundancy 65%~90%. Compared with defaultQP, while modifiedQP reduces the redundancy about 20% by modifying the QPs of NRB frames, the RD performance improvement as shown in Fig. 43 is quite limited. Compared with modifiedQP, the proposed hybrid models have much lower redundancy as shown in TABLE. XIII and much better RD performance than defaultQP and modifiedQP as shown in Fig. 43.



(a) Coastguard (CIF)

(b) Flower (CIF)

(c) Foreman (CIF)

(c) News (CIF)

Fig. 43 Rate-distortion performance comparison in error free environment.

TABLE. XIII The bit-rate redundancy comparison. The redundancy is defined as the Bjontegarrd bit-rate difference[55] between JM and each method.

Bit-rate redundancy (%)							
Sequence	WTDM	RefD	RefMD	H_{4+1}	H_{4+2}	H_{4+3}	H_{4+4}
coastguard	0	65.1	48.8	24.7	35.4	43.3	47.3
flower	0	68.8	39.6	23.9	28.9	34.3	38.3
foreman	0	76.9	53.9	32.3	41.9	43.7	44.0
news	0	91.1	73.4	46.1	50.4	46.8	44.8

3.3 Rate-Distortion Optimized Mode Selection Method for Multiple Description Video Coding

Multiple description coding is a technique that encodes a single video stream into two or more equally important sub-streams, called descriptions, each of which can be decoded independently. Different from the traditional single description coding (SDC) where the entire video stream (single description) is sent in one channel, in MDC, these multiple descriptions are sent to the destination through different channels, resulting in much less probability of losing the entire video stream (all the descriptions), where the packet losses of all the channels are assumed to be independently and identically distributed. The first MD video coder, called multiple description scalar quantizer (MDSQ)[81], has been realized in 1993 by Vaishampayan who proposed an index assignment table that maps a quantized coefficient into two indices each could be coded with fewer bits. Due to effectiveness in providing error resilience, a variety of research on different MDC approaches had been proposed afterwards. These approaches can be intuitively classified through the stage where it split the signal, such as, frequency domain[81][82], spatial domain [83][83][84], and temporal domain[85][86]. In our previous works [87][87], a hybrid MDC method has been proposed, which applies MDC first in spatial domain to split motion compensated residual data, and then in frequency domain to split quantized coefficients. A hybrid MDC method with spatial and temporal splitting was proposed in [88] and a hierarchical B-picture based hybrid MDC method was proposed in [89][89]. The results in [87][88][89] show that, by properly utilizing more than one splitting technique, the hybrid MDC method can improve error-resilient performance.

To improve coding performance, some researchers proposed to optimize the encoding coefficient for rate-distortion performance. This concept has been adopt in many studies [90]. For MDC, in [91], a R-D optimization technique is proposed for the MDC with one descriptor containing all DCT coefficients and the second one containing only few low frequency coefficients. The R-D technique aims at optimizing the number of pruning coefficients. In [92], the method to find out optimized quantization parameters was proposed for the MDC based on H.264/AVC

redundant slices [93]. Then, Lin et al.[94] extended the method from the slice level to the macroblock level.

There are two major benefits of the rate-distortion optimization concept. First, video contents vary spatially and temporally, so it would be inefficient to use a fixed encoding method to encode whole contents. In addition, the importance of different parts of video contents may be different, so adopting an unequal error protection can achieve better rate-distortion performance. Second, the channel condition also varies over time, so a mechanism to dynamically adjust protection level is necessary. With rate-distortion optimization, the encoder can change coding strategy according to video contents and channel conditions, and therefore improve the performance. However, the previous optimization frameworks were based on the specific MDC systems. Since a variety of new MDC coding tools are being proposed and each tool has different characteristics. To enable the rate-distortion optimization concept on these MDC tools, a general framework is desirable. Therefore, this paper aims at proposing a general optimization framework. The proposed framework is suitable to most coding tools and not restrict to coding structures, such as IPPP or hierarchical B-picture structure. This allows ones to easily integrate their proposed coding tools into the optimization framework and achieve better performance.

The remainder of this paper is organized as follows. First, the MDC system [89] for applying the proposed optimization framework is presented in section II. Section III introduces the proposed framework, and section IV verifies it with simulation data. Section V concludes the paper by summarizing the main results, and discussing possible future work.

3.3.1 Proposed MDC based on a hierarchical B-picture structure

This paper proposes a general R-D optimization framework for MDC systems. To illustrate and evaluate the proposed framework, the MDC system in [89] is adopted, although our optimization approach is not restricted to this specific MDC method. The adopted MDC is a complex system with a wide choice of splitters on a hierarchical B-picture coding structure. With the illustration of applying our approach to this complex MDC system, one can easily apply it to relatively simple MDC

systems. The details of the adopt MDC system is described in the following, and the proposed R-D optimization framework is described in the section III.

3.3.1.1 The encoder architecture

Fig. 44 shows the encoder architecture of the proposed MDC system. The architecture contains three MDC coding tools: duplicator, spatial splitter, and temporal splitter. The three tools divide a SDC bitstream into two MDC descriptors with different amount of redundancy on each. The proposed architecture is similar to the one in [89] except that the mode selection module is added. To encode a frame, the mode selection module analyzes the importance of a macroblock in the frame and the channel condition and then chooses a suitable splitter for the macroblock, thereby optimizing R-D performance. After determining the coding tool, each macroblock is split and encoded into two individual descriptors.

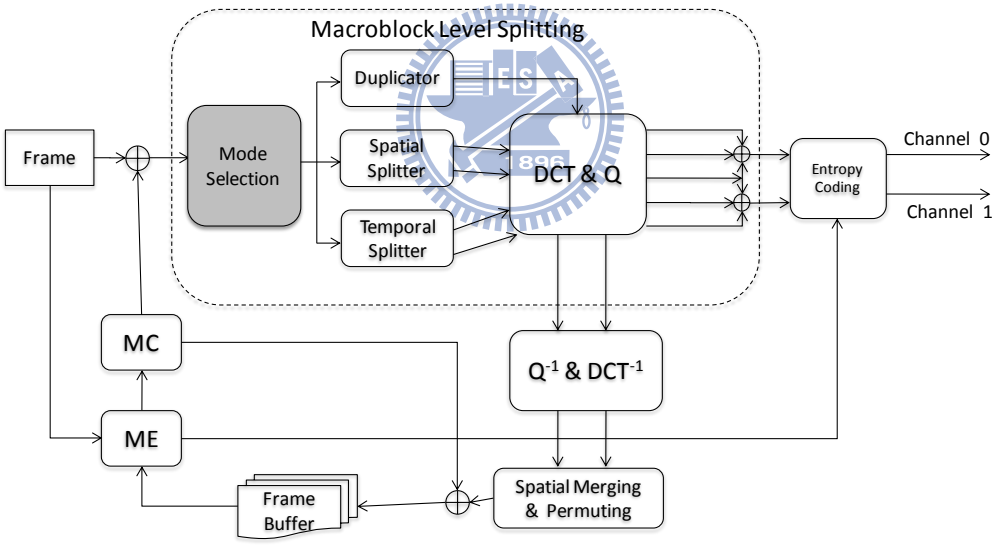


Fig. 44 The encoder architecture of the proposed MDC system.

The system contains three MDC coding tools: duplicator, temporal splitter, and spatial splitter. The duplicator generates twodescriptors by directly duplicating the SDC data into each descriptor. Because each descriptor contains complete SDC data, the decoder can perfectly reconstruct the image as long as any one descriptor is received.

The temporal splitter splits the SDC bitstream in temporal domain, which assigns input frames, in turn, to the two output paths such that successive frames will go to

different descriptors. In other word, if any one descriptor is lost, frames belonging to the lost descriptor are completely lost and could only be estimated by the frames in the other descriptor.

Spatial splitter splits each input macroblock into two parts which are then separately transformed, quantized, and entropy encoded before going to their respective descriptors. The spatial splitter performs splitting on an 8x8 block basis in residual domain. For each 8x8 residual block, it is first polyphase permuted inside the block and then is split to two, as shown in Fig. 45. The permuting mechanism is that, for every 2x2 pixels inside the 8x8 residual block, the top-left pixel (labeled 0) is re-arranged to the top-left 4x4 block, the top-right pixel (labeled 1) to the top-right 4x4 block, the bottom-left pixel (labeled 2) to the bottom-left 4x4 block, and the bottom-right pixel (labeled 3) to the bottom-right 4x4 block, as illustrated in the middle of Fig. 45. After polyphase permutation, the 8x8 block is split into two 8x8 blocks, each carries two 4x4 blocks chosen in diagonal and the remaining two 4x4 blocks are given all-zero residuals (labeled as 'x' in Fig. 45). Note that there are four 8x8 residual blocks in each macroblock, all of them are permuted and split in the same way. Since these split frames need to be merged to serve as reference frames, a Spatial Merger is applied after de-quantization (Q^{-1}) and inverse transform (DCT^{-1}) as shown in Fig. 44. The Spatial Merger first discards the all-zero 4x4 blocks and then adopts Polyphase Inverse Permuting (the reversed process of Fig. 440) to reconstruct the original 8x8 blocks.

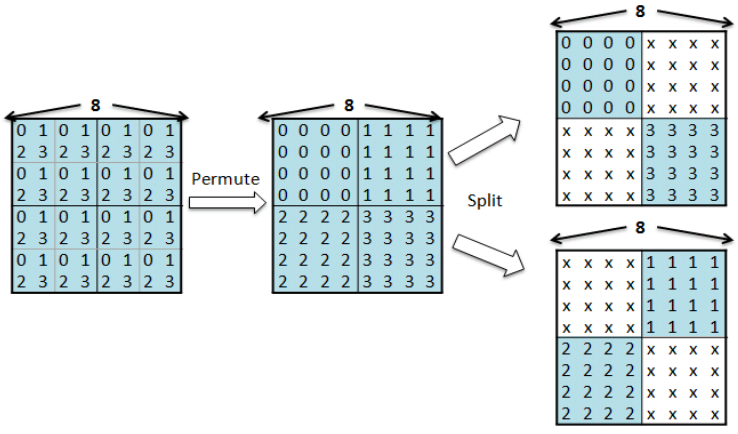


Fig. 45 Spatial splitting of the proposed MDC.

The proposed MDC system is based on a non-dyadic hierarchical B-picture coding structure with 4 levels as depicted in Fig. 46. For the same structure, the MDC

in [89] applies duplicator on the I/P frames at the lowest hierarchical level for providing the highest error resilience, spatial-splitter (S) on the reference B frames at intermediate levels for modest error resilience, and temporal-splitter (T) on the non-reference B frames at the highest level for the lowest error resilience. The rationale behind the assignment in [89] is that the frames at the lower hierarchical level are more important and thus should be protected with more redundancy. In this paper, we extend the idea from frame level to macroblock level. In other word, we adeptly choose the splitters macroblock by macroblock according to its importance. A macroblock in the non-reference B frames at the highest level could be split by the temporal splitter or the duplicator; while a macroblock in other frames could be split by the spatial splitter or the duplicator. The proposed mode selection module plays a role to find out a splitter assignment that has better R-D performance.

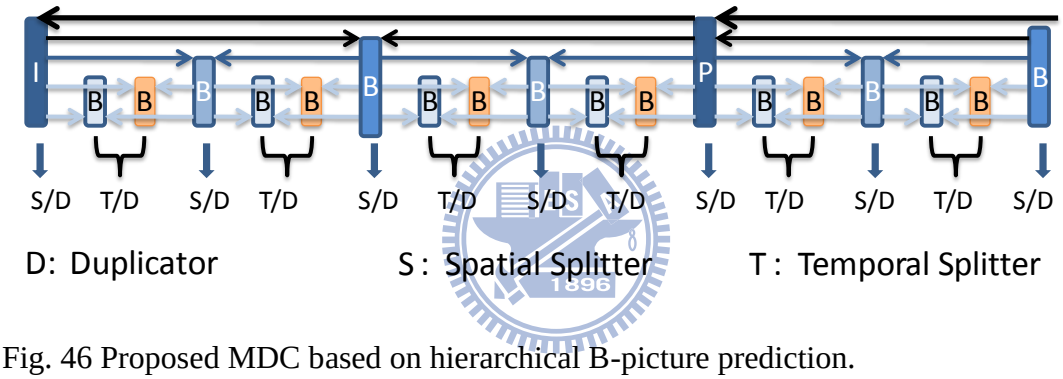


Fig. 46 Proposed MDC based on hierarchical B-picture prediction.

3.3.1.2 The decoder estimation methods

With the proposed MDC, assume the generated two descriptors are denoted by D0 and D1, respectively. Assuming one description, D0, is lost, the macroblocks split by duplicator can be easily reconstructed at decoder by using the same macroblocks in the other description, D1. For the macroblocks split by the spatial splitter, one descriptor loss will cause partially loss of the macroblocks, which can be estimated by using the information of their counterparts in D1. As for the macroblocks split by the temporal splitter, one descriptor loss will cause loss of all the macroblocks in a frame, which can only be estimated by using other frames. In case of two-description loss, D0 and D1, it will result in whole-frame loss regardless of splitter types. For whole-frame loss, each macroblock is recovered based on temporal correlation. TABLE. XIV summarizes the cases for different estimation methods to be applied,

where S denotes the spatial method, T the temporal method, and D the duplication method. The columns describe the two loss cases; while the rows describe three types of splitters. Since the estimation methods are not the focus of this paper, we simply adopt the spatial estimation and the temporal estimation methods in [89] for our experiments in the later section.

TABLE. XIV Summary of the cases for different estimation methods.

Estimation Methods		Descriptor Status	
		One-descriptor Loss	Two-descriptor Loss
Splitter Type	Duplicator	D	T
	Spatial Splitter	S	T
	Temporal Splitter	T	T

3.3.2 Rate-Distortion mode selection method

A MDC system might contain lots of coding tools and have a complex coding structure. How to find out the mode assignment which has good R-D performance is a challenging problem. This paper proposes a R-D optimization framework. With the framework, encoder can decide a suitable splitter mode for each macroblock, thereby optimize the R-D performance. In following, we first explain the proposed framework on an ideal MDC channel. Then, the framework is extended to a packet loss channel. Finally, we summarize the proposed framework.

3.3.2.1 Rate-Distortion optimization on an ideal MDC channel

An ideal MDC channel assumes that some descriptors are received without losing any information while the others are totally lost. Such a situation is referred to as side reconstruction. In the MDC system with two descriptors, e.g. the system introduced in section. II, there are two cases of side reconstruction.

Assume a video is encoded by traditional close-loop codec, and the resulting coding rate and distortion are R_{SDC} and D_{SDC}, respectively. A MDC system tries to

divide the SDC data into two MDC descriptors. First, consider a naive design as a baseline design: the system that directly duplicates the whole SDC data into two descriptors, which is denoted as duplicator-only-MDC (DO-MDC). In this system, the bit-rate of each descriptor, say R_1 and R_2, is equal to R_{SDC}. And, the distortion of side decoders are equal to D_{SDC}.

For DO-MDC which has two cases of side reconstruction, the average distortion of the two side decoders and the total bit-rate of the two descriptors are calculated as

$$D_{\mathrm{Side,DO-MDC}} = (D_1 + D_2)/2 = D_{\mathrm{SDC}}$$
$$R_{\mathrm{Side,DO-MDC}} = R_1 + R_2 = 2 \times R_{\mathrm{SDC}}. \quad (3.3.1)$$

When multiple MDC splitters are available, the encoder can choose a different splitter, instead of the duplicator, to split macroblocks. If the encoder well chooses the splitters for each macroblock, the overall R-D performance would be improved. A challenging R-D optimization problem is that how to find out a good splitter assignment. Assume the encoder choose a mode assignment (say $\boldsymbol{M}$) for all macroblocks in the sequence and the changes of the resulting distortion and bitrate, compared with DO-MDC, are denoted as $(\Delta D_{\mathrm{Side},\boldsymbol{M}}, \Delta R_{\mathrm{Side},\boldsymbol{M}})$. Then, the new distortion and bitrate are:

$$D_{\mathrm{Side,RDO}} = D_{\mathrm{Side,DO-MDC}} + \Delta D_{\mathrm{Side},\boldsymbol{M}},$$
$$R_{\mathrm{Side,RDO}} = R_{\mathrm{Side,DO-MDC}} + \Delta R_{\mathrm{Side},\boldsymbol{M}}. \quad (3.3.2)$$

The R-D optimization problem is to find out the $\boldsymbol{M}$ for better $(D_{\mathrm{Side,RDO}}, R_{\mathrm{Side,RDO}})$. To solve the problem, we propose a strategy that makes $(\Delta D_{\mathrm{Side},\boldsymbol{M}}, \Delta R_{\mathrm{Side},\boldsymbol{M}})$ satisfy the equation:

$$\frac{\mathrm{d}(\Delta D_{\mathrm{Side},\boldsymbol{M}})}{\mathrm{d}(\Delta R_{\mathrm{Side},\boldsymbol{M}})} = \frac{\mathrm{d}(D_{\mathrm{Side,\ DO-MDC}})}{\mathrm{d}(R_{\mathrm{Side,\ DO-MDC}})} = \frac{1}{2} \times \frac{\mathrm{d}(D_{\mathrm{SDC}})}{\mathrm{d}(R_{\mathrm{SDC}})}. \quad (3.3.3)$$

In Eq. (3.3.3), the first two terms represent the slope of the R-D curve, which means the ratio of distortion improvement over bitrate consumption. Larger ratio indicates that increasing little bitrate can improve distortion greatly. If we try to divide bitrate resource into two targets as Eq. (3.3.2), the best strategy is to keep the slopes of two targets the same. Otherwise, we can easily move rates from the target with the small slope to the target with the large one, and the overall R-D performance will thereby improved. Eq. (3) expresses this concept.

To better understand the proposed method's characteristic on R-D performance, we take an example in Fig. 47 to illustrate the concept of Eq. (3.3.3). The Foreman CIF sequence is encoded by a MDC system and its R-D curve is shown in Fig. 47, where there are four R-D points, **A**, **B**, **C**, and **D**. In the right-down legend, the bitrates of four R-D points are shown in the form of $R_{\mathrm{Side,DO-MDC}} + \Delta R_{\mathrm{Side},\boldsymbol{M}}$. Points **A** and **B** are the R-D points of DO-MDC, where only the duplicator is adopted, so $\Delta R_{\mathrm{Side},\boldsymbol{M}}$ equals to zero. With other splitters adopted to replace the duplicator for some macroblocks, the R-D points move along the dashed curve from point **A** to **C**. Keeping adopting the splitters for more macroblocks, the R-D curve will go to point **D**. For the R-D curve in Fig. 47, it is observed that point **C** has the best R-D performance and that the bitrate allocated to $\Delta R_{\mathrm{Side},\boldsymbol{M}}$ is too small for point **A** and too large for point **D**. Since different splitting-mode assignments will result in different R-D performances, Eq. (3.3.3) provides a guide to select a good splitting-mode assignment.

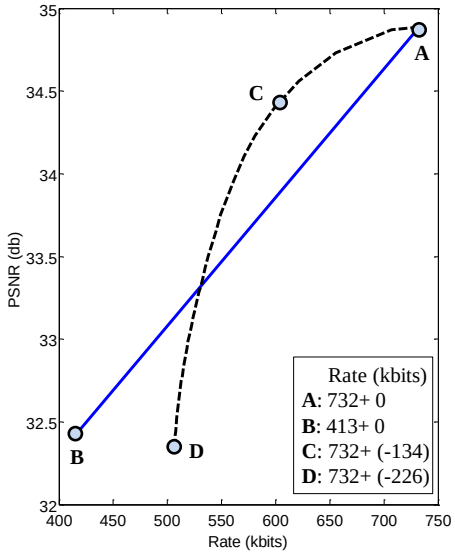


Fig. 47 An example of R-D optimization.

According to the concept in Eq.(3.3.3), a splitting-mode selection method is proposed. For macroblock *i*, the encoder firstly encodes it by DO-MDC and then try each splitter candidate. For each splitter, calculate the bitrate and distortion changes from using DO-MDC and then choose the one closest to Eq. (3.3.3).

In the proposed mode selection method, the encoder should calculate the R-D impact for each splitter candidate. However, accurate R-D impact is hard to calculate,

because the distortion will propagate among frames according to traditional predictive coding scheme. For each splitter candidate applied on a macroblock, all frames that directly or indirectly reference to this macroblock should be re-encoded to calculate the distortion propagation and then the exact R-D change can be obtained. However, the computation is too complex and is not realistic. In following, we proposed a realistic method to estimate the R-D impact of each splitter candidate.

3.3.2.2 Rate-Distortion estimation

Compared with DO-MDC, if a macroblock *i* is encoded by a splitter mode *j*, rather than the duplicator, the bitrate change due to this macroblock is denoted by $\Delta R^{\mathrm{MB}\,i}_{\mathrm{Side,mode}\,j}$ and the distortion change is by $\Delta D^{\mathrm{MB}\,i}_{\mathrm{Side,mode}\,j}$. The bitrate change can be calculated as

$$\Delta R^{\mathrm{MB}\,i}_{\mathrm{Side,mode}\,j} = R^{\mathrm{MB}\,i}_{\mathrm{Side,DO-MDC}} - R^{\mathrm{MB}\,i}_{\mathrm{Side,mode}\,j} \ . \tag{3.3.4}$$

The distortion change, $\Delta D^{\mathrm{MB}\,i}_{\mathrm{Side,mode}\,j}$, however, is hard to be calculated because it needs to take into account all the affected macroblocks caused by motion prediction which results in distortion propagation. To reduce the complexity of distortion calculation, an estimation method is proposed as Eq.(3.3.5), where each pixel has a *distortion weight*, *w*, to approximate the distortion from the pixel itself and the propagation effect.

$$\Delta D^{\mathrm{MB}\,i}_{\mathrm{Side,mode}\,j} = \sum_{k\in \mathrm{MB}_i} w_k \times (d^{\mathrm{pxl}\,k}_{\mathrm{Side,DO-MDC}} - d^{\mathrm{pxl}\,k}_{\mathrm{Side,mode}\,j}) \ , \tag{3.3.5}$$

where $(d^{\mathrm{pxl}\,k}_{\mathrm{Side,DO-MDC}} - d^{\mathrm{pxl}\,k}_{\mathrm{Side,mode}\,j})$ is the distortion change of pixel *k* by replacing the duplicator with a splitter mode *j* on macroblock *i*. Note that uncapitalized "*d*" represents distortion of pixel *k* itself. In contrast, capitalized "*D*" represents distortion superimposed on the entire sequence, including the distortion on macroblok *i* itself and the distortion propagating to other macroblocks. In Eq.(3.3.5), if there is no propagation effect, distortion weight of each pixel will be equal to one. With propagation effects, the distortion weight is approximated by a linear model which sequentially estimates the propagated distortion of each pixel from the trajectory of motion prediction. Since distortion propagation is caused by motion prediction, the amount of propagated error should be larger if a pixel is referred by more pixels, namely, its distortion weight *w* should be set larger. According to this concept, we calculate *w* from the motion prediction trajectory.

Although similar idea has been proposed in [94], there are two major differences between their approach and ours. First, we adopt pixel-level instead of macroblock-level estimation. Second, we consider that the propagated distortion will decay over time[97][98] and thus adopt a linear model for this effect.

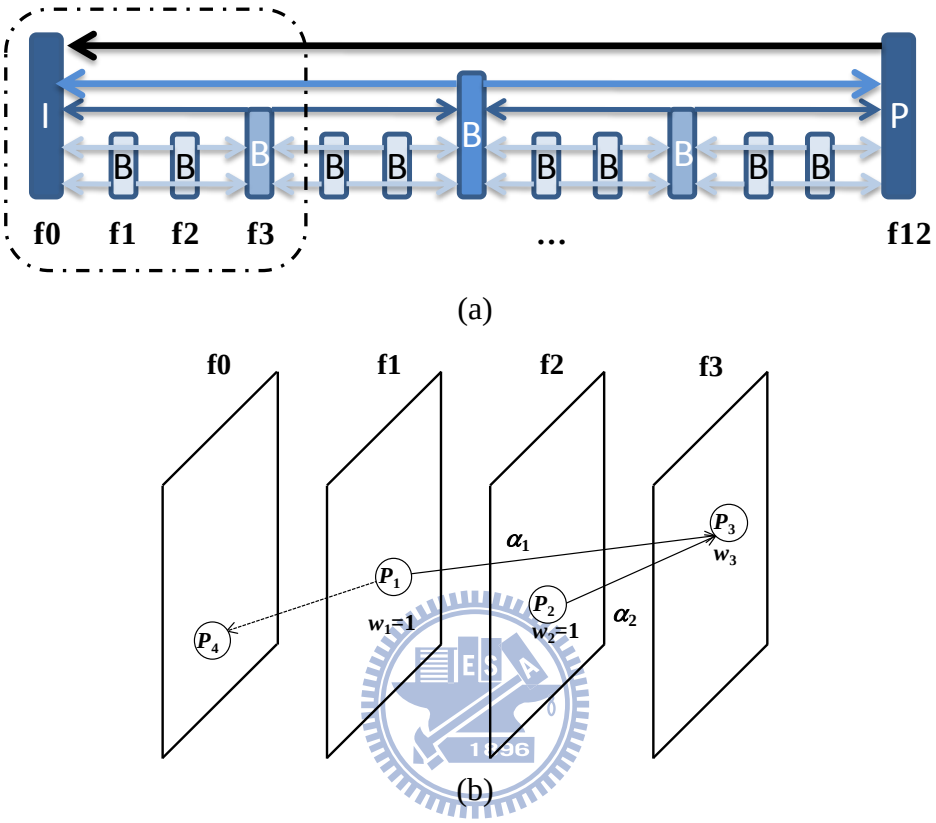


Fig. 48 Illustration of Error Weight.

Take an example in Fig. 48 to illustrate how to calculate distortion weights. Fig. 48(a) shows successive frames in a hierarchical B coding architecture, where the arrow signs indict the directions of motion prediction. We enlarge the first four frames in Fig. 48(b) and highlight four pixels, P_1, P_2, P_3 and P_4, to explain the method of w_k calculation. Since P_1 and P_2 are in non-reference frames, their distortion will not propagate to other frames and thus the corresponding weights, w_1 and w_2, both equal to 1. Assuming that P_3 is referred by P_1 and P_2, since the distortion of P_3 will propagate to P_1 and P_2, we add some distortion to P_3 to elevate its impact on the overall distortion. In the case of Fig. 48, since P_1 and P_2 are non-reference pixels, the distortion propagated from P_3 will stop on these two pixels. The distortion weight of P_3 can be thereby calculated as $1 + \alpha_1 + \alpha_2$, where 1 represents the distortion of P_3 itself, and α_1 and α_2 represent the distortion propagated to P_1 and P_2, respectively. The values of α depends on motion prediction schemes of P_1 and P_2. In this

example, P_1 is bi-predicted by P_3 and P_4 (0.5* P_1+0.5* P_4); P_2 is uni-predicted by P_3. Many distortion estimation methods [94][99] assume that the distortion will propagate to other pixels without any decay. By this assumption, α_1 and α_2 are 0.5 and 1, respectively. However, some coding tools will mitigate the error propagation effects, e.g., de-blocking filter, sub-pixel interpolation filter, quantizer, and so on. Therefore, we adopt a factor, α_{PD},representing propagation decays and then α_1 and α_2 become $0.5 \times \alpha_{\mathrm{PD}}$ and α_{PD}, respectively. Some studies[98] have proposed theoretical derivation of propagation decays. In our approach, the decay factor α_{PD} is statistically determined by experiments. In the experiments, we introduced little error in a frame and observed the propagated errors in those frames that refer to this frame. The factor, α_{PD}, can be thereby calculated. To conduct the experiments, four CIF sequences, *Coastguard*, *Hall*, *Harbour*, and *Soccer* were adopted and encoded by hierarchical B picture structures with QPs equal to 16, 22, 28, and 34, respectively. We introduced errors into frames on each hierarchical layer and observed the propagated error. The experimental results are shown in Fig. 49, where the vertical axis is the observed decay factors and the horizontal axis is QP settings. It can be seen that the results of four sequences can be approached by Eq.(3.3.6), a linear function of decay factor and QP, using least square method.

$$\alpha_{\mathrm{PD}} = 0.0032 \times \mathrm{QP} + 0.7466. \tag{3.3.6}$$

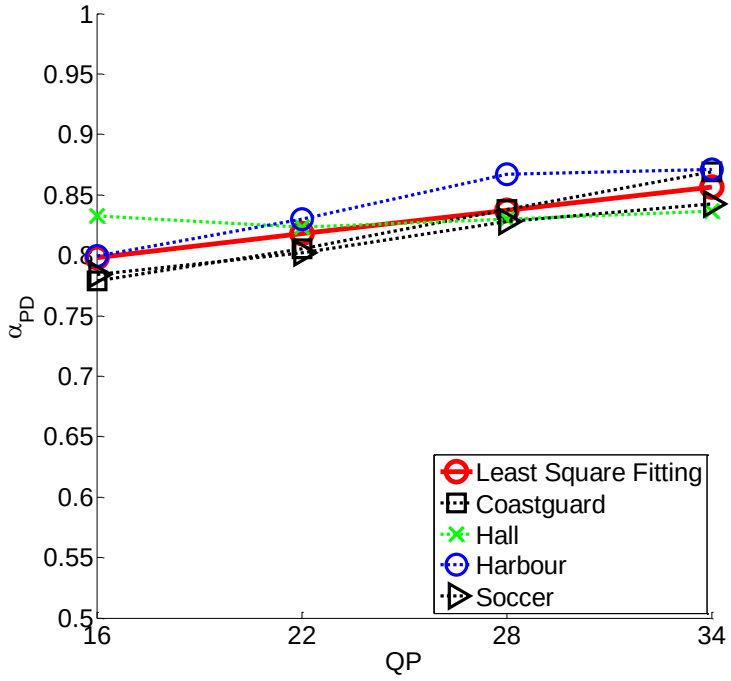


Fig. 49 Fitting result of propagation decays factors, α_{PD}.

In the example of Fig. 48, if P_1 and P_2 are also referred by other pixels, then the w_1 and w_2 will not equal to 1. The distortion of P_3 will propagate not only to P_1 and P_2 but also to the pixels referring to them. The distortion weight of P_3 will be the sum of the distortion weights of P_1 and P_2, i.e. $w_3 = 1 + 0.5 \times \alpha_{\mathrm{PD}} \times w_1 + 1 \times \alpha_{\mathrm{PD}} \times w_2$.

To summarize, the distortion weight of pixel k is

$$w_k = \begin{cases} 1, & \text{if } k \text{ is a non-reference pixel} \\ \sum_{l \in \boldsymbol{\Omega}_k} \alpha_l \ w_l, & \text{if } k \text{ is a reference pixel} \end{cases}, \tag{3.3.7}$$

where $\boldsymbol{\Omega}_k$ is the set of the pixels referring to pixel k and α_l represents the distortion propagation factor which can be calculated as

$$\alpha_l = \begin{cases} \alpha_{\mathrm{PD}}, & \text{if } l \text{ is an uni-predicted pixel} \\ 0.5 \times \alpha_{\mathrm{PD}}, & \text{if } l \text{ is a bi-predicted pixel} \end{cases}, \tag{3.3.8}$$

where α_{PD} is calculated by Eq.(3.3.6). To determine the best mode assignment, we start from non-reference frames to all the reference frames in the same GOP, so the distortion weights of all pixels in the GOP can be derived from Eq.(3.3.7). And then the bit-rate and distortion impact of each mode on each individual macroblock can be calculated by Eq. (3.3.4) and Eq. (3.3.5), respectively. Finally, the best mode

assignment for each macroblock can be found by Eq. (3.3.3). The proposed mode selection method is summarized in section III.D.

3.3.2.3 Rate-Distortion optimization on a packet loss channel

In section III.A, the proposed mode selection method is discussed in an ideal MDC channel. In following, we will extend it to a general packet loss channel.

Assume a frame is divided into two descriptors. Each descriptor forms a packet and is transmitted through a packet loss network. In the decoder side, the frame can be perfectly reconstructed if two descriptors are received. If any description loss, the data will be recovered by the estimation method proposed in section II. For a macroblock MB_i, let $D_{2D}^{MB\,i}$ denote the distortion superimposed on the whole sequence when two descriptions are received, and $D_{1D}^{MB\,i}$ and $D_{0D}^{MB\,i}$ when one and no descriptor is received, respectively. Note that, for a macroblock, the distortion superimposed on the sequence includes the distortion caused by itself and the distortion propagated to other macroblocks in the sequence.

Given packet loss rate, P_l, the expectation of the distortion caused by MB_i is derived as

$$D_{P_l}^{MB\,i} = (1-P_l)^2 \cdot D_{2D}^{MB\,i} + 2(1-P_l)P_l \cdot D_{1D}^{MB\,i} + {P_l}^2 \cdot D_{0D}^{MB\,i}. \quad (3.3.9)$$

The last part of Eq. (3.3.9) can be neglected for low P_l. Assuming that the distortion caused by the loss of a number of macroblocks is mutually un-correlated[94], the expectation of their loss can be evaluated as

$$D_{P_l} = \sum_i D_{P_l}^{MB\,i} = (1-P_l)^2 \cdot \left(\sum_i D_{2D}^{MB\,i}\right) + 2(1-P_l)P_l \cdot \left(\sum_i D_{1D}^{MB\,i}\right). \quad (3.3.10)$$

To see how Eq.(3.3.10) is affected by mode assignment, we firstly consider RO-MDC where the distortion when one or two descriptors are received is equal to the distortion of SDC, namly, $\sum_i D_{2D,DO-MDC}^{MB\,i} = \sum_i D_{1D,DO-MDC}^{MB\,i} = D_{SDC}$. When two descriptors are received, since all information distributed into descriptors are collected on the decoder side without any loss, we assume $D_{2D}^{MB\,i}$ would not change. The mode assignment will result in distortion change only when there is any description loss. Let $\Delta D_{P_l,\mathbf{M}}$ denote the distortion change when assignment **M** is applied and one description is lost. With mode assignment, Eq. (3.3.10) will be re-written as

$$D_{P_l,\mathrm{RDO}} = (1-P_l)^2 \times D_{\mathrm{SDC}} + 2(1-P_l)P_l \times (D_{\mathrm{SDC}} +\Delta\, D_{P_l,\mathbf{M}})$$
$$= \{(1-P_l)^2 + 2(1-P_l)P_l\} \times D_{\mathrm{SDC}}^{\mathrm{Seq}} + 2(1-P_l)P_l \times\Delta\, D_{P_l,\mathbf{M}}. \quad (3.3.11)$$

On the other hand, the bit-rate taking account for mode assignment **M** is

$$R_{P_l,\mathrm{RDO}} = 2 \times R_{\mathrm{SDC}} +\Delta\, R_{P_l,\mathbf{M}} \quad (3.3.12)$$

According to Eq. (3), the assignment **M** should satisfy

$$\frac{2(1-P_l)P_l}{1} \times \frac{\mathrm{d}(\Delta D_{P_l,\mathbf{M}})}{\mathrm{d}(\Delta R_{P_l,\mathbf{M}})} = \frac{\{(1-P_l)^2+2(1-P_l)P_l\}}{2} \times \frac{\mathrm{d}D_{\mathrm{SDC}}}{\mathrm{d}R_{\mathrm{SDC}}} \quad (3.3.13)$$

which can be rewritten as

$$\frac{\mathrm{d}(\Delta D_{P_l,\mathbf{M}})}{\mathrm{d}(\Delta R_{P_l,\mathbf{M}})} = \frac{\{(1-P_l)^2+2(1-P_l)P_l\}}{2\times 2(1-P_l)P_l} \times \frac{\mathrm{d}D_{\mathrm{SDC}}}{\mathrm{d}R_{\mathrm{SDC}}} \ . \quad (3.3.14)$$

Using Eq. (3.3.14) instead of Eq. (3.3.3), the best assignment **M** under packet loss network can be found using the method proposed in section III.A.

3.3.2.4 Summary of proposed Rate-Distortion mode selection method

Let *N*, *I*, and *P* respectively denote GOP length, the number of macroblocks in one frame, and the number of pixels in one frame. Λ( ) is a function, which indicates the frame encoding order. The proposed mode selection method is shown in the following:

---

*/*Step1: Record R-D performance and motion prediction trajectory */*
For frame *n* = Λ(1) to Λ(*N*) in a GOP
  For macroblock *i* = 1 to *I* in the frame *n*
    Encode macroblock *i* by SDC codec.
    Record $R_{\mathrm{SDC}}^{\mathrm{MB}\,i}$ and $D_{\mathrm{SDC}}^{\mathrm{MB}\,i}$.
    Record the motion vectors.
  end
end
/ Step2: Calculate distortion weights */*
For frame *n* = Λ(*N*) to Λ(1) in a GOP.
  For pixel *p* = 1 to *P* in the frame n
    Calculate distortion weights of pixel *p* by Eq.(3.3.7).
  end
end
/ Step3: optimize R-D performance*/*
For frame *n* = Λ(1) to Λ(*N*) in a GOP
  For macroblock *i* = 1 to *I* in the frame n
    Calculate $\Delta D_{\mathrm{mode\,j}}^{\mathrm{MB}\,i}$ and $\Delta R_{\mathrm{mode\,j}}^{\mathrm{MB}\,i}$ by Eq.(3.3.4) and Eq.(3.3.5) for each splitting mode *j*.
    Select the best mode by Eq.(3.3.3) or Eq.(3.3.14).
  end
end

In Eq.(3.3.3) and Eq.(3.3.14), the R-D slope of SDC, $\mathrm{d}(D_{\mathrm{SDC}})/\mathrm{d}(R_{\mathrm{SDC}})$, is related to adopted SDC codec. For H.264/AVC codec, the slope can be approximated by

$$\frac{\mathrm{d}D_{\mathrm{SDC}}}{\mathrm{d}R_{\mathrm{SDC}}} = \beta \times 2^{\left(\frac{\mathrm{QP}-12}{3}\right)}, \tag{3.3.15}$$

where β is empirically fitted as -0.85 in [95][96]. However, this value is not good enough for the proposed system. To clarify this, experiments have been conducted to find a better β for our framework. We choose four CIF versions of sequences, Coastguard, Hall, Harbour, and Soccer and encode them with different combinations of QPs (22, 25, 28, and 31) and packet loss rates (10%, 20%, 30%, 40%, and 50%). For each packet loss rate, we calculate mode assignments by using ten values of β, equally distributed from 0 to 1. Among these ten values, the one with best R-D performance by B-D method is selected. The best β value selected for each packet loss rate is shown in Fig. 50. It can be found that when packet loss rate increases, the optimal β value increases. We adopt a linear model to fit the relation between β and packet loss rates. The least square fitting result is:

$$\beta = 1.04P_l\text{-}0.67 \tag{3.3.16}$$

Even though the data do not exactly distributed linearly, we found that the fitting error is not sensitive. Since simple linear model can provide acceptable performance, we adopt linear fitting results to conduct the following experiments.

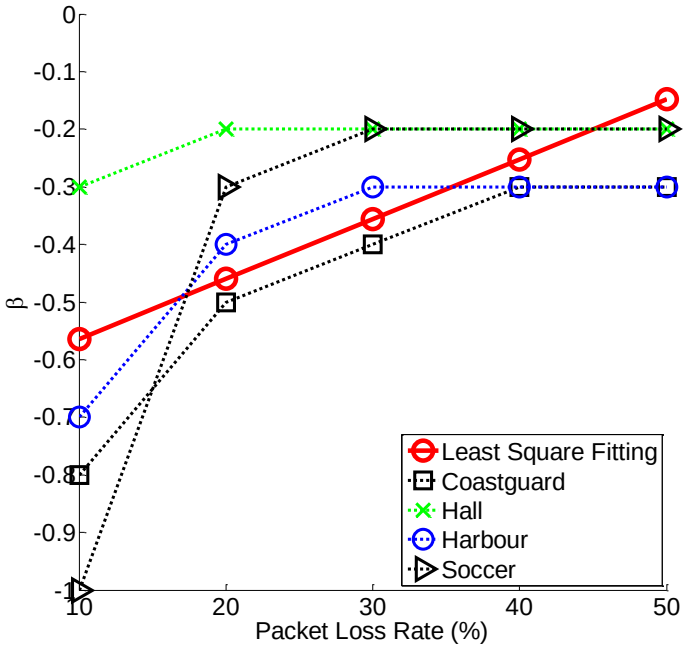


Fig. 50 Fitting result of β in Eq. (15).

3.3.3 Experimental Result

In this section, the performance of the proposed mode selection method was evaluated under both packet loss channels and ideal MDC channels.

3.3.3.1Packet Loss Performance

For conducting the experiments, four CIF versions of sequences, Foreman, News, Stefan, and Table Tennis, were chosen. We select these sequences because they contain different types of contents. Note that, for fair comparison, these sequences are different from those sequences used for the coefficient fitting described in section III. All sequences were encoded using a dyadic hierarchical structure with 4 levels. For the optimized encoding, it is better to set smaller QPs for the frames that are referenced by other frames. In the Joint Scalable Video Model 11 (JSVM11)[100], QPs of the B frames at level-1 equal to the QPs of the I/P frames plus 4, and the QPs at level-i increase by 1 from level-($i-1$), with $i \geq 2$.

Three MDC systems were adopted for performance evaluation. They are the proposed method, the MDC system in [89], and the DO-MDC. Each of the three MDC systems generated two descriptors and transmitted them through packet loss channels. Four packet loss rates, 1%, 5%, 10%, and 20%, were chosen for evaluation.

The resulting R-D curves were shown in Fig. 51 to Fig. 54. Bjontegarrd bit rate savings (BD-rate) and PSNR gains (BD- PSNR) are calculated using the methodology presented in [101] and shown in TABLE. XV. In all experiments, the proposed method has the best performance. Compared with the MDC system in [89], the proposed method has significant improvement when packet loss rate is low (0%~10%). As the packet loss rate increases (10%~20%), the proposed method still performs better, although the improvement becomes moderate. However, if packet loss rate further increases, resulting in one descriptor is totally lost, the performance gap between the proposed method and the MDC in [89] will be turned to increase again, which is presented in the next subsection. Compared with the MDC in [89], since the proposed method can adjust error resilience ability according to channel conditions, the R-D performance can be optimized for various packet loss rates, resulting in better performance than the MDC in [89] for every loss rate. Comparing with DO-MDC, the proposed method performed better with the performance gap even

larger. The reason is that DO-MDC allocated too much redundancy for the channel with low error rates. Although the performance gap decreases as the packet loss rate increases, especially when one descriptor is totally lost which is presented in the next subsection, the overall results still show the superiority of the proposed method over DO-MDC.

TABLE. XV BD results of the proposed framework on packet loss channels. The column of "Comparing with the MDC system in [89]" shows the BD difference between the proposed method and the MDC system in [89]; The column of "Comparing with DO-MDC" shows the difference between the proposed method and DO-MDC.

Sequence	P_l	Comparing with the MDC system in [89]		Comparing with DO-MDC	
		BD-PSNR(db)	BD-Rate(%)	BD-PSNR(db)	BD-Rate(%)
Foreman	1%	0.457	-9.352	1.345	-26.209
	5%	0.385	-8.329	1.040	-21.554
	10%	0.372	-8.493	0.783	-17.307
	20%	0.376	-10.136	0.418	-11.158
News	1%	0.232	-3.964	1.579	-24.132
	5%	0.174	-3.102	1.340	-21.257
	10%	0.127	-2.387	1.080	-18.082
	20%	0.105	-2.352	0.628	-12.548
Stefan	1%	0.682	-11.070	2.018	-32.057
	5%	0.482	-8.710	1.396	-24.501
	10%	0.401	-8.059	0.864	-16.848
	20%	0.398	-9.567	0.243	-5.945
Table Tennis	1%	0.430	-8.151	1.485	-26.541
	5%	0.293	-5.873	1.110	-21.131
	10%	0.188	-4.044	0.743	-15.177
	20%	0.109	-2.758	0.299	-7.172

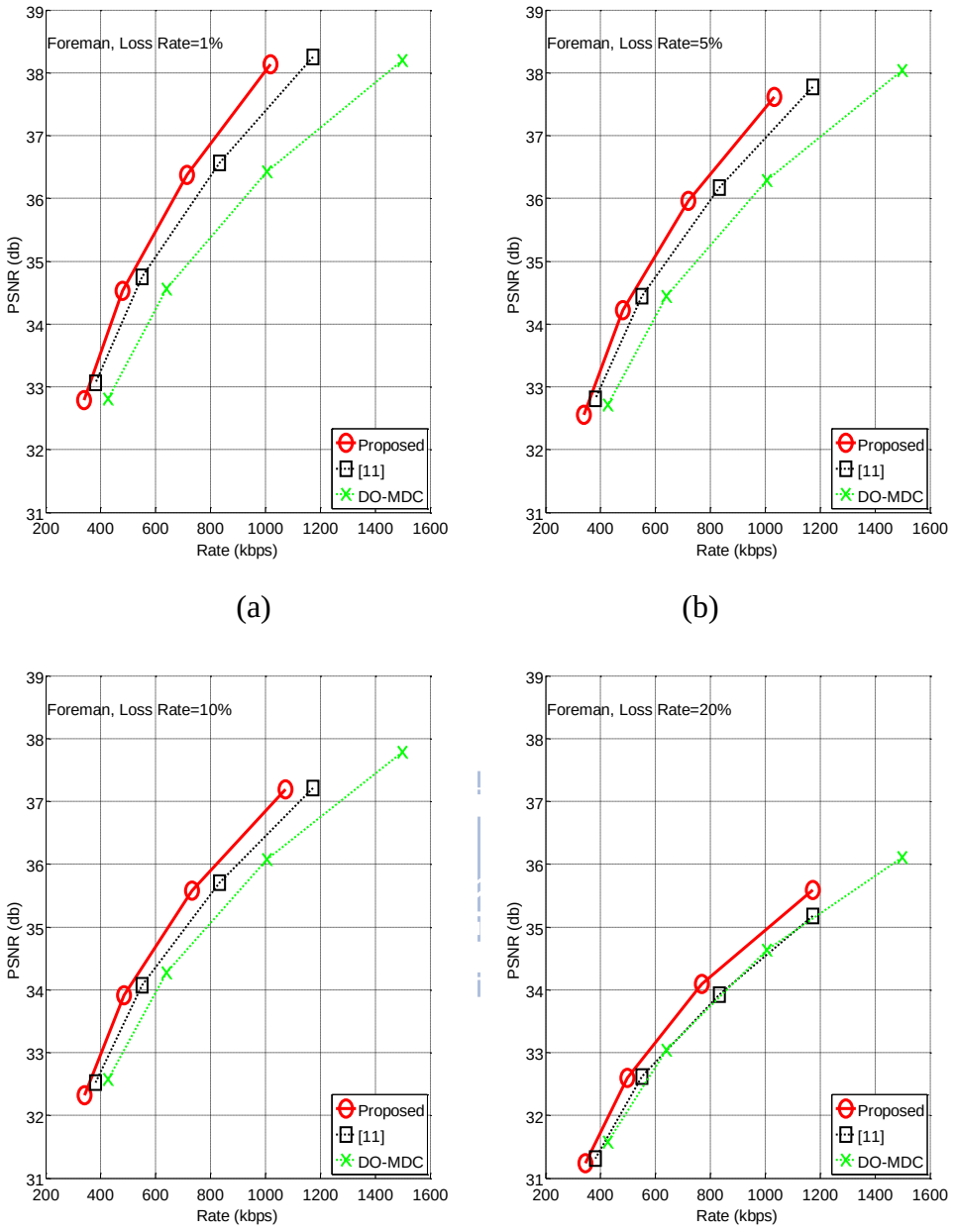


Fig. 51 R-D performance of the Forman Sequence. (a) Packet loss rate = 1%. (b) Packet loss rate = 5%. (c) Packet loss rate = 10%. (d) Packet loss rate = 20%.

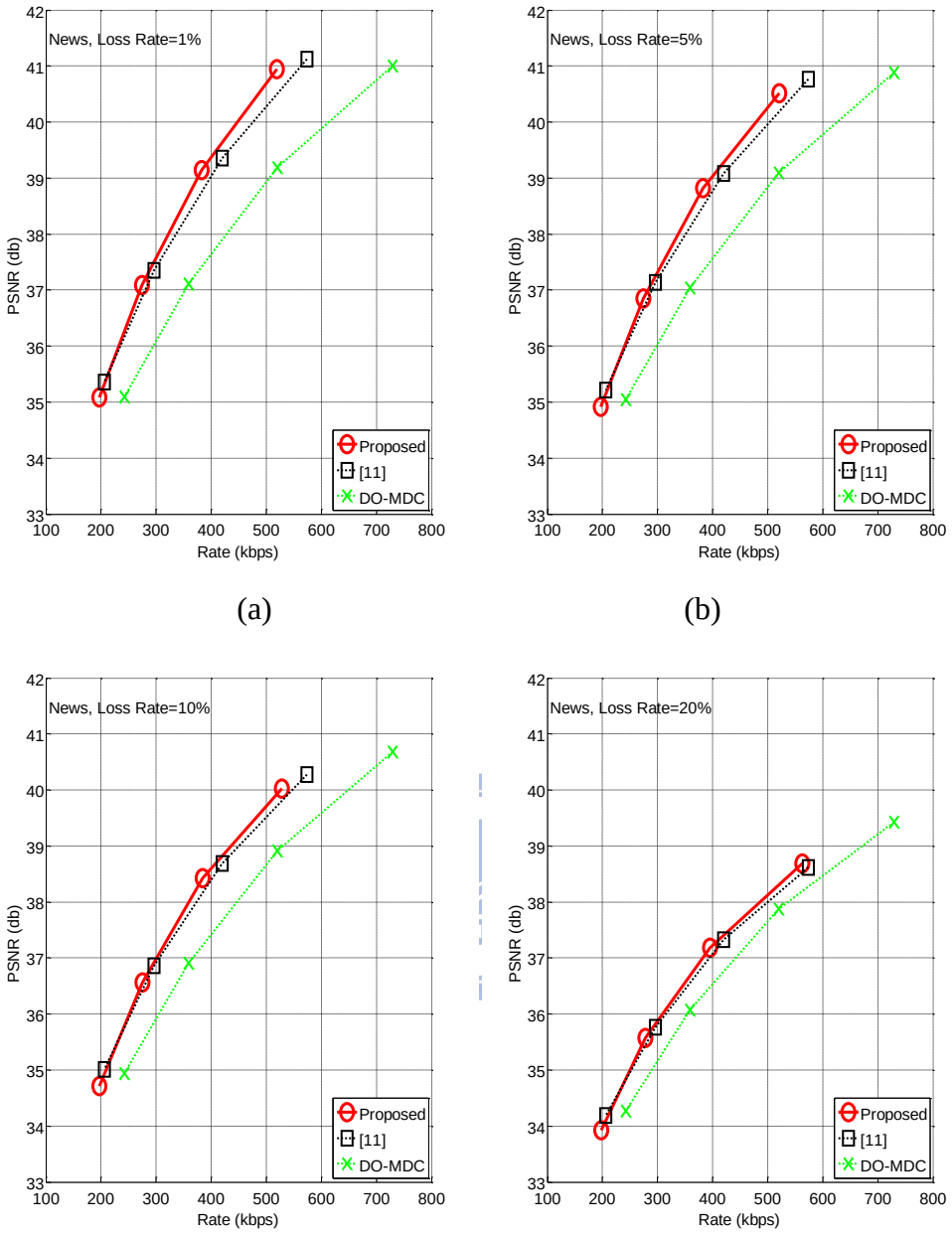


Fig. 52 R-D performance of the News Sequence. (a) Packet loss rate = 1%. (b) Packet loss rate = 5%. (c) Packet loss rate = 10%. (d) Packet loss rate = 20%.

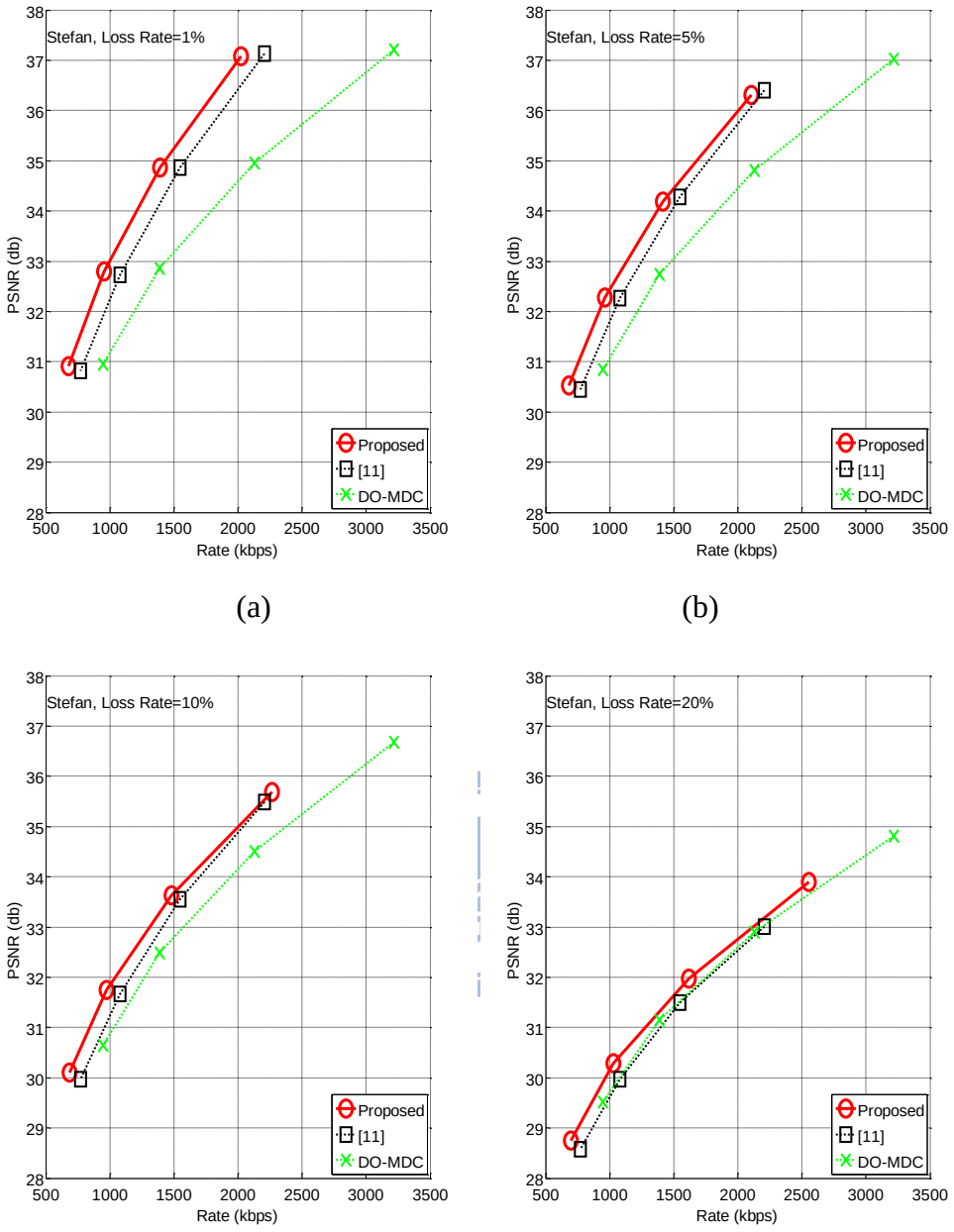


Fig. 53 R-D performance of the Stefan Sequence. (a) Packet loss rate = 1%. (b) Packet loss rate = 5%. (c) Packet loss rate = 10%. (d) Packet loss rate = 20%.

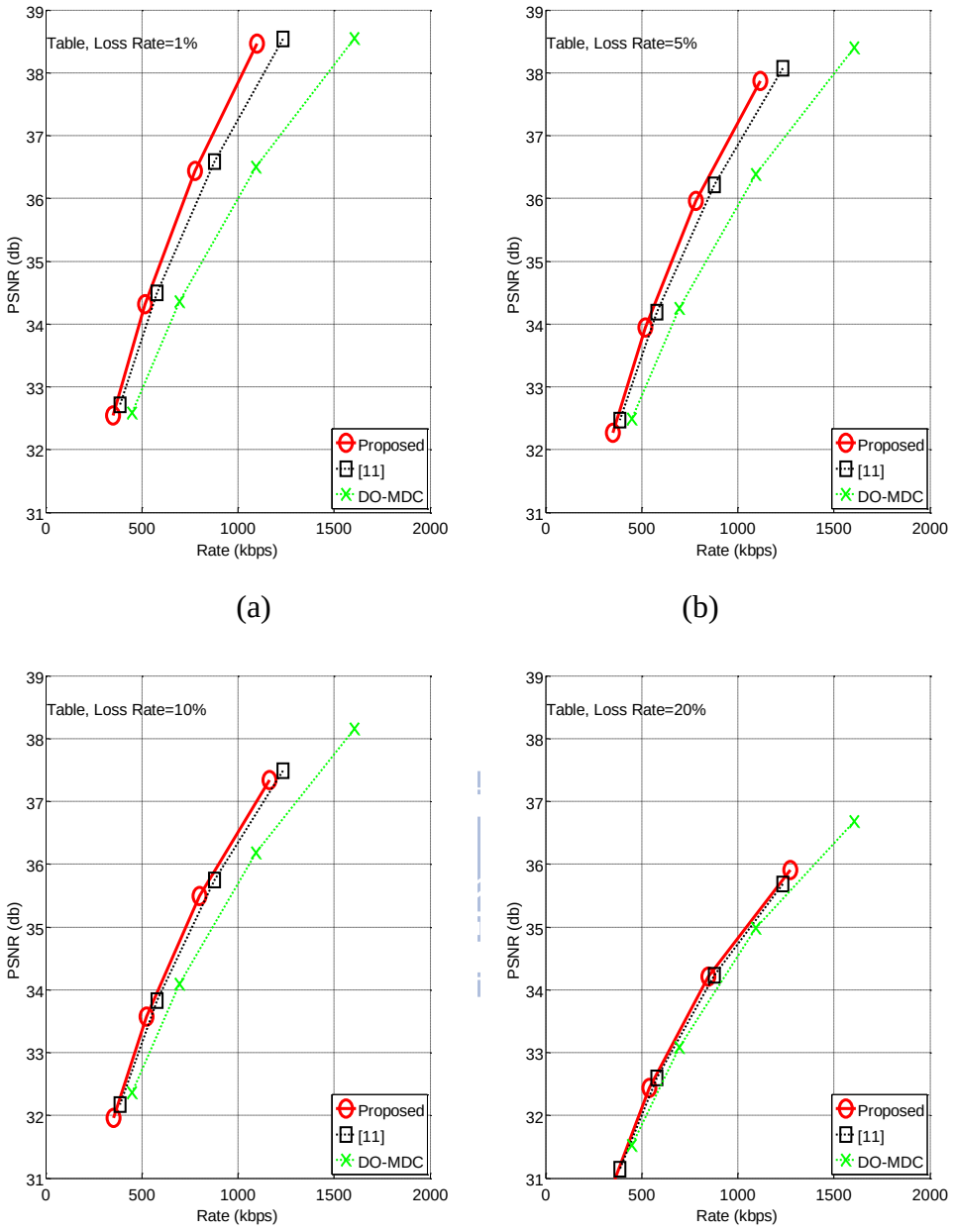


Fig. 54 R-D performance of the Table Tennis Sequence. (a) Packet loss rate = 1%. (b) Packet loss rate = 5%. (c) Packet loss rate = 10%. (d) Packet loss rate = 20%.

3.3.3.2 Side Reconstruction Performance

In following, we evaluated the performance of the proposed method on ideal MDC channels which means that one descriptor is received without losing any informance while the other is totally lost. Such performance is called side reconstruction performance and the results were shown in TABLE. XVI and Fig. 55. It can be found that the proposed method has the best performance. Comparing with the MDC system in [89], the performance improvement can be up to 3.7dB. This is due to that the MDC in [89] adopted fixed redundancy assignment and hence is only suitable for a certain range of packet loss rates. When the loss rate comes to 50% (one descriptor is lost), it is obviously that the redundancy is insufficient to reconstruct well. The proposed method, however, determines the mode assignment taking into account for channel conditions, and thus performs better. Compared with DO-MDC, the proposed method still has better performance but the improvement is little. The reason might be that the splitting methods adopt in this paper are not good enough. If some advanced MDC tools could be adopt in the system in the future, the performance improvement might increase.

We also showed the performance of center decoding in TABLE. XVII and Fig. 56. When error amount equals zero, the value of Eq. (3.3.14) goes to negative infinity. Therefore, the optimization framework would remove redundancy as much as possible and the proposed method thereby has the best R-D performance.

TABLE. XVI Side decoding BD results of the proposed framework. The column of "Comparing with the MDC system in [89]" shows the BD difference between the proposed method and the MDC system in [89]; The column of "Comparing with DO-MDC" shows the difference between the proposed method and DO-MDC.

Sequence	Comparing with the MDC system in [89]		Comparing with DO-MDC	
	BD-PSNR(db)	BD-Rate(%)	BD-PSNR(db)	BD-Rate(%)
Foreman	1.973	-42.235	0.166	-3.872
News	0.942	-17.252	0.178	-3.350
Stefan	3.668	-61.209	0.092	-1.826
Table Tennis	1.847	-37.359	0.044	-0.954

TABLE. XVII Center decoding BD results of the proposed framework. The column of "Comparing with the MDC system in [89]" shows the BD difference between the proposed method and the MDC system in [89]; The column of "Comparing with DO-MDC" shows the difference between the proposed method and DO-MDC.

Sequence	Comparing with the MDC system in [89]		Comparing with DO-MDC	
	BD-PSNR(db)	BD-Rate(%)	BD-PSNR(db)	BD-Rate(%)
Foreman	0.656	-12.557	1.379	-26.398
News	0.469	-7.518	1.617	-24.381
Stefan	0.646	-10.398	2.120	-32.928
Table Tennis	0.587	-10.551	1.547	-27.126

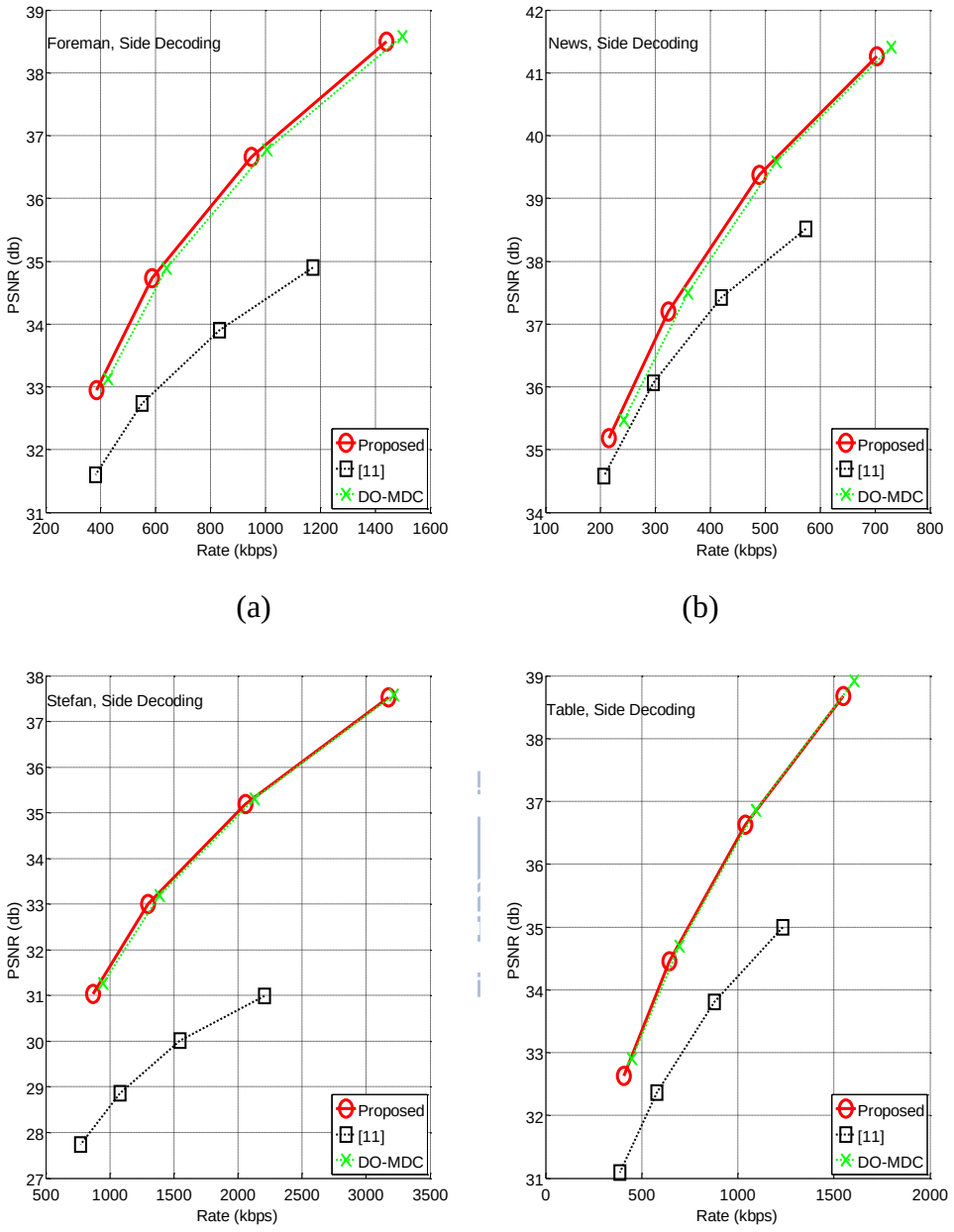


Fig. 55 Side decoding R-D performance. (a) Foreman. (b) News. (c) Stefan. (d) Table Tennis.

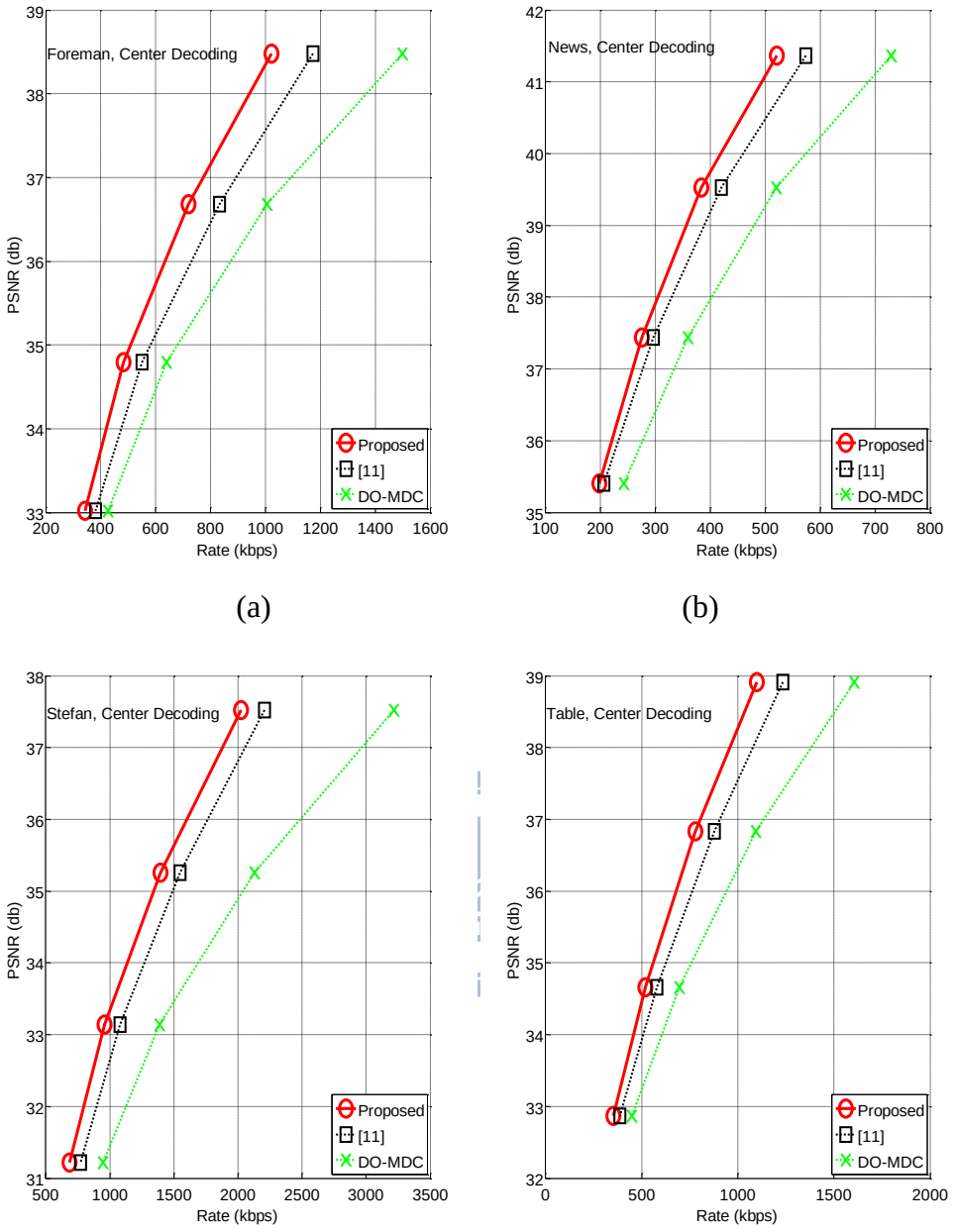


Fig. 56 Center decoding R-D performance. (a) Foreman. (b) News. (c) Stefan. (d) Table Tennis.

Chapter 4 Conclusion and Future Works

4.1 Comments on Distributed Video Coding

For distributed video coding, a new coding tool, adaptive macroblock grouping for WZ coding, is proposed. The tool uses an SI error estimator to adaptively group macroblocks of similar error characteristics into same WZ coding block such that the SI error correction efficiency can be improved. In the proposed framework, video macroblocks in a Wyner-Ziv frame are classified into different groups based on the estimated prediction error of the side information. The information is transmitted via uplink channel back to the encoder so that macroblocks with similar error statistics can be grouped together into same coding blocks for channel coding. Experimental results show that the bit rate can be reduced about 5%-10% by grouping source data.

In addition, we propose a perceptual-based WZ coding technique for DVC codecs. In the proposed framework, the decoder estimates the visual distortion levels of SI macroblocks and marks the macroblocks that require WZ reconstruction as the macroblocks in the region-of-interest (ROI). The ROI information is then transmitted back to the encoder using a bitmap so that parity bits can be generated to correct only these macroblocks. Experimental results show that the proposed perceptual-based coding technique improves coding efficiency of DVC both subjectively and objectively.

Although the proposed technique works well for video sequences that have distinctive regions of interest, it does not detect all the visually distorted regions for sequences with multiple complex moving objects. For example, for the QCIF Soccer sequence, the proposed technique misses 128 macroblocks (out of 7326) when adaptive thresholds of θMC and θTSC are used. Although we can select fixed threshold values such that all visually distorted blocks in Soccer are included into the ROI, the size of ROI will become large (contains 62% of macroblocks on average)

and makes the proposed techniques less coding effective. More sophisticated visual distortion detection techniques will be investigated to deal with complex scenes such as the Soccer sequence.

Currently, the proposed technique has been tested using DVC coding structure of GOP size equals two. When GOP size becomes larger, both the SI prediction errors and the discrepancy between true and estimated motion fields will become large. As a result, the majority of the macroblocks will be included into the ROI. If larger GOP size is to be used, a more sophisticated SI generation algorithm has to be used to maintain efficiency of the proposed framework. For example, in current implementation, we only use the texture distribution map for perceptual-based analysis. It is possible to also use the texture map to constrain the motion-projection algorithm so that predicted SI and motion field are closer to the original WZ frame and true motion fields, respectively.

Finally, in the proposed framework, LDPCA is used for WZ reconstruction in the ROI. Since the size of the ROI is only composed of 20 ~ 30% of macroblocks in a frame, the LDPCA coding efficiency may suffer due to short coding block length. More detail analysis on the error characteristics of ROI macroblocks will be conducted in the future for the design of a more efficient WZ reconstruction algorithm.

4.2 Comments on Robust Video Coding

In this thesis, a RDO-based error resilient scheme using MRF-MCP is presented, which employs the nearest error-resilient frames (i.e., *ER-frame*) as part of the reference frames and adopts ER-RDO for reference block selection. With ER-RDO, the choice of blocks predicting from ER-frames will be adaptive to various channel conditions and video sequences. Besides, this paper also presents some techniques to reduce the time complexity of the proposed scheme. The experimental results show that, with these techniques, the computational cost can be reduced dramatically with neglectable performance loss.

In addition, a hybrid model based on hierarchical B pictures is proposed, which improves error concealment effects by combining two hierarchical B-picture coding structures. For a four-level hierarchical structure, there are four variations of the proposed hybrid model. They are H_{4+1}, H_{4+2}, H_{4+3} and H_{4+4}. In H_{4+1} model, each

base-level key frame has a buddy frame which is used to serve as the data recovery frame when it is lost. In H_{4+2} and H_{4+3}, not only key-frames, but also RB-frames have buddy frames. In H_{4+4}, all the frames, including NRB-frames, have buddy frames. With buddy frames, data recovery distance can be reduced and the error concealment performance can be substantially improved. Experiments have been conducted for eight methods: four variations of the proposed model (H_{4+1}, H_{4+2}, H_{4+3}, and H_{4+4}), WTDM [7], two methods (defaultQP and modifiedQP) in [78], and JM16.0. The experimental results show that the proposed H_{4+3} has the overall best performance among them.

Finally, we also propose a rate-distortion optimization framework for MDC systems. With the proposed framework, the encoder can dynamically adjust coding strategy according to both video contents and channel conditions. Experimental results show that the proposed optimization framework improves coding efficiency significantly.

Although the proposed technique can optimize coding strategy for different channel conditions, the improvement is moderated in the channels with large error rates. This might be due to the MDC tools adopted in this paper are not good enough to deal with these channels well. If more MDC tools can be adopted in the proposed framework, it is possible to further improve R-D performance in the channels with large errors. Based on the proposed results, more detail analysis on designing splitters capable of handling the channels with large errors will be conducted in the future for the design of a more efficient MDC tool.

Reference

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

[2] L. Liu, Z. Li, and E.J. Delp, "Efficient and Low-Complexity Surveillance Video Compression Using Backward-Channel Aware Wyner-Ziv Video Coding," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 19, no. 4, pp. 453-465, Apr. 2009.

[3] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. on Inform. Theory*, vol. 19, no.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. Inform. Theory*, vol. 22, no.1, pp. 1–10, Jan. 1976.

[5] A. Nafaa, T. Taleb, and L. Murphy, "Forward error correction strategies for media streaming over wireless networks," *IEEE Commun. Mag.*, vol. 46, no.1, pp. 72–79, Jan. 2008.

[6] R. Zhang and S. Regunathan and K. Rose, "Video coding with optimal inter/intra-mode switching for packet loss resilience," *IEEE J. Sel. Areas Commun.*, vol. 18, no.6, pp. 966-976, Jun. 2000.

[7] X. Ji, D. Zhao, and W. Gao, "Concealment of Whole-Picture Loss in Hierarchical B-Picture Scalable Video Coding," *IEEE Trans. on Multimedia*, vol.11, no.1, pp. 11-22, Jan. 2009.

[8] C. W. Hsiao and W. J. Tsai, "Hybrid multiple description coding based on H.264," *IEEE Trans. on Circuits and Systems for Video Technology*, vol.20, no.1, pp.76-87, Jan. 2010.

[9] W. J. Tsai and J. Y. Chen "Joint temporal and spatial error concealment for multiple description video coding," *IEEE Trans. on Circuits and Systems for Video Technology*, vol.20, no.12, pp.1822-1833, Dec. 2010.

[10] R. Puri, K. Ramchandran, "PRISM: a Reversed Multimedia Coding Paradigm," *Proc. of IEEE Int. Conf. on Image Processing*, 2003.

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

[12] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, M. Ouaret, "The DISCOVER codec: Architecture, Techniques and Evaluation," *Proc. of Picture Coding Symposium 2007*, Lisbon, Portugal.

[13] J. Ascenso and F. Pereira, "Advanced Side Information Creation Techniques and

Framework for Wyner-Ziv Video Coding", *Journal of Visual Communication and Image Representation*, vol. 19, no. 8, pp. 600-613, Dec. 2008.

[14] D. Varodayan, D. Chen, M. Flierl, and B. Girod, "Wyner-Ziv coding of video with unsupervised motion vector learning," *EURASIP Signal Processing: Image Communication Journal*, vol. 23, no. 5, pp. 369-378, June 2008.

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

[16] D. Varodayan, A. Aaron, and B. Girod, "Rate-daptive Distributed Source Coding using Low-Density Parity-Check Codes,"*EURASIP Signal Processing Journal*, vol. 86, pp. 3123-3130, Nov. 2006.

[17] J. Ascenso and F. Pereira, "Adaptive hash-based side information exploitation for efficient Wyner-Ziv video coding," *Proc. of IEEE. Int. Conf. on Image Processing*, 2007.

[18] C. Brites and F. Pereira, "Correlation Noise Modeling for Efficient Pixel and Transform Domain Wyner–Ziv Video Coding," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 18, issue 9, pp. 1177-1190, 2008.

[19] X. Huang and S. Forchhammer, "Improved virtual channel noise model for transform domain Wyner-Ziv video coding" *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2009.

[20] J. Ascenso, C. Brites, and F. Pereira, "Content Adaptive Wyner-ZIV Video Coding Driven by Motion Activity," *Proc. of IEEE Int. Conf. on Image Processing*, 2006.

[21] J. Zhang, H. Li, Q. Liu, and C. Chen, "A Transform Domain Classification Based Wyner-Ziv Video Codec," *Proc. of IEEE Int. Conf. on Multimedia and Expo*, 2007

[22] D. Tsai, C. Lee, and W. Lie, "Dynamic Key Block Decision with Spatio-Temporal Analysis for Wyner-Ziv Video Coding," *Proc. of IEEE Int. Conf. on Image Processing*, 2007.

[23] G. Ding, "Wyner-Ziv Video Coding with Part Intracoding," *Proc. of Fourth Int. Conf. on Image and Graphics*, 2007.

[24] J. Park, B. Jeon, D. Wang, and A. Vincent, "Wyner-Ziv video coding with region adaptive quantization and progressive channel noise modeling" *Proc. of IEEE Int. Symposium on Broadband Multimedia Systems and Broadcasting*, 2009.

[25] Y. Sun and C. Tsai, "Low Complexity Motion Model Analysis for Distributed Video Coding," *Proc. of Int. Symposium on Multimedia over Wireless*, 2008

[26] Y. Sun, S. Lian and C. Tsai, "Prioritized Side Information Correction for Distributed Video Coding," *Proc. of Picture Coding Symposium*, 2009

[27] E. Steinbach, N. Farber, and B. Girod, “Standard compatible extension of H.263 for robust video transmission in mobile environments,” *IEEE Trans. on Circuits System for Video Technology*, vol. 7, pp. 872-881, Dec. 1997.
[28] T. Stockhammer and D. Kontopodis and T. Wieg, “Rate-Distortion Optimization for JVT/H.26L Video Coding in Packet Loss Environment,” *Proc. of Packet Video Workshop*, Jan. 2002.
[29] Y. Zhang, W. Gao, Y. Lu, Q. Huang and D. Zhao, “Joint Source-Channel Rate-Distortion Optimization for H.264 Video Coding Over Error-Prone Networks,” *IEEE Trans. on Multimedia*, vol. 9, pp. 445-454, Apr. 2007.
[30] H. Yang and K. Rose, “Rate-distortion optimized motion estimation for error resilient video coding,” *Proc. of IEEE Int. Conf. on Acoustics*, Jan. 2005.
[31] J. Yang and X. Fang, “Rate-distortion optimized selection of motion vectors for video transmission over packet-loss channels ,” *IEICE Trans. Commun.*, vol. E89-b, no. 12, pp. 3494-3495, Dec. 2006.
[32] H. Yang and J. Boyce, “Concealment-aware motion estimation and mode selection for error resilient video coding,” *Proc. of IEEE Int. Conf. Image Processing*, 2006.
[33] S. Wan and E. Izquierdo, "Rate-Distortion Optimized Motion- Compensated Prediction for Packet Loss Resilient Video Coding," *IEEE Trans. on Image Processing*, vol.16, no.5, pp.1327-1338, May 2007.
[34] J. Zheng and L.-P. Chau, “Error-resilient coding of H.264 based on periodic macroblock,” *IEEE Trans. on Broadcasting*, vol. 52, pp. 223-229, Jun. 2006.
[35] Q. Zhang and G. Liu, “Error resilient coding of H.264 using intact long-term reference frames,” *Proc. of 5th Int. Conf. on Visual Information Engineering*, 2008.
[36] A. Leontaris and P. C. Cosman, "Video compression for lossy packet networks with mode switching and a dual-frame buffer," *IEEE Trans. on Image Processing*, vol. 13, no. 7, pp. 885-897, Jul. 2004.
[37] A. Leontaris and P. C. Cosman, "Optimal mode selection for a pulsed-quality dual frame video coders," *IEEE Signal Processing Letter*, vol. 11, no. 12, pp. 952-955, Dec. 2004.
[38] D. Liu, D. Zhao, X, Ji and W. Gao, "Dual frame motion compensation with optimal long-term reference frame selection and bit allocation," *IEEE Trans. on Circuits System for Video Technology*, vol. 20, no. 3, pp.325-339, March 2010.
[39] Z. Li, L. Liu, and E.J. Delp, “Rate distortion analysis of motion side estimation in Wyner–Ziv video coding,” *IEEE Transactions on Image Processing*, vol.16, no.1, pp. 98–113, Jan. 2007.
[40] J.D. Areia, F. Pereira, and W.A.C. Fernando, “Impact of the key frames quality on the overall Wyner-Ziv video coding performance” *Proc. of 50th Int. Symposium ELMAR*, 2008.
[41] D. Kubasov, F. Lajnef, and C. Guillemot, “A hybrid encoder/decoder rate control for Wyner-Ziv Video Coding with a feedback channel,” *Proc. of IEEE 9th Workshop on Multimedia Signal Processing*, 2007.
[42] C. Brites and F. Pereira, “Encoder rate control for transform domain Wyner-Ziv Video Coding,” *Proc. of IEEE Int. Conf. on Image Processing*, 2007.
[43] A. Roca, M. Morbee, J. Prades-Nebot and, E. J. Delp, “Rate control algorithm

for pixel-domain Wynet-Ziv Video Coding," *Proc. of SPIE Int. Conf. on Visual Communication and Image Processing*, 2008.

[44] A. Aaron, R. Zhang, and B. Girod, "Wyner-Ziv coding of motion video," *Proc. of Asilomar Conf. on Signals and Systems*, 2005.

[45] Y. Cheng, "Mean Shift, Mode Seeking, and Clustering". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.17, no.8, pp. 790-799, 1995.

[46] D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach toward Feature Space Analysis". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.24, no.5, pp.603–619, 2002.

[47] H.264/AVC reference software (JM 16.0).

[48] V. Toto-Zarasoa, A. Roumy, and C. Guillemot, "Hidden Markov model for Distributed Video Coding," *Proc. of Int. Conf. on Image Processing*, 2010.

[49] J.D. Areia, F. Pereira, and W.A.C. Fernando, "Impact of the Key Frames Quality on the Overall Wyner-Ziv Video Coding Performance" *Proc. of 50th Int. Symposium ELMAR*, 2008.

[50] Y.-C. Sun, S.-Y. Lian, and C.-J. Tsai, "Prioritized Side Information Correction for Distributed Video Coding," *Proc. of Picture Coding Symposium*, 2009.

[51] Z. Wang, A. C. Bovik, H. R. Sheikh, "Image Quality Assessment: Form Error Visibility to Structural Similarity," *IEEE Trans. on Image Processing*, Vol. 13, No.4, pp.600-612, Apr., 2004.

[52] R. Zabih, J. Miller and K. Mai, "A feature-based algorithm for detecting and classifying production effects," *Multimedia Systems*, vol. 7, no.2, pp. 119-128, Mar. 1999.

[53] H. S. Song, I. K. Kim and N. I. Cho, "Scene change detection by feature extraction from strong edge blocks," *Proc. of Proceedings of Visual Communications and Image Processing*, 2002.

[54] R. Martins, C. Brites, J. Ascenso, F. Pereira, "Statistical Motion Learning Approach for Improved Transform Domain Wyner-Ziv Video Coding," *IET Image Processing Journal*, vol. 4, no 1, pp. 28 – 41, Feb. 2010.

[55] G. Bjontegaard, "Improvement of the BD-PSNR model," *VCEG document VCEG-AI11*, ITU-T SG16/Q6, 35th VCEG Meeting, July 2008.

[56] D. Kubasov, F. Lajnef, and C. Guillemot, "A Hybrid Encoder/Decoder Rate Control for Wyner-Ziv Video Coding with A Feedback Channel," *Proc. of Int. W. on Multimedia Signal Processing*, 2007.

[57] C. Brites and F. Pereira, "Encoder Rate Control for Transform Domain Wyner-Ziv Video Coding," *Proc. of Int. Conf. on Image Processing*, 2007.

[58] Y.-S. Pai, Y.-C. Shen and J.-L. Wu, "High efficient distributed video coding with parallelized design for LDPCA decoding on CUDA based GPGPU," *Journal of*

Visual Communication and Image Representation, Vol. 23, Issue. 1, pp. 63-74, Jan., 2012.

[59] S. Sofke, F. Pereira, and E. Muller, "Dynamic Quality Control for Transform Domain Wyner-Ziv Video Coding," *EURASIP Journal on Image and Video Processing*, Vol. 2009, pp.1-15, Jan.2009.

[60] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "FSIM: A Feature Similarity Index for Image Quality Assessment," *IEEE Trans. on Image Processing*, vol.20, no.8, pp.2378-2386, Aug. 2011.

[61] Y.-C. Lin, D. Varodayan, and B. Girod, "Image authentication using distributed source coding," *IEEE Trans. on Image Processing*, vol.21, no.1, pp.273-283, Jan. 2012.

[62] T. Sheng, X. Zhu, G. Hua, H. Guo, J. Zhou, and C.-W. Chen, "Feedback-free rate-allocation scheme for transform domain Wyner-Ziv video coding," *Multimedia Systems*, vol. 16, no.2, pp. 127-137, Feb. 2010.

[63] N. Rahnavard, and F. Fekri, "New results on unequal error protection using LDPC codes," *IEEE Communications Letters*, vol. 10, no. 1, pp. 43- 45, Jan. 2006.

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

[65] F. Cen, "Design of degree distributions for LDPCA codes," *IEEE Communications Letters*, Vol. 13, no. 7, pp. 525-527, July 2009.

[66] Y. Zhang, W. Gao, Y. Lu, Q. Huang and D. Zhao, "Joint Source-Channel Rate-Distortion Optimization for H.264 Video Coding Over Error-Prone Networks," *IEEE Trans. on Multimedia*, vol. 9, pp. 445-454, Apr. 2007.

[67] J. Zheng and L.-P. Chau, "Error-resilient coding of H.264 based on periodic macroblock," *IEEE Trans. on Broadcasting*, vol. 52, pp. 223-229, Jun. 2006.

[68] Q. Zhang and G. Liu, "Error resilient coding of H.264 using intact long-term reference frames," *Proc. of 5th Int. Conf. on Visual Information Engineering*, 2008.

[69] S. Yang, D. Kim, Y. Jeon and J Jeong, "An efficient motion re-estimation algorithm for frame-skipping video transcoding," *Proc. of* IEEE International Conference on Image Processing, 2005.

[70] J. Youn and M. T. Sun, "A fast motion vector composition method for temporal transcoding," *Proc. of IEEE International Symposium on Circuits and Systems*, 1999.

[71] S. Wan and E. Izquierdo, "Rate-Distortion Optimized Motion- Compensated Prediction for Packet Loss Resilient Video Coding," *IEEE Trans. on Image*

Processing, vol.16, no.5, pp.1327-1338, May 2007.

[72] H.-Y. Yin, A. M. Tourapis, and J. Boyce, "Fast mode decision and motion estimation for H.264," *IEEE Int'l Conf. on Image Processing*, 2003.

[73] Z. Chen, J. Xu, Y. He, and J. Zheng, "Fastinteger-pel and fractional-pel motion estimation for H.264/AVC," *Journal of Visual Comm. & Image Representation*, vol.17, no.2, pp. 264-290, April 2006.

[74] H. Schwarz, D. Marpe , T. Wiegand, "Analysis of hierarchical B pictures and MTCF," *Proc. of IEEE International Conference on Multimedia & Expo*, 2006.

[75] J. Reichel, H. Schwarz, and M. Wien, "Joint Scalable Video Model 11 (JSVM 11)," Joint Video Team, Doc. JVT-X202, Jul. 2007.

[76] M. Flierl and B. Girod, "Generalized B pictures and the draft H.264/AVC video compression standard," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 587–597, July 2003.

[77] X. Ji, D. Zhao, and W. Gao, "Concealment of Whole-Picture Loss in Hierarchical B-Picture Scalable Video Coding," *IEEE Trans. on Multimedia*, vol.11, no.1, pp. 11-22, Jan. 2009.

[78] C. Zhu and M. Liu, "Multiple description video coding based on hierarchical B pictures," *IEEE Trans. Circuits and Systems for Video Technology*, vol.19, No.4, April 2009.

[79] Y. Wang, A. Reibman, and S. Lin, "Multiple description coding for video delivery," *Proc. of IEEE*, vol. 93, pp. 57–70, Jan. 2005.

[80] C.-S. Lin and W.-T. Syu, "A fine-grained balancing scheme for improved scalability in P2P streaming," *Multimedia Tools and Applications*, vol.46, Issue.1, pp.71-91, Jan. 2012.

[81] V.A. Vaishampayan, "Design of Multiple Description Scalar Quantizers," *IEEE Trans. on Information Theory*, vol. 39, no.3, pp.821-834, May 1993.

[82] O. Campana, R. Contiero, "An H.264/AVC Video Coder Based on Multiple Description Scalar Quantizer," *Proc. of IEEE Asilomar Conference on Signals, Systems and Computers*, 2006

[83] R. Bemardini, M. Durigon, R. Rinaldo, L. Celetto, and A. Vitali, "Polyphase Spatial Subsampling Multiple Description Coding of Video Streams with H.264," *Proc. of IEEE Intel. Conf. on Image Processing*, 2004.

[84] J. Jia and H. K. Kim, "Polyphase Downsampling Based Multiple Description Coding Applied to H.264 Video Coding," *IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E89-A, Issue 6, pp. 1601-1606, June 2006

[85] J. G. Apostolopoulos, "Error-Resilient Video Compression Through the Use of Multiple States," *Proc. of IEEE Intel. Conf. on Image Processing*, 2000.

[86] S. Gao, H. Gharavi, "Multiple Description Video Coding over Multiple Path Routing Networks," *Proc. of Intl. Conf. on Digital Communication Proceedings*, 2006.

[87] C.-W. Hsiao and W.-J. Tsai, "Hybrid Multiple Description Coding Based on H.264," *IEEE Trans. on Circuits and Syst. for Video Technology*, vol. 20, no.1, pp.76-87, Jan. 2012.

[88] W. J. Tsai and J.-Y. Chen, "Joint Temporal and Spatial Error Concealment for Multiple Description Video Coding," *IEEE Trans. on Circuits and Syst. for Video Technology*, pp.1822-1833, vol. 20, no.12, Dec. 2010.

[89] W.-J. Tsai and H.-Y. You, "Multiple description video coding based on hierarchical B pictures using unequal redundancy," *IEEE Trans. on Circuits and Syst. for Video Technology*, pp.309-320, vol. 22, no.2., Feb. 2012

[90] C.-P. Ho, Y.-C. Sun, J.-Y Yu, and S.-Y. Lee, "Resource-aware Replication Strategies in P2P VoD Services with Scalable Video Streams," *Proc. of International Conference on Electronic & Mechanical Engineering and Information Technology*, 2012

[91] D. Comas, R. Singh, and A. Ortega, "Rate-distortion optimization in a robust video transmission based on unbalanced multiple description coding," *Proc. of IEEE Int. Workshop Multimedia Signal Processing*, 2001.

[92] T. Tillo, M. Grangetto, and M. Olmo, "Redundant Slice Optimal Allocation for H.264 Multiple Description Coding," *IEEE Trans. on Circuits and Syst. for Video Technol.*, vol. 18, no.1, pp. 59-70, Jan. 2008

[93] T. Wiegand, G.J. Sullivan, G. Bjntegaard, and A. Luthra "Overview of the H.264/AVC video coding standard," *IEEE Trans. on Circuits and Syst. for Video Technology*, pp.560-576, vol. 13, no.7, pp.560-576, July 2003.

[94] C. Lin, T. Tillo, Y. Zhao, and B. Jeon, "Multiple Description Coding for H.264/AVC with Redundancy Allocation at Macro Block Level," *IEEE Trans. on Circuits and System for Video Technology*, vol. 21, no.5, pp. 559-600, May 2011.

[95] G.J. Sullivan and T. Wiegand, "Rate-Distortion Optimization for Video Compression," *IEEE Signal Processing Magazine*, vol.15, no.6, pp. 76-90, June 1998.

[96] S. Ma, W. Gao, and Y. Lu, "Rate-distortion analysis for H.264/AVC video coding and its application to rate control," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 15, no. 12, pp. 1533–1544, Dec. 2005

[97] N. Farber, K. Stuhlmuller, and B. Girod "Analysis of error propagation in hybrid video coding with application to error resilience," *Proc. of IEEE Intel. Conf. on Image Processing*, 1999.

[98] Y. Wang, Z. Wu., and J. M. Boyce, "Modeling of Transmission-Loss-Induced Distortion in Decoded Video," *IEEE Trans. on Circuits and System for Video*

Technology, vol. 16, no.6, pp. 716-732, June 2006.

[99] P. Correia, P. Assuncao, and V. Silva, "Multiple Description of Coded Video for Path Diversity Streaming Adaptation," *IEEE Trans. on Multimedia*, vol. 14, no.3, pp. 923-935, June 2012..

[100] J. Reichel, H. Schwarz, and M. Wien, "Joint Scalable Video Model 11 (JSVM 11)," Joint Video Team, Doc. JVT-X202, 2007.

[101] G. Bjontegaard, "Improvement of the BD-PSNR model," VCEG document VCEG-AI11, ITU-T SG16/Q6, 35th VCEG Meeting, 2008.

Publication List

■ Journal papers

[J1] Wen-Jiin Tsai, **Yu-Chen Sun**, and Po-Jui Chiu, "A Robust Video Coding Based on Hybrid Hierarchical B Pictures," *IEEE Trans. Circuits Systems for Video Technology* (under revision)

[J2] **Yu-Chen Sun**, and Wen-Jiin Tsai, "Rate-Distortion Optimized Mode Selection Method for Multiple Description Video Coding," *Multimedia Tools and Applications*, (accepted)

[J3] **Yu-Chen Sun**, and Wen-Jiin Tsai, "Analysis of Unequal Error Protection for LDPCA codes," *Electronics Letters*, Vol.49, Issue.2, 2013, pp.102-104, 2013 (SCI)

[J4] **Yu-Chen Sun**, and Chun-Jen Tsai, "Perceptual-based Distributed Video Coding," *Journal of Visual Communication and Image Representation*, Vol.23, Issue 3, p.535-548, 2012. (SCI)

■ Conference papers

[C1] Wen-Jiin Tsai and **Yu-Chen Sun**, "Error-Resilient Video Coding Using Multiple Reference Frames," *Proc. of International Conference on Image Processing (ICIP) 2013*, (accepted)

[C2] Chien-Peng Ho, **Yu-Chen Sun**, Jen-Yu Yu, and Suh-Yin Lee, "Resource-aware Replication Strategies in P2P VoD Services with Scalable Video Streams," *Proc. of International Conference on Electronic & Mechanical Engineering and Information Technology*, 2012

[C3] **Yu-Chen Sun**, Shiau-Yu Lian, and Chun-Jen Tsai, "Prioritized Side Information Correction for Distributed Video Coding," *Proc. of Picture Coding Symposium 2009*

[C4] **Yu-Chen Sun**, and Chun-Jen Tsai, "Low Complexity Motion Model Analysis for Distributed Video Coding," *Proc. of International Symposium on Multimedia over Wireless 2008*