VIDEO DISTORTION ANALYSIS AND SYSTEM DESIGN FOR WIRELESS VIDEO COMMUNICATION

By

ZHIFENG CHEN

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2010

I dedicate this dissertation to my father.

ACKNOWLEDGMENTS

First and the foremost, I would like to express my deepest gratitude to my advisor Prof. Dapeng Wu for his guidance and help in the development of my research. This work would not have been possible without his enlightening instruction, constructive advice, and willingness to provide funding. His extensive knowledge, strong analytical skills, and commitment to the excellence of research are truly treasures to his students.

I would also like to thank Prof. John Harris, Prof. Tao Li, and Prof. Shigang Chen for serving on my dissertation committee and providing valuable suggestions on this dissertation. I have been fortunate to be a student of Prof. John M. Shea, who is one of the best teachers that I have had in my life. His deep knowledge, responsible attitude and impressive kindness have helped me to develop the fundamental and essential academic competence.

I am indebted to Taoran Lu for her explanation of my questions when I first encountered challenges in studying signal processing. I gratefully acknowledge the help of Xiaochen Li for my understanding in communication theory. I especially thank Jun Xu for his valuable discussions when I began my research on video coding. My work also owes much to Qian Chen for her help in the use of correct grammar, which improves the presentation of this dissertation. I would like to take this opportunity to thank Xihua Dong, Qin Chen, Lei Yang, Bing Han, Wenxing Ye, Zongrui Ding, Yakun Hu, and Jiangping Wang for many fruitful discussions related to this work.

I wish to express my special appreciation to Peshala Pahalawatta and Alexis Michael Tourapis for their help in solving my questions about the H.264/AVC JM reference software and assisting me with more rigorous expression of many ideas in this work.

Last but not least, I need to express my warmest thanks to my parents and my wife for their continued encouragement and support.

TABLE OF CONTENTS

page

ACKNOWLEDGMENTS 4

LIST OF TABLES 10

LIST OF FIGURES 11

ABSTRACT 13

CHAPTER

1 INTRODUCTION 14

1.1 Problem Statement 14
1.1.1 Theoretical Background 14
1.1.2 Challenges in the Practical System 16
1.2 Contributions of This Dissertation 19
1.3 Structure of the Dissertation 21

2 PREDICTION OF TRANSMISSION DISTORTION FOR WIRELESS VIDEO COMMUNICATION: ANALYSIS 23

2.1 Background on Transmission Distortion Prediction 23
2.2 System Description 28
2.2.1 Structure of a Wireless Video Communication System 28
2.2.2 Clipping Noise 29
2.2.3 Definition of Transmission Distortion 32
2.2.4 Limitations of the Existing Transmission Distortion Models 34
2.3 Transmission Distortion Formulae 36
2.3.1 Overview of the Approach to Analyzing PTD and FTD 37
2.3.2 Analysis of Distortion Caused by RCE 39
2.3.2.1 Pixel-level distortion caused by RCE 39
2.3.2.2 Frame-level distortion caused by RCE 40
2.3.3 Analysis of Distortion Caused by MVCE 42
2.3.3.1 Pixel-level distortion caused by MVCE 42
2.3.3.2 Frame-level distortion caused by MVCE 43
2.3.4 Analysis of Distortion Caused by Propagated Error Plus Clipping Noise 44
2.3.4.1 Pixel-level distortion caused by propagated error plus clipping noise 44
2.3.4.2 Frame-level distortion caused by propagated error plus clipping noise 48
2.3.5 Analysis of Correlation Caused Distortion 50
2.3.5.1 Pixel-level correlation caused distortion 50
2.3.5.2 Frame-Level correlation caused distortion 55

2.3.6 Summary . . . 55
2.3.6.1 Pixel-Level transmission distortion . . . 55
2.3.6.2 Frame-Level transmission distortion . . . 56
2.4 Relationship between Theorem 2.2 and Existing Transmission Distortion Models . . . 56
2.4.1 Case 1: Only the $(k-1)$-th Frame Has Error, and the Subsequent Frames are All Correctly Received . . . 57
2.4.2 Case 2: Burst Errors in Consecutive Frames . . . 57
2.4.3 Case 3: Modeling Transmission Distortion as an Output of an LTI System with PEP as input . . . 58
2.5 PTD and FTD under Multi-Reference Prediction . . . 60
2.5.1 Pixel-level Distortion under Multi-Reference Prediction . . . 60
2.5.2 Frame-level Distortion under Multi-Reference Prediction . . . 61

3 PREDICTION OF TRANSMISSION DISTORTION FOR WIRELESS VIDEO COMMUNICATION: ALGORITHM AND APPLICATION . . . 63

3.1 A Literature Review on Estimation Algorithms of Transmission Distortion . 63
3.2 Algorithms for Estimating FTD . . . 67
3.2.1 FTD Estimation without Feedback Acknowledgement . . . 67
3.2.1.1 Estimation of residual caused distortion . . . 67
3.2.1.2 Estimation of MV caused distortion . . . 70
3.2.1.3 Estimation of propagation and clipping caused distortion 72
3.2.1.4 Estimation of correlation-caused distortion . . . 74
3.2.1.5 Summary . . . 75
3.2.2 FTD Estimation with Feedback Acknowledgement . . . 75
3.3 Pixel-level Transmission Distortion Estimation Algorithm . . . 76
3.3.1 Estimation of PTD . . . 77
3.3.2 Calculation of $\hat{E}[\widetilde{\zeta}_{\mathrm{u}}^{k}]$ . . . 78
3.3.3 Calculation of $\hat{E}[\widetilde{\zeta}_{\mathrm{u+mv}_{\mathrm{u}}^{k}}^{k-j} + \widetilde{\Delta}_{\mathrm{u}}^{k}\{\bar{r}, \bar{m}\}]$ and $\hat{D}_{\mathrm{u}}^{k}(p)$ . . . 78
3.3.4 Summary . . . 79
3.4 Pixel-level End-to-end Distortion Estimation Algorithm . . . 79
3.5 Applying RMPC-PEED Algorithm to H.264 Prediction Mode Decision . . . 81
3.5.1 Rate-distortion Optimized Prediction Mode Decision . . . 81
3.5.2 Complexity of RMPC-MS, ROPE, and LLN Algorithm . . . 83
3.5.2.1 RMPC-MS algorithm . . . 84
3.5.2.2 ROPE algorithm . . . 85
3.5.2.3 LLN algorithm . . . 86
3.6 Experimental Results . . . 87
3.6.1 Estimation Accuracy and Robustness . . . 88
3.6.1.1 Experiment setup . . . 88
3.6.1.2 Estimation accuracy of different estimation algorithms . . 89
3.6.1.3 Robustness of different estimation algorithms . . . 92
3.6.2 R-D Performance of Mode Decision Algorithms . . . 93
3.6.2.1 Experiment setup . . . 94

3.6.2.2 R-D performance under no interpolation filter and no deblocking filter . . . 94
3.6.2.3 R-D performance with interpolation filter and deblocking filter . . . 96

4 THE EXTENDED RMPC ALGORITHM FOR ERROR RESILIENT RATE DISTORTION OPTIMIZED MODE DECISION . . . 101

4.1 An Overview on Subpixel-level End-to-end Distortion Estimation for a Practical Video Codec . . . 101
4.2 The Extended RMPC Algorithm for Mode Decision . . . 103
4.2.1 Subpixel-level Distortion Estimation . . . 104
4.2.2 A New Theorem for Calculating the Second Moment of a Weighted Sum of Correlated Random Variables . . . 106
4.2.3 The Extended RMPC Algorithm for Mode Decision . . . 107
4.2.4 Merits and Limitations of ERMPC Algorithm . . . 109
4.2.4.1 Merits . . . 109
4.2.4.2 Limitations . . . 110
4.3 Experimental Results . . . 110
4.3.1 Experiment Setup . . . 110
4.3.2 R-D Performance . . . 111
4.3.3 subjective Performance . . . 113
4.3.4 Discussion . . . 114
4.3.4.1 Effect of clipping noise on the mode decision . . . 114
4.3.4.2 Effect of transmission errors on mode decision . . . 115

5 RATE-DISTORTION OPTIMIZED CROSS-LAYER RATE CONTROL IN WIRELESS VIDEO COMMUNICATION . . . 117

5.1 An Literature Review on Rate Distortion Models in Wireless Video Communication Systems . . . 117
5.2 Problem Formulation . . . 122
5.3 Derivation of Bit Rate Function, Quantization Distortion Function and Transmission Distortion Function . . . 125
5.3.1 Derivation of Source Coding Bit Rate Function . . . 125
5.3.1.1 The entropy of quantized transform coefficients for I.I.D. zero-mean Laplacian source under uniform quantizer . . . 125
5.3.1.2 Improve with run length model . . . 126
5.3.1.3 Practical consideration of Laplacian assumption . . . 128
5.3.1.4 Improvement by considering the model inaccuracy . . . 128
5.3.1.5 Source coding bit rate estimation for the H.264 encoder . . . 129
5.3.2 Derivation of Quantization Distortion Function . . . 129
5.3.3 Derivation of Transmission Distortion Function . . . 130
5.3.3.1 Transmission distortion as a function of PEP . . . 130
5.3.3.2 PEP as a function of SNR, transmission rate, and channel coding rate in a fading channel . . . 131

5.3.3.3 Transmission distortion as a function of SNR, transmission rate, and channel coding rate in a fading channel 134
5.4 Rate-Distortion Optimized Cross-layer Rate Control and Algorithm Design 134
5.4.1 Optimization of Cross-layer Rate Control Problem 134
5.4.2 Algorithm Design 135
5.5 Experimental Results 137
5.5.1 Model Accuracy 137
5.5.1.1 Bit rate model 137
5.5.1.2 Quantization distortion model 138
5.5.1.3 PEP model 139
5.5.2 Performance Comparison 140
5.5.2.1 Experiment setup 141
5.5.2.2 PSNR performance 142
5.5.2.3 Subjective performance 144

6 CONCLUSION 147

6.1 Summary of the Dissertation 147
6.2 Future Work 148

APPENDIX

A PROOFS IN CHAPTER 2 150

A.1 Proof of Lemma 1 150
A.2 Proof of Proposition 1 150
A.3 Proof of Lemma 2 151
A.4 Proof of Lemma 3 152
A.5 Proof of Lemma 4 152
A.6 Lemma 5 and Its Proof 154
A.7 Lemma 6 and Its Proof 154
A.8 Proof of Corollary 1 156

B PROOFS IN CHAPTER 3 157

B.1 Proof of Proposition 3 157
B.2 Proof of Theorem 3.1 157
B.3 Proof of Proposition 4 159

C PROOFS IN CHAPTER 4 160

D PROOFS IN CHAPTER 5 162

D.1 Proof of Equation (5–5) 162
D.2 Calculation of Entropy for Different Quantized Transform Coefficients . . . 162
D.3 Proof of Proposition 5 163

REFERENCES 165

BIOGRAPHICAL SKETCH 170

LIST OF TABLES

Table page

2-1 Notations 32

2-2 An example that shows the effect of clipping noise on transmission distortion. . 36

3-1 Complexity Comparison 87

3-2 Average PSNR gain (in dB) of RMPC-MS over ROPE and LLN 96

3-3 Average PSNR gain (in dB) of RMPC-MS over ROPE and LLN under interpolation filtering 99

4-1 Average PSNR gain (in dB) of ERMPC over RMPC, LLN and ROPE 114

5-1 RCPC encoder parameters 140

LIST OF FIGURES

Figure page

1-1 Theoretical system model. 15

1-2 Theoretical system model with separate source coding and channel coding. . . 16

1-3 Practical system model of a wireless video communication system. 18

2-1 System structure, where T, Q, Q^{-1}, and T^{-1} denote transform, quantization, inverse quantization, and inverse transform, respectively. 30

2-2 The effect of clipping noise on distortion propagation. 48

2-3 Temporal correlation between the residuals in one trajectory. 51

2-4 Temporal correlation matrix between residual and MVCE in one trajectory. . . . 51

2-5 Temporal correlation matrix between MVCEs in one trajectory. 52

2-6 Comparison between measured and estimated correlation coefficients. 53

3-1 Transmission distortion D^k vs. frame index k for 'foreman': (a) good channel, (b) poor channel. 90

3-2 Transmission distortion D^k vs. frame index k for 'stefan': (a) good channel, (b) poor channel. 91

3-3 Transmission distortion D^k vs. PEP for 'foreman'. 91

3-4 Transmission distortion D^k vs. PEP for 'stefan'. 92

3-5 Transmission distortion D^k vs. frame index k for 'foreman' under imperfect knowledge of PEP: (a) good channel, (b) poor channel. 93

3-6 Transmission distortion D^k vs. frame index k for 'stefan' under imperfect knowledge of PEP: (a) good channel, (b) poor channel. 93

3-7 PSNR vs. bit rate for 'foreman', with no interpolation filter and no deblocking filter: (a) PEP=2%, (b) PEP=5%. 95

3-8 PSNR vs. bit rate for 'football', with no interpolation filter and no deblocking filter: (a) PEP=2%, (b) PEP=5%. 95

3-9 PSNR vs. bit rate for 'foreman', with interpolation and no deblocking: (a) PEP=2%, (b) PEP=5%. 98

3-10 PSNR vs. bit rate for 'football', with interpolation and no deblocking: (a) PEP=2%, (b) PEP=5%. 98

3-11 PSNR vs. bit rate for 'foreman', with interpolation and deblocking: (a) PEP=2%, (b) PEP=5%. 100

3-12 PSNR vs. bit rate for 'football', with interpolation and deblocking: (a) PEP=2%, (b) PEP=5%. 100

4-1 PSNR vs. bit rate for 'foreman': (a) PEP=0.5%, (b) PEP=2%. 111

4-2 PSNR vs. bit rate for 'mobile': (a) PEP=0.5%, (b) PEP=2%. 112

4-3 (a) ERMPC at the 84-th frame, (b) RMPC at the 84-th frame, (c) LLN at the 84-th frame, (d) ROPE at the 84-th frame, (e) ERMPC at the 99-th frame, (f) RMPC at the 99-th, (g) LLN at the 99-th frame, (h) ROPE at the 99-th frame. . . 115

4-4 PSNR vs. bit rate for 'foreman': (a) PEP=0.5%, (b) PEP=2%. 115

4-5 PSNR vs. bit rate for 'mobile': (a) PEP=0.5%, (b) PEP=2%. 116

5-1 Channel model. 123

5-2 Variance model. 127

5-3 bpp vs. Frame index: (a) foreman, (b) mobile. 138

5-4 Quantization vs. Frame index: (a) foreman, (b) mobile. 139

5-5 PEP under different RCPC coding rates. 141

5-6 PSNR vs. average SNR: (a) foreman, (b) mobile. 142

5-7 PSNR vs. bandwidth: (a) foreman, (b) mobile. 144

5-8 A random channel sample under average SNR=10dB and bit rate=1000kbps: (a) A random SNR sample, (b) Distortion vs. Frame index for foreman_cif under this channel. 145

5-9 For the 10-th frame: (a) original, (b) CLRC, (c) proposed-constant-PEP, (d) constant-PEP-QP-limit; for the 11-th frame: (e) original, (f) CLRC, (g) proposed-constant-PEP, (h) constant-PEP-QP-limit. 146

A-1 Comparison of $\Phi^2(x, y)$ and x^2. 154

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

VIDEO DISTORTION ANALYSIS AND SYSTEM DESIGN FOR WIRELESS VIDEO COMMUNICATION

By

Zhifeng Chen

December 2010

Chair: Dapeng Wu
Major: Electrical and Computer Engineering

In this dissertation, we address the problem of minimizing the end-to-end distortion in wireless video communication. We first analytically derive transmission distortion as a function of video statistics, channel conditions and system parameters for wireless video communication systems. Then we design practical algorithms to estimate the system parameters and video statistics. Given the channel condition, we may accurately predict the instantaneous transmission distortion by our formulae and estimation algorithms. We also prove a new theorem to extend our algorithms to support rate-distortion optimized mode decision in practical video codecs. Finally, we derive a more accurate source bit rate model and quantization distortion model than existing parametric models. Our models help us to design a rate-distortion optimized cross-layer rate control algorithm for minimizing the end-to-end distortion under resource constraints in wireless video communication systems. Our results achieve remarkable performance gains over existing solutions.

CHAPTER 1
INTRODUCTION

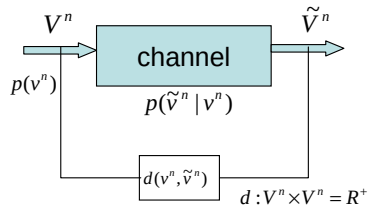
1.1 Problem Statement

Both multimedia technology and mobile communications have experienced massive growth and commercial success in recent years. As these two technologies converge, wireless video, such as videophone calls and mobile TV in 3G/4G systems, is expected to achieve unprecedented growth and worldwide success. Therefore, how to improve the video quality reproduced at the video decoder in a wireless video communication system becomes more compelling.

1.1.1 Theoretical Background

A theoretical system model for video transmission over a wireless channel is shown in Fig. 1-1, where V^n is the input video sequence and $\tilde{V}^n$ is the output video sequence after V^n passing through the wireless channel. The target of the transmission is to convey the video information at the input side to the output side as much as possible. However, 1) the video sequence is usually highly redundant, which causes the waste of resources if it is transmitted without removing any redundancy; and 2) the source bit stream is usually not well distinguishable at the output side after passing through the channel, which causes serious distortion. Therefore, to convey maximum distinguishable video information while consuming minimum resources for the information to be transmitted from the transmitter to the receiver, we need 1) compressing the input using as few bits as possible, that is, source coding; and 2) mapping the source bit stream into a better, in the bit error sense, bit stream for transmission, that is, channel coding.

Now the problems are 1) what is the minimum requirement of resources for reliably transmitting the given source? 2) for the given channel, how much information at most can be reliably transmitted? and 3) what is the minimum distortion which may happen if the information contained in the given source is more than the information the channel



$$D = E[d(V^n, \tilde{V}^n)]$$

Figure 1-1. Theoretical system model.

may convey? In 1948, Shannon published his seminal work “A Mathematical Theory of Communication” in **Bell Systems Technical Journal** [1]. In this paper, Shannon mathematically defines the measure of information by entropy, which is expressed by the average number of bits needed for storage or communication. In this seminal work the answers are given, for the first time, to the first two aforementioned questions, that is, 1) the minimum number of bits required for reliably transmitting the given source is its entropy; 2) the maximum number of bits can be reliably transmitted for the given channel is the channel capacity.Although not rigorously proved, the answer to the third question is also presented in Ref. [1] (his Theorem 21). That is 3) the minimum distortion for the given source and channel is the minimum distortion achieved by lossy source coding under the condition that the encoded source rate is less than the channel capacity.

In 1959, Shannon published another famous work “Coding Theorems for a Discrete Source With a Fidelity Criterion” [2], where the rate-distortion function is first coined and the greatest lower bound of rate for a given distortion is proved. The joint source channel coding theorem proves that the optimal performance can be achieved by the source channel separation theorem as stated in Ref. [3] “The source channel separation theorem shows that we can design the source code and the channel code separately and combine the results to achieve optimal performance.” Based on the source channel

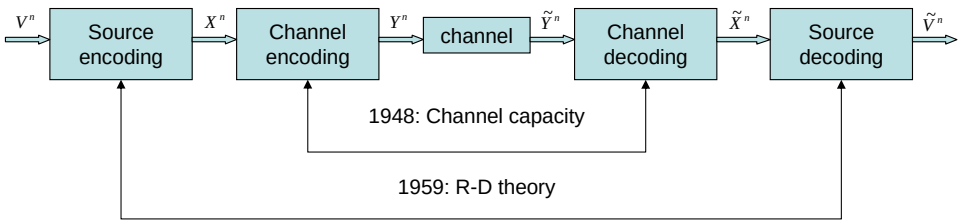


Figure 1-2. Theoretical system model with separate source coding and channel coding.

separation theorem, the theoretical system model is to separate source coding and channel coding separately and sequentially as in Fig. 1-2

However, this theorem is derived under the condition that all transmission errors can be corrected by the channel coding to an arbitrarily low probability. That is, it implicitly assumes that there is no distortion caused by the transmission error in the system model. Although decreasing the channel protection, i.e., redundant bits will increase the transmission error, it also reduces the distortion caused by lossy source coding given the same channel capacity. Therefore, it is still not clear that what is the minimum distortion for the given source and channel if the restriction of arbitrarily low probability of transmission error is lifted. In addition, the channel capacity is derived based on the assumptions of infinite block length, random coding and stationary channel. On the other hand, the rate-distortion (R-D) bound is derived based on the assumptions of infinite block length, random coding, and stationary sources. These assumptions in both channel capacity and R-D bound incur infinite delay, infinitely high complexity, and mismatch between theoretical and practical source and channel models.

1.1.2 Challenges in the Practical System

In a practical wireless video communication system, the resources are very limited. There are usually four kinds of resources, that is, time, bandwidth, power and space, which can be utilized to improve wireless video performance. However, all of these

four resources are usually limited. Specifically, 1) the end-to-end delay, sum of source coding delay and transmission delay, for the video signal to be reproduced by video decoder is under certain delay bound; 2) the achievable data rate, sum of information rate and redundant rate, is under certain bandwidth limit; 3) the total power consumed by video encoding and by transmission are under certain constraint; 4) the channel gain in a wireless fading channel statistically depends on the geographical position and environment. Therefore, due to the limited resources, the probability of transmission error cannot be arbitrarily low in the practical system. Instead, a more desirable system design is to minimize the end-to-end distortion under the resource constraints by allowing the transmission error at a certain level .

In a practical wireless communication system, modulation and error control coding are designed to mitigate the bit error during transmission through an error-prone channel. In application layer, error-resilient coding at the encoder and error concealment at the decoder are designed to reduce the distortion caused by such transmission error. We call the distortion caused by the transmission error as **transmission distortion**, denoted by D_t. In a practical video coding system, the predictive coding, quantization, transform, and entropy coding are adopted together to compress the bits. Such a source coding scheme produces error during quantization[1] . We call the distortion caused by quantization error as **quantization distortion**, denoted by D_q. As a result, the distortion between the original video and the reconstructed video at the video decoder is caused by both the quantization error and transmission error. We call them together as **end-to-end distortion**, denoted by D_{ete}. The practical system model is shown in Fig. 1-3

On the one hand, the transmission distortion is a function of the transmission error, which is again a function of signal-to-noise ratio (SNR), bandwidth, delay requirement

[1] In modern video codec, e.g. H.264 codec, the transform is so designed that it is reversible.

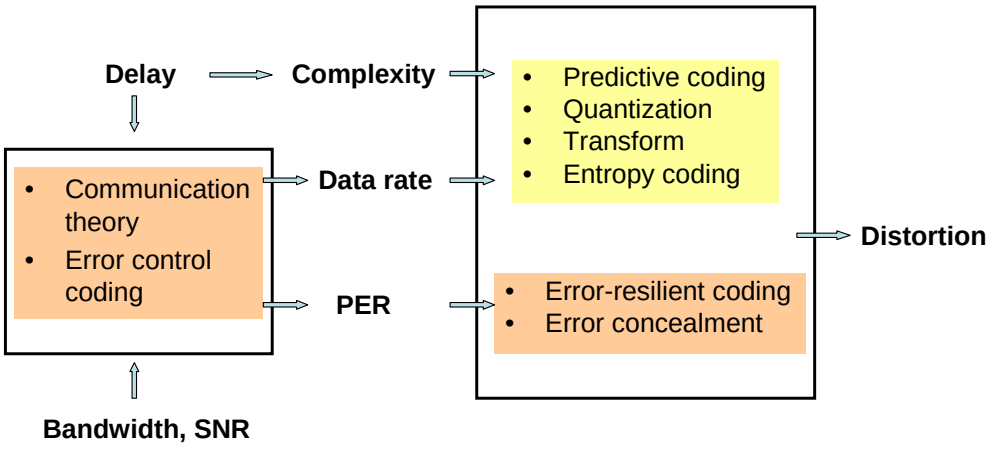


Figure 1-3. Practical system model of a wireless video communication system.

and channel protection parameters, e.g., modulation order and channel coding rate. On the other hand, the quantization distortion is a function of available source data rate, complexity requirement, source encoder structure and source coding parameters, e.g., the allowable finite set for quantization. Now the problem in a practical system can be formulated by "Given the source, channel, resources and system structure, how to tune the system parameters to minimize the end-to-end distortion."

This problem is very challenging since 1) the statistical properties of the video source is unknown and the source is usually not stationary; 2) the wireless channel is time varying; 3) all resources are limited; 4) the system is a complex system, e.g., non-linear; 5) the system parameters in different layers are usually coupled; for example, increasing the the channel coding rate in the transmitter will decrease the source data rate for compression.

To tackle this complex problem, we need to follow the following steps: 1) Finding stable video statistics for quantization distortion and deriving the quantization distortion as a function of source rate constraint (R_s), complexity constraint (C_s), video codec structure and those stable video statistics ($\vec{\theta}$); 2) Finding stable video statistics for transmission distortion and deriving the transmission distortion as a function of packet error probability (PEP), codec structure and those stable video statistics ($\vec{\theta}$); 3) Deriving

the PEP as a function of SNR, channel coding rate (R_c), bandwidth, and transmission delay (d_t); 4) Minimizing the end-to-end distortion under the resource constraints. Thanks to the source channel separation theorem, the source coding and channel coding has been extensively studied separately. In other words, the first step has been extensively studied by the source coding society and the third step has been extensively studied by the communication society separately. In the Open System Interconnection Reference Model (OSI Reference Model or OSI Model), the first step belongs to the application layer problem, and the third step belongs to the lower layers. Although they are relatively extensively researched, they still need to be further investigated in order to design a practical system with the minimum end-to-end distortion. On the other hand, the second step, which in fact is a cross-layer problem, has long be omitted by both societies. Untill now, there is still no well accepted theoretical analysis for this cross-layer problem. If we can find transmission distortion as a closed-form function of PEP, we may be able to analytically derive the minimum end-to-end distortion for most existing wireless video communication systems, which are designed based on the source channel separation theorem.

1.2 Contributions of This Dissertation

The major contributions of our work are summarized as follows:

1. We analytically derive the transmission distortion formulae as a function of PEP and video statistics for wireless video communication systems.
2. With consideration of spatio-temporal correlation, nonlinear codec and time-varying channel, our formulae provide, for the first time, the following capabilities:
    - support of distortion prediction at different levels (e.g., pixel/frame/GOP level).
    - support of multi-reference picture motion compensated prediction.
    - support of slice data partitioning.
    - support of arbitrary slice-level packetization with FMO mechanism.
    - being applicable to time-varying channels.
    - one unified formula for both I-MB and P-MB.

- support of both low motion and high motion video sequences.

3. Besides deriving the transmission distortion formulae, we also identified two important properties of transmission distortion for the first time:

   - clipping noise, produced by non-linear clipping, causes decay of propagated error.
   - the correlation between motion vector concealment error and propagated error is negative, and has dominant impact on transmission distortion, among all the correlations between any two of the four components in transmission error.

4. We also discussed the relationship between our formula and existing models, and specify the conditions, under which those existing models are accurate.

5. We design algorithms to estimate correlation ratio and propagation factor, which facilitates the design of low complexity algorithm for estimating the frame-level transmission distortion (FTD).

6. By using the formulae analytically derived and the parameter estimated by statistics, our FTD estimation algorithm, called **RMPC-FTD**, is more accurate and more robust than existing FTD algorithms.

7. Another advantage of our RMPC-FTD algorithm is that all parameters in the formulae can be estimated by using the instantaneous video frame statistics and channel conditions, which allows the video frame statistics to be time-varying and the transmission error processes to be non-stationary. As a result, our RMPC-FTD algorithm is more suitable for real-time video communication.

8. We also design the estimation algorithm, called **RMPC-PTD**, for pixel-level transmission distortion (PTD) by utilizing the known values of the MV and corresponding residual to further improve the estimation accuracy and decrease the estimation complexity.

9. We also extend RMPC-PTD to estimate pixel-level end-to-end distortion (PEED) by the algorithm called **RMPC-PEED**. Our RMPC-PEED algorithm provides not only more accurate estimation but also lower complexity and higher degree of extensibility than the existing methods.

10. We apply our RMPC-PEED algorithm to prediction mode decision in H.264; the resulting algorithm is called **RMPC-MS**. Experimental results show that our RMPC-MS algorithm achieves more than 1dB gain than existing algorithms.

11. To facilitate the design of subpixel-level Mean Square Error (MSE) distortion estimation for mode decision in H.264 video encoders, we prove a general theorem

for calculating the second moment of a weighted sum of correlated random variables without the requirement of their probability distribution.

12. We apply our theorem to the design of a very low-complexity algorithm, which we call ERMPC algorithm, for mode decision in H.264. Experimental results show that, ERMPC further achieves 0.25dB PSNR gain over the RMPC-MS algorithm.

13. We derive more accurate source bit rate model and quantization distortion model than existing parametric models.

14. We improve the performance bound for channel coding with convolutional codes and a Viterbi decoder, and derive its performance under Rayleigh block fading channel.

15. We design a R-D optimized cross-layer rate control (CLRC) algorithm by jointly choosing quantization step size and channel coding rate based on the given instantaneous channel condition, e.g., SNR and channel bandwidth.

1.3 Structure of the Dissertation

In Chapter 2, we analytically derive the transmission distortion formulae as a function of PEP and video statistics for wireless video communication systems. We explain the limitations in existing transmission distortion models, where the significant effect of clipping noise on the transmission distortion has long been omitted. We then derive both the PTD and FTD with considering the clipping noise in the system. We also discussed the relationship between our formula and existing models; we specify the conditions, under which those existing models are accurate.

In Chapter 3, we design practical algorithms to estimate the system parameters, and from the estimated parameters, we may calculate the FTD by using the formulae derived in Chapter 2. For PTD, we utilize the known values, e.g. residual, in some video codec replacing the statistics of the corresponding random variables to simplify the PTD estimation and design a low-complexity and high-accuracy PTD estimation algorithm. We also extend RMPC-PTD algorithm to estimate PEED with high degree of extensibility. we then apply our RMPC-PEED algorithm for mode decision in H.264 to achieve the minimum R-D cost. The complexity and memory requirement of our RMPC-MS algorithm and existing mode selection algorithms are carefully compared in

this chapter. Experimental results are given to compare the estimation accuracy, robust, R-D performance and extensibility between our algorithms and existing algorithms.

In Chapter 4, we extend our RMPC-MS algorithm designed in Chapter 3 to support some performance-enhanced parts, e.g. interpolation filter, in H.264 codec. We first prove a new theorem for calculating the second moment of a weighted sum of correlated random variables without the requirement of their probability distribution. Then, we apply the theorem to extend the design of previous RMPC-MS algorithm to support the interpolation filtering in H.264. We call the new algorithm as ERMPC algorithm. We also discuss the merits and limitations of our ERMPC algorithm. Experimental results are given to compare the R-D performance and subjective performance between ERMPC and existing algorithms.

In Chapter 5, we aim to design a rate-distortion optimized cross-layer rate control (CLRC) algorithm for wireless video communication. To this end, we derive a more accurate source bit rate model and quantization distortion model than existing parametric models. We also improve the performance bound of channel coding with convolutional codes and a Viterbi decoder, and derive its performance under Rayleigh block fading channels. Given the instantaneous channel condition, i.e. SNR and bandwidth, we design the rate-distortion optimized CLRC algorithm by jointly choosing quantization step size and channel coding rate. Experimental results are given to compare the models accuracy between ours and existing models. We also compare the R-D performance and subjective performance between our algorithms and existing algorithms in this chapter.

Finally, Chapter 6 concludes the dissertation and provides an outlook for our future work.

CHAPTER 2
PREDICTION OF TRANSMISSION DISTORTION FOR WIRELESS VIDEO COMMUNICATION: ANALYSIS

In this chapter, we analytically derived the transmission distortion formulae for wireless video communication systems. We also discussed the relationship between our formula and existing models.

2.1 Background on Transmission Distortion Prediction

Transmission distortion is caused by packet errors during the transmission of a video sequence, and it is the major part of the end-to-end distortion in delay-sensitive wireless video communication[1] under high packet error probability (PEP), e.g., in a wireless fading channel. The capability of predicting transmission distortion at the transmitter can assist in designing video encoding and transmission schemes that achieve maximum video quality under resource constraints. Specifically, transmission distortion prediction can be used in the following three applications in video encoding and transmission: 1) mode selection, which is to find the best intra/inter-prediction mode for encoding an macroblock (MB) with the minimum rate-distortion (R-D) cost given the instantaneous PEP, 2) cross-layer rate control, which is to control the instantaneously encoded bit rate for a real-time encoder to minimize the frame-level end-to-end distortion given the instantaneous PEP, e.g., in video conferencing, 3) packet scheduling, which chooses a subset of packets of the pre-coded video to transmit and intentionally discards the remaining packets to minimize the GOP-level (Group of Picture) end-to-end distortion given the average PEP and average burst length, e.g., in streaming pre-coded video over networks. All the three applications require a formula for predicting how transmission distortion is affected by their respective control policy, in order to choose the optimal mode or encoding rate or transmission schedule.

[1] Delay-sensitive wireless video communication usually does not allow retransmission to correct packet errors since retransmission may cause long delay.

However, predicting transmission distortion poses a great challenge due to the spatio-temporal correlation inside the input video sequence, the nonlinearity of both the encoder and the decoder, and varying PEP in time-varying channels. In a typical video codec, the temporal correlation among consecutive frames and the spatial correlation among the adjacent pixels of one frame are exploited to improve the coding efficiency. Nevertheless, such a coding scheme brings much difficulty in predicting transmission distortion because a packet error will degrade not only the video quality of the current frame but also the following frames due to error propagation. In addition, as we will see in Section 2.3, the nonlinearity of both the encoder and the decoder makes the instantaneous transmission distortion not equal to the sum of distortions caused by individual error events. Furthermore, in a wireless fading channel, the PEP is time-varying, which makes the error process a non-stationary random process and hence, as a function of the error process, the distortion process is also a non-stationary random process.

According to the aforementioned three applications, the existing algorithms for estimating transmission distortion can be categorized into the following three classes: 1) pixel-level or block-level algorithms (applied to mode selection), e.g., Recursive Optimal Per-pixel Estimate (ROPE) algorithm [4] and Law of Large Number (LLN) algorithm [5, 6]; 2) frame-level or packet-level or slice-level algorithms (applied to cross-layer rate control) [7–11]; 3) GOP-level or sequence-level algorithms (applied to packet scheduling) [12–16]. Although the existing distortion estimation algorithms work at different levels, they share some common properties, which come from the inherent characteristics of wireless video communication system, that is, spatio-temporal correlation, nonlinear codec and time-varying channel. In this chapter, we use the divide-and-conquer approach to decompose complicated transmission distortion into four components, and analyze their effects on transmission distortion individually. This

divide-and-conquer approach enables us to identify the governing law that describes how the transmission distortion process evolves over time.

Stuhlmuller et al. [8] observed that the distortion caused by the propagated error decays over time due to spatial filtering and intra coding of MBs, and analytically derived a formula for estimating transmission distortion under spatial filtering and intra coding. The effect of spatial filtering is analyzed under the implicit assumption that MVs are always correctly received at the receiver, while the effect of intra coding is modeled as a linear decay under another implicit assumption that the I-MBs are also always correctly received at the receiver. However, these two assumptions are usually not valid in realistic delay-sensitive wireless video communication. To address this, this chapter derives the transmission distortion formula under the condition that both I-MBs and MVs may be erroneous at the receiver. In addition, we observe an interesting phenomenon that even without using the spatial filtering and intra coding, the distortion caused by the propagated error still decays! We identify, for the first time, that this decay is caused by non-linear clipping, which is used to clip those out-of-range[2] reconstructed pixel after motion compensation; this is the first of the two properties identified in this chapter. While such out-of-range values produced by the inverse transform of quantized transform coefficients is negligible at the encoder, its counterpart produced by transmission error at the decoder has significant impact on transmission distortion.

Some existing works [8, 9] estimate transmission distortion based on a linear time-invariant (LTI) system model, which regards packet error as input and transmission distortion as output. The LTI model simplifies the analysis of transmission distortion. However, it sacrifices accuracy in distortion estimation since it neglects the effect of correlation between newly induced error and propagated error. Liang et al. [16]

[2] A reconstructed pixel value may be out of the range of the original pixel value, e.g., [0, 255].

studied the effect of correlation and observed that the LTI models [8, 9] underestimate transmission distortion due to the positive correlation between two adjacent erroneous frames; however, they did not consider the effect of motion vector (MV) error on transmission distortion and their algorithm was not tested with high motion videos. To address these issues and find the root cause of that underestimation, this chapter classifies the transmission reconstructed error into three independent random errors, namely, Residual Concealment Error (RCE), MV Concealment Error (MVCE), and propagated error; the first two types of error are called *newly induced error*. We identify, for the first time, that MVCE is negatively correlated with propagated error and this correlation has dominant impact on transmission distortion, among all the correlations between any two of the three error types, for high motion videos; this is the second of the two properties identified in this chapter. For this reason, as long as MV transmission errors exist in high motion videos, the LTI model over-estimates transmission distortion. We also quantifies the effect of individual error types and their correlations on transmission distortion in this chapter. Thanks to the analysis of correlation effect, our distortion formula is accurate for both low motion video and high motion video as verified by experimental results. Another merit of considering the effect of MV error on transmission distortion is the applicability of our results to video communication with slice data partitioning, where the residual and MV could be transmitted under Unequal Error Protection (UEP).

Refs. [4, 5, 10, 11] proposed some models to estimate transmission distortion under the consideration that both MV and I-MB may experience transmission errors. However, the parameters in the linear models [10, 11] can only be acquired by experimentally curve-fitting over multiple frames, which forbids the models from estimating instantaneous distortion. In addition, the linear models [10, 11] still assume there is no correlation between the newly induced error and propagated error. In Ref. [4], the ROPE algorithm considers the correlation between MV concealment error and

propagated error by recursively calculating the second moment of the reconstructed pixel value. However, ROPE neglects the non-linear clipping function and therefore over-estimates the distortion. In addition, the extension of ROPE algorithm [17] to support the averaging operations, such as interpolation and deblocking filtering in H.264, requires intensive computation of correlation coefficients due to the high correlation between reconstructed values of adjacent pixels, and thereby prohibiting it from applying to H.264. In H.264 reference code JM14.0[3] , the LLN algorithm [5] is adopted since it is capable of supporting both clipping and averaging operations. However, in order to predict transmission distortion, all possible error events for each pixel in all frames should be simulated at the encoder, which significantly increases the complexity of the encoder. Different from Ref. [4, 5], the divide-and-conquer approach in this chapter enables our formula to provide not only more accurate prediction but also lower complexity and higher degree of extensibility. The multiple reference picture motion compensated prediction extended from the single reference is analyzed in Section 2.5, and, for the first time, the effect of multiple references on transmission distortion is quantified. In addition, the transmission distortion formula derived in this chapter is unified for both I-MBs and P-MBs, in contrast to two different formulae in Refs. [4, 10, 11].

Different from wired channels, wireless channels suffer from multipath fading, which can be regarded as multiplicative random noise. Fading leads to time-varying PEP and burst errors in wireless video communication. Ref. [8] uses a two-state stationary Markov chain to model burst errors. However, even if the channel gain is stationary, packet error process is a non-stationary random process. Specifically, since PEP is a function of the channel gain [18], which is not constant in a wireless fading channel, instantaneous PEP is also not constant. This means the probability

[3] http://iphome.hhi.de/suehring/tml/download/old_jm/jm14.0.zip

distribution of packet error state is time-varying in wireless fading channels, that is, the packet error process is a non-stationary random process. Hence the Markov chain in Ref. [8] is neither stationary, nor ergodic for wireless fading channel. As a result, averaging the burst length and PEP as in Ref. [8] cannot accurately predict instantaneous distortion. To address this, this chapter derives the formula for Pixel-level Transmission Distortion (PTD) by considering non-stationarity over time. Regarding the Frame-level Transmission Distortion (FTD), since two adjacent MBs may be assigned to two different packets, under the slice-level packetization and FMO mechanism in H.264 [19, 20], their error probability could be different. However, existing frame-level distortion models [8–11] assume all pixels in the same frame experience the same channel condition. As a result, the applicable scope of those models are limited to video with small resolution. In contrast, this chapter derives the formula for FTD by considering non-stationarity over space. Due to consideration of non-stationarity over time and over space, our formula provides an accurate prediction of transmission distortion in a time-varying channel.

The rest of the chapter is organized as follows. Section 2.2 presents the preliminaries of the system under study to facilitate the derivations in the later sections, and illustrates the limitations of existing transmission distortion models. In Section 2.3, we derive the transmission distortion formula as a function of video statistics, channel condition, and codec system parameters. Section 2.4 discusses the relationship between our formula and the existing models. In Section 2.5, we extend formulae for PTD and FTD from single-reference to multi-reference.

2.2 System Description

2.2.1 Structure of a Wireless Video Communication System

Fig. 2-1 shows the structure of a typical wireless video communication system. It consists of an encoder, two channels and a decoder where residual packets and MV packets are transmitted over their respective channels. If residual packets or MV packets

are erroneous, the error concealment module will be activated. In typical video encoders such as H.263/264 and MPEG-2/4 encoders, the functional blocks can be divided into two classes: 1) basic parts, such as predictive coding, transform, quantization, entropy coding, motion compensation, and clipping; and 2) performance-enhanced parts, such as interpolation filtering, deblocking filtering, B-frame, multi-reference prediction, etc. Although the up-to-date video encoder includes more and more performance-enhanced parts, the basic parts do not change. In this chapter, we use the structure in Fig. 2-1 for transmission distortion analysis. Note that in this system, both residual channel and MV channel are application-layer channels; specifically, both channels consist of entropy coding and entropy decoding, networking layers[4] , and physical layer (including channel encoding, modulation, wireless fading channel, demodulation, channel decoding). Although the residual channel and MV channel usually share the same physical-layer channel, the two application-layer channels may have different parameter settings (e.g., different channel code-rate) for the slice data partitioning under UEP. For this reason, our formula obtained from the structure in Fig. 2-1 can be used to estimate transmission distortion for an encoder with slice data partitioning.

2.2.2 Clipping Noise

In this subsection, we examine the effect of clipping noise on the reconstruction pixel value along each pixel trajectory over time (frames). All pixel positions in a video sequence form a three-dimensional spatio-temporal domain, i.e., two dimensions in spatial domain and one dimension in temporal domain. Each pixel can be uniquely represented by $\mathbf{u}^k$ in this three-dimensional time-space, where k means the k-th frame in temporal domain and $\mathbf{u}$ is a two-dimensional vector in spatial domain. The philosophy behind inter-coding of a video sequence is to represent the video sequence by virtual motion of each pixel, i.e., each pixel recursively moves from position $\mathbf{v}^{k-1}$ to position $\mathbf{u}^k$.

[4] Here, networking layers can include any layers other than physical layer.

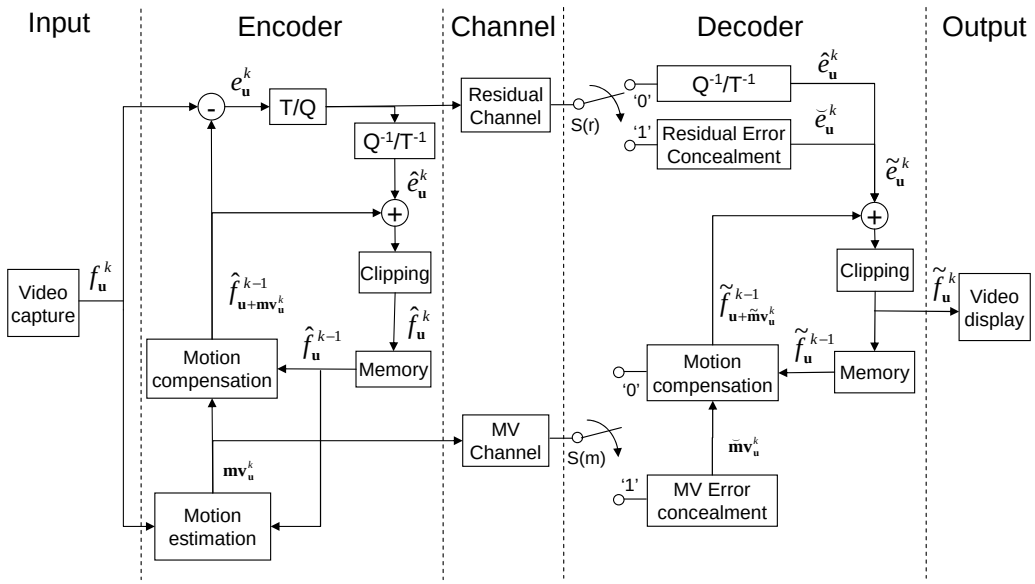


Figure 2-1. System structure, where T, Q, Q^{-1}, and T^{-1} denote transform, quantization, inverse quantization, and inverse transform, respectively.

The difference between these two positions is a two-dimensional vector called MV of pixel $\mathbf{u}^k$, i.e., $\mathbf{mv}_{\mathrm{u}}^k = \mathbf{v}^{k-1} - \mathbf{u}^k$. The difference between the pixel values of these two positions is called residual of pixel $\mathbf{u}^k$, that is, $e_{\mathrm{u}}^k = f_{\mathrm{u}}^k - \hat{f}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-1}$ [5]. Recursively, each pixel in the k-th frame has one and only one reference pixel trajectory backward towards the latest I-frame.

At the encoder, after transform, quantization, inverse quantization, and inverse transform for the residual, the reconstructed pixel value may be out-of-range and should be clipped as

$$\hat{f}_{\mathrm{u}}^k = \Gamma(\hat{f}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-1} + \hat{e}_{\mathrm{u}}^k), \tag{2–1}$$

[5] For simplicity of notation, we move the superscript k of $\mathbf{u}$ to the superscript k of f whenever $\mathbf{u}$ is the subscript of f.

where $\Gamma(\cdot)$ function is a clipping function defined by

$$\Gamma(x) = \begin{cases} \gamma_L, & x < \gamma_L \\ x, & \gamma_L \leq x \leq \gamma_H \\ \gamma_H, & x > \gamma_H, \end{cases} \tag{2–2}$$

where γ_L and γ_H are user-specified low threshold and high threshold, respectively. Usually, $\gamma_L = 0$ and $\gamma_H = 255$.

The residual and MV at the decoder may be different from their counterparts at the encoder because of channel impairments. Denote $\widetilde{\mathbf{mv}}_{\mathrm{u}}^k$ and $\widetilde{e}_{\mathrm{u}}^k$ the MV and residual at the decoder, respectively. Then, the reference pixel position for $\mathbf{u}^k$ at the decoder is $\widetilde{\mathbf{v}}^{k-1} = \mathbf{u}^k + \widetilde{\mathbf{mv}}_{\mathrm{u}}^k$, and the reconstructed pixel value for $\mathbf{u}^k$ at the decoder is

$$\widetilde{f}_{\mathrm{u}}^k = \Gamma(\widetilde{f}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1} + \widetilde{e}_{\mathrm{u}}^k). \tag{2–3}$$

In error-free channels, the reconstructed pixel value at the receiver is exactly the same as the reconstructed pixel value at the transmitter, because there is no transmission error and hence no transmission distortion. However, in error-prone channels, we know from (2–3) that $\widetilde{f}_{\mathrm{u}}^k$ is a function of three factors: the received residual $\widetilde{e}_{\mathrm{u}}^k$, the received MV $\widetilde{\mathbf{mv}}_{\mathrm{u}}^k$, and the propagated error $\widetilde{f}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1}$. The received residual $\widetilde{e}_{\mathrm{u}}^k$ depends on three factors, namely, 1) the transmitted residual $\hat{e}_{\mathrm{u}}^k$, 2) the residual packet error state, which depends on instantaneous residual channel condition, and 3) the residual error concealment algorithm if the received residual packet is erroneous. Similarly, the received MV $\widetilde{\mathbf{mv}}_{\mathrm{u}}^k$ depends on 1) the transmitted $\mathbf{mv}_{\mathrm{u}}^k$, 2) the MV packet error state, which depends on instantaneous MV channel condition, and 3) the MV error concealment algorithm if the received MV packet is erroneous. The propagated error $\widetilde{f}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1}$ includes the error propagated from the reference frames, and therefore depends on all samples in the previous frames indexed by $i < k$ and their reception error states as well as concealment algorithms.

Table 2-1. Notations

$\mathbf{u}^k$	: Three-dimensional vector that denotes a pixel position in an video sequence
f_{u}^k	: Value of the pixel $\mathbf{u}^k$
e_{u}^k	: Residual of the pixel $\mathbf{u}^k$
$\mathbf{mv}_{\mathrm{u}}^k$	: MV of the pixel $\mathbf{u}^k$
Δ_{u}^k	: Clipping noise of the pixel $\mathbf{u}^k$
$\varepsilon_{\mathrm{u}}^k$	: Residual concealment error of the pixel $\mathbf{u}^k$
ξ_{u}^k	: MV concealment error of the pixel $\mathbf{u}^k$
ζ_{u}^k	: Transmission reconstructed error of the pixel $\mathbf{u}^k$
S_{u}^k	: Error state of the pixel $\mathbf{u}^k$
P_{u}^k	: Error probability of the pixel $\mathbf{u}^k$
D_{u}^k	: Transmission distortion of the pixel $\mathbf{u}^k$
D^k	: Transmission distortion of the k-th frame
$\mathcal{V}^k$	: Set of all the pixels in the k-th frame
$\lvert\mathcal{V}\rvert$	: Number of elements in set $\mathcal{V}$ (cardinality of $\mathcal{V}$)
α^k	: Propagation factor of the k-th frame
β^k	: Percentage of I-MBs in the k-th frame
λ^k	: Correlation ratio of the k-th frame
$w^k(j)$	: pixel percentage of using frame $k-j$ as reference in the k-th frame

The non-linear clipping function within the pixel trajectory makes the distortion estimation more challenging. However, it is interesting to observe that clipping actually reduces transmission distortion. In Section 2.3, we will quantify the effect of clipping on transmission distortion.

Table 2-1 lists notations used in this chapter. All vectors are in bold font. Note that the encoder needs to reconstruct the compressed video for predictive coding; hence the encoder and the decoder have a similar structure for pixel value reconstruction. To distinguish the variables in the reconstruction module of the encoder from those in the reconstruction module of the decoder, we add $\hat{}$ onto the variables at the encoder and add $\tilde{}$ onto the variables at the decoder.

2.2.3 Definition of Transmission Distortion

In this subsection, we define PTD and FTD to be derived in Section 2.3. To calculate FTD, we need some notations from set theory. In a video sequence, all pixel positions in the k-th frame form a two-dimensional vector set $\mathcal{V}^k$, and we denote

the number of elements in set $\mathcal{V}^k$ by $|\mathcal{V}^k|$. So, for any pixel at position $\mathbf{u}$ in the k-th frame, i.e., $\mathbf{u} \in \mathcal{V}^k$, its reference pixel position is chosen from set $\mathcal{V}^{k-1}$ for single-reference. Usually, the set $\mathcal{V}^k$ in a video sequence is the same for all frame k, i.e., $\mathcal{V}^1 = \cdots = \mathcal{V}^k$ for all $k > 1$. Hence, we remove the frame index k and denote the set of pixel positions of an arbitrary frame by $\mathcal{V}$. Note that in H.264, a reference pixel may be in a position out of picture boundary; however, the set of reference pixels, which is larger than the input pixel set, is still the same for all frame k.

For a transmitter with feedback acknowledgement of whether a packet is correctly received at the receiver (called acknowledgement feedback), $\widetilde{f}_{\mathrm{u}}^k$ at the decoder side can be perfectly reconstructed by the transmitter, as long as the transmitter knows the error concealment algorithm used by the receiver. Then, the transmission distortion for the k-th frame can be calculated by mean squared error (MSE) as

$$MSE^k = \frac{1}{|\mathcal{V}|} \cdot \sum_{\mathrm{u} \in \mathcal{V}} [(\hat{f}_{\mathrm{u}}^k - \widetilde{f}_{\mathrm{u}}^k)^2]. \tag{2–4}$$

For the encoder, every pixel intensity f_{u}^k of the random input video sequence is a random variable. For any encoder with hybrid coding (see Fig. 2-1), the residual $\hat{e}_{\mathrm{u}}^k$, MV $\mathbf{mv}_{\mathrm{u}}^k$, and reconstructed pixel value $\hat{f}_{\mathrm{u}}^k$ are functions of f_{u}^k; so they are also random variables before motion estimation[6] . Given the Probability Mass Function (PMF) of $\hat{f}_{\mathrm{u}}^k$ and $\widetilde{f}_{\mathrm{u}}^k$, we define the transmission distortion for pixel $\mathbf{u}^k$ or PTD by

$$D_{\mathrm{u}}^k \triangleq E[(\hat{f}_{\mathrm{u}}^k - \widetilde{f}_{\mathrm{u}}^k)^2], \tag{2–5}$$

and we define the transmission distortion for the k-th frame or FTD by

$$D^k \triangleq E[\frac{1}{|\mathcal{V}|} \cdot \sum_{\mathrm{u} \in \mathcal{V}} (\hat{f}_{\mathrm{u}}^k - \widetilde{f}_{\mathrm{u}}^k)^2]. \tag{2–6}$$

[6] In applications such as cross-layer encoding rate control, distortion estimation for rate-distortion optimized bit allocation is required before motion estimation.

It is easy to prove that the relationship between FTD and PTD is characterized by

$$D^k = \frac{1}{|\mathcal{V}|} \cdot \sum_{\mathrm{u} \in \mathcal{V}} D_{\mathrm{u}}^k. \tag{2–7}$$

If the number of bits used to compress a frame is too large to be contained in one packet, the bits of the frame are split into multiple packets. In a time-varying channel, different packet of the same frame may experience different packet error probability (PEP). If pixel $\mathbf{u}^k$ and pixel $\mathbf{v}^k$ belong to different packets, the PMF of $\widetilde{f}_{\mathrm{u}}^k$ may be different from the PMF of $\widetilde{f}_{\mathrm{v}}^k$ even if $\hat{f}_{\mathrm{u}}^k$ and $\hat{f}_{\mathrm{v}}^k$ are identically distributed. In other words, D_{u}^k may be different from D_{v}^k even if pixel $\mathbf{u}^k$ and pixel $\mathbf{v}^k$ are in the neighboring MBs when FMO is activated. As a result, FTD D^k in (2–7) may be different from PTD D_{u}^k in (2–5). For this reason, we will derive formulae for both PTD and FTD, respectively. Note that most existing frame-level distortion models [8–11] assume that all pixels in the same frame experience the same channel condition and simply use (2–5) for FTD; however this assumption is not valid for high-resolution/high-quality video transmission over a time-varying channel.

In fact, (2–7) is a general form for distortions of all levels. If $|\mathcal{V}| = 1$, (2–7) reduces to (2–5). For slice/packet-level distortion, $\mathcal{V}$ is the set of the pixels contained in a slice/packet. For GOP-level distortion, $\mathcal{V}$ is the set of the pixels contained in a GOP. In this chapter, we only show how to derive formulae for PTD and FTD. Our methodology is also applicable to deriving formulae for slice/packet/GOP-level distortion by using appropriate $\mathcal{V}$.

2.2.4 Limitations of the Existing Transmission Distortion Models

In this subsection, we show that clipping noise has significant impact on transmission distortion, and neglect of clipping noise in existing models results in inaccurate estimation of transmission distortion. We define the clipping noise for pixel $\mathbf{u}^k$ at the encoder as

$$\hat{\Delta}_{\mathrm{u}}^k \triangleq (\hat{f}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-1} + \hat{e}_{\mathrm{u}}^k) - \Gamma(\hat{f}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-1} + \hat{e}_{\mathrm{u}}^k), \tag{2–8}$$

and the clipping noise for pixel $\mathbf{u}^k$ at the decoder as

$$\widetilde{\Delta}_{\mathrm{u}}^k \triangleq (\widetilde{f}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1} + \widetilde{e}_{\mathrm{u}}^k) - \Gamma(\widetilde{f}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1} + \widetilde{e}_{\mathrm{u}}^k). \tag{2–9}$$

Using (2–1), Eq. (2–8) becomes

$$\hat{f}_{\mathrm{u}}^k = \hat{f}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-1} + \hat{e}_{\mathrm{u}}^k - \hat{\Delta}_{\mathrm{u}}^k, \tag{2–10}$$

and using (2–3), Eq. (2–9) becomes

$$\widetilde{f}_{\mathrm{u}}^k = \widetilde{f}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1} + \widetilde{e}_{\mathrm{u}}^k - \widetilde{\Delta}_{\mathrm{u}}^k, \tag{2–11}$$

where $\hat{\Delta}_{\mathrm{u}}^k$ only depends on the video content and encoder structure, e.g., motion estimation, quantization, mode decision and clipping function; and $\widetilde{\Delta}_{\mathrm{u}}^k$ depends on not only the video content and encoder structure, but also channel conditions and decoder structure, e.g., error concealment and clipping function.

In most existing works, both $\hat{\Delta}_{\mathrm{u}}^k$ and $\widetilde{\Delta}_{\mathrm{u}}^k$ are neglected, i.e., these works assume $\hat{f}_{\mathrm{u}}^k = \hat{f}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-1} + \hat{e}_{\mathrm{u}}^k$ and $\widetilde{f}_{\mathrm{u}}^k = \widetilde{f}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1} + \widetilde{e}_{\mathrm{u}}^k$. However, this assumption is only valid for stored video or error-free communication. For error-prone communication, decoder clipping noise $\widetilde{\Delta}_{\mathrm{u}}^k$ has a significant impact on transmission distortion and hence should not be neglected. To illustrate this, Table 2-2 shows an example for the system in Fig. 2-1, where only the residual packet in the $(k-1)$-th frame is erroneous at the decoder (i.e., $\hat{e}_{\mathrm{v}}^{k-1}$ is erroneous), and all other residual packets and all the MV packets are error-free. Suppose the trajectory of pixel $\mathbf{u}^k$ in the $(k-1)$-th frame and $(k-2)$-th frame is specified by $\mathbf{v}^{k-1} = \mathbf{u}^k + \mathbf{mv}_{\mathrm{u}}^k$ and $\mathbf{w}^{k-2} = \mathbf{v}^{k-1} + \mathbf{mv}_{\mathrm{v}}^{k-1}$. Since $\hat{e}_{\mathrm{v}}^{k-1}$ is erroneous, the decoder needs to conceal the error; a simple concealment scheme is to let $\widetilde{e}_{\mathrm{v}}^{k-1} = 0$. From this example, we see that neglect of clipping noise (e.g., $\widetilde{\Delta}_{\mathrm{u}}^k = 45$) results in highly inaccurate estimate of distortion, e.g., the estimated distortion $\hat{D}_{\mathrm{u}}^k = 2500$ (without considering clipping) is much larger than the true distortion $D_{\mathrm{u}}^k = 25$. Note that if an MV is erroneous at the decoder, the pixel trajectory at the decoder will be different from the trajectory at

Table 2-2. An example that shows the effect of clipping noise on transmission distortion.

Encoder	Transmitted	$\hat{f}_{\mathrm{w}}^{k-2} = 250$	$\hat{e}_{\mathrm{v}}^{k-1} = -50$ (erroneous)	$\hat{e}_{\mathrm{u}}^{k} = 50$
	Reconstructed	$\hat{f}_{\mathrm{w}}^{k-2} = 250$	$\hat{f}_{\mathrm{v}}^{k-1} = \Gamma(\hat{f}_{\mathrm{w}}^{k-2} + \hat{e}_{\mathrm{v}}^{k-1}) = 200$	$\hat{f}_{\mathrm{u}}^{k} = \Gamma(\hat{f}_{\mathrm{v}}^{k-1} + \hat{e}_{\mathrm{u}}^{k}) = 250$
Decoder	Received	$\widetilde{f}_{\mathrm{w}}^{k-2} = 250$	$\widetilde{e}_{\mathrm{v}}^{k-1} = 0$ (concealed)	$\widetilde{e}_{\mathrm{u}}^{k} = 50$
	Reconstructed	$\widetilde{f}_{\mathrm{w}}^{k-2} = 250$	$\widetilde{f}_{\mathrm{v}}^{k-1} = \Gamma(\widetilde{f}_{\mathrm{w}}^{k-2} + \widetilde{e}_{\mathrm{v}}^{k-1}) = 250$	$\widetilde{f}_{\mathrm{u}}^{k} = \Gamma(\widetilde{f}_{\mathrm{v}}^{k-1} + \widetilde{e}_{\mathrm{u}}^{k}) = 255$
	Clipping noise	$\widetilde{\Delta}_{\mathrm{w}}^{k-2} = 0$	$\widetilde{\Delta}_{\mathrm{v}}^{k-1} = 0$	$\widetilde{\Delta}_{\mathrm{u}}^{k} = 45$
	Distortion	$D_{\mathrm{w}}^{k-2} = 0$	$D_{\mathrm{v}}^{k-1} = (\hat{f}_{\mathrm{v}}^{k-1} - \widetilde{f}_{\mathrm{v}}^{k-1})^2 = 2500$	$D_{\mathrm{u}}^{k} = (\hat{f}_{\mathrm{u}}^{k} - \widetilde{f}_{\mathrm{u}}^{k})^2 = 25$
Prediction without clipping	Received	$\widetilde{f}_{\mathrm{w}}^{k-2} = 250$	$\widetilde{e}_{\mathrm{v}}^{k-1} = 0$ (concealed)	$\widetilde{e}_{\mathrm{u}}^{k} = 50$
	Reconstructed	$\widetilde{f}_{\mathrm{w}}^{k-2} = 250$	$\widetilde{f}_{\mathrm{v}}^{k-1} = \widetilde{f}_{\mathrm{w}}^{k-2} + \widetilde{e}_{\mathrm{v}}^{k-1} = 250$	$\widetilde{f}_{\mathrm{u}}^{k} = \widetilde{f}_{\mathrm{v}}^{k-1} + \widetilde{e}_{\mathrm{u}}^{k} = 300$
	Distortion	$\hat{D}_{\mathrm{w}}^{k-2} = 0$	$\hat{D}_{\mathrm{v}}^{k-1} = (\hat{f}_{\mathrm{v}}^{k-1} - \widetilde{f}_{\mathrm{v}}^{k-1})^2 = 2500$	$\hat{D}_{\mathrm{u}}^{k} = (\hat{f}_{\mathrm{u}}^{k} - \widetilde{f}_{\mathrm{u}}^{k})^2 = 2500$

the encoder; then the resulting clipping noise $\widetilde{\Delta}_{\mathrm{u}}^{k}$ may be much larger than 45 as in this example, and hence the distortion estimation of the existing models without considering clipping may be much more inaccurate.

On the other hand, the encoder clipping noise $\hat{\Delta}_{\mathrm{u}}^{k}$ has negligible effect on quantization distortion and transmission distortion. This is due to two reasons: 1) the probability that $\hat{\Delta}_{\mathrm{u}}^{k} = 0$, is close to one, since the probability that $\gamma_L \leq \hat{f}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-1} + \hat{e}_{\mathrm{u}}^{k} \leq \gamma_H$, is close to one; 2) in case that $\hat{\Delta}_{\mathrm{u}}^{k} \neq 0$, $\hat{\Delta}_{\mathrm{u}}^{k}$ usually takes a value that is much smaller than the residuals. Since $\hat{\Delta}_{\mathrm{u}}^{k}$ is negligible, the clipping function can be removed at the encoder if only quantization distortion needs to be considered, e.g., for stored video or error-free communication. Since $\hat{\Delta}_{\mathrm{u}}^{k}$ is very likely to be a very small value, we would neglect it and assume $\hat{\Delta}_{\mathrm{u}}^{k} = 0$ in deriving our formula for transmission distortion.

2.3 Transmission Distortion Formulae

In this section, we derive formulae for PTD and FTD. The section is organized as below. Section 2.3.1 presents an overview of our approach to analyzing PTD and FTD. Then we elaborate on the derivation details in Section 2.3.2 through Section 2.3.5. Specifically, Section 2.3.2 quantifies the effect of RCE on transmission distortion; Section 2.3.3 quantifies the effect of MVCE on transmission distortion; Section 2.3.4 quantifies the effect of propagated error and clipping noise on transmission distortion;

Section 2.3.5 quantifies the effect of correlations (between any two of the error sources) on transmission distortion. Finally, Section 2.3.6 summarizes the key results of this chapter, i.e., the formulae for PTD and FTD.

2.3.1 Overview of the Approach to Analyzing PTD and FTD

To analyze PTD and FTD, we take a divide-and-conquer approach. We first divide transmission reconstructed error into four components: three independent random errors (RCE, MVCE and propagated error) based on their explicitly different root causes, and clipping noise, which is a non-linear function of those three random errors. This error decomposition allows us to further decompose transmission distortion into four terms, i.e., distortion caused by 1) RCE, 2) MVCE, 3) propagated error plus clipping noise, and 4) correlations between any two of the error sources, respectively. This distortion decomposition facilitates the derivation of a simple and accurate closed-form formula for each of the four distortion terms. Next, we elaborate on error decomposition and distortion decomposition.

Define transmission reconstructed error for pixel $\mathbf{u}^k$ by $\widetilde{\zeta}_{\mathrm{u}}^k \triangleq \hat{f}_{\mathrm{u}}^k - \widetilde{f}_{\mathrm{u}}^k$. From (2–10) and (2–11), we obtain

$$\begin{aligned}\widetilde{\zeta}_{\mathrm{u}}^k &= (\hat{e}_{\mathrm{u}}^k + \hat{f}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-1} - \hat{\Delta}_{\mathrm{u}}^k) - (\widetilde{e}_{\mathrm{u}}^k + \widetilde{f}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1} - \widetilde{\Delta}_{\mathrm{u}}^k) \\ &= (\hat{e}_{\mathrm{u}}^k - \widetilde{e}_{\mathrm{u}}^k) + (\hat{f}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-1} - \hat{f}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1}) + (\hat{f}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1} - \widetilde{f}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1}) - (\hat{\Delta}_{\mathrm{u}}^k - \widetilde{\Delta}_{\mathrm{u}}^k).\end{aligned} \tag{2–12}$$

Define RCE $\widetilde{\varepsilon}_{\mathrm{u}}^k$ by $\widetilde{\varepsilon}_{\mathrm{u}}^k \triangleq \hat{e}_{\mathrm{u}}^k - \widetilde{e}_{\mathrm{u}}^k$, and define MVCE $\widetilde{\xi}_{\mathrm{u}}^k$ by $\widetilde{\xi}_{\mathrm{u}}^k \triangleq \hat{f}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-1} - \hat{f}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1}$. Note that $\hat{f}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1} - \widetilde{f}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1} = \widetilde{\zeta}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1}$, which is the transmission reconstructed error of the concealed reference pixel in the reference frame; we call $\widetilde{\zeta}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1}$ propagated error. As mentioned in Section 2.2.4, we assume $\hat{\Delta}_{\mathrm{u}}^k = 0$. Therefore, (2–12) becomes

$$\widetilde{\zeta}_{\mathrm{u}}^k = \widetilde{\varepsilon}_{\mathrm{u}}^k + \widetilde{\xi}_{\mathrm{u}}^k + \widetilde{\zeta}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1} + \widetilde{\Delta}_{\mathrm{u}}^k. \tag{2–13}$$

(2–13) is our proposed **error decomposition**.

Combining (2–5) and (2–13), we have

$$\begin{aligned} D_{\mathrm{u}}^{k} &= E[(\widetilde{\varepsilon}_{\mathrm{u}}^{k} + \widetilde{\xi}_{\mathrm{u}}^{k} + \widetilde{\zeta}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1} + \widetilde{\Delta}_{\mathrm{u}}^{k})^2] \\ &= E[(\widetilde{\varepsilon}_{\mathrm{u}}^{k})^2] + E[(\widetilde{\xi}_{\mathrm{u}}^{k})^2] + E[(\widetilde{\zeta}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1} + \widetilde{\Delta}_{\mathrm{u}}^{k})^2] \\ &+ 2E[\widetilde{\varepsilon}_{\mathrm{u}}^{k} \cdot \widetilde{\xi}_{\mathrm{u}}^{k}] + 2E[\widetilde{\varepsilon}_{\mathrm{u}}^{k} \cdot (\widetilde{\zeta}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1} + \widetilde{\Delta}_{\mathrm{u}}^{k})] + 2E[\widetilde{\xi}_{\mathrm{u}}^{k} \cdot (\widetilde{\zeta}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1} + \widetilde{\Delta}_{\mathrm{u}}^{k})]. \end{aligned} \tag{2–14}$$

Denote $D_{\mathrm{u}}^{k}(r) \triangleq E[(\widetilde{\varepsilon}_{\mathrm{u}}^{k})^2]$, $D_{\mathrm{u}}^{k}(m) \triangleq E[(\widetilde{\xi}_{\mathrm{u}}^{k})^2]$, $D_{\mathrm{u}}^{k}(P) \triangleq E[(\widetilde{\zeta}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1} + \widetilde{\Delta}_{\mathrm{u}}^{k})^2]$ and $D_{\mathrm{u}}^{k}(c) \triangleq 2E[\widetilde{\varepsilon}_{\mathrm{u}}^{k} \cdot \widetilde{\xi}_{\mathrm{u}}^{k}] + 2E[\widetilde{\varepsilon}_{\mathrm{u}}^{k} \cdot (\widetilde{\zeta}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1} + \widetilde{\Delta}_{\mathrm{u}}^{k})] + 2E[\widetilde{\xi}_{\mathrm{u}}^{k} \cdot (\widetilde{\zeta}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1} + \widetilde{\Delta}_{\mathrm{u}}^{k})]$. Then, (2–14) becomes

$$D_{\mathrm{u}}^{k} = D_{\mathrm{u}}^{k}(r) + D_{\mathrm{u}}^{k}(m) + D_{\mathrm{u}}^{k}(P) + D_{\mathrm{u}}^{k}(c). \tag{2–15}$$

(3–15) is our proposed **distortion decomposition** for PTD. The reason why we combine propagated error and clipping noise into one term (called clipped propagated error) is because clipping noise is mainly caused by propagated error and such decomposition will simplify the formulae.

There are three major reasons for our decompositions in (2–13) and (3–15). First, if we directly substitute the terms in (2–5) by (2–10) and (2–11), it will produce 5 second moments and 10 cross-correlation terms (assuming $\hat{\Delta}_{\mathrm{u}}^{k} = 0$); since there are 8 possible error events due to three independent random errors, there are a total of $8 \times (5+10) = 120$ terms for PTD, making the analysis highly complicated. In contrast, our decompositions in (2–13) and (3–15) significantly simplify the analysis. Second, each term in (2–13) and (3–15) has a clear physical meaning, and therefore can be accurately estimated with low complexity. Third, such decompositions allow our formulae to be easily extended for supporting advanced video codec with more performance-enhanced parts, e.g., multi-reference prediction and interpolation filtering.

To derive the formula for FTD, from (2–7) and (3–15), we obtain

$$D^{k} = D^{k}(r) + D^{k}(m) + D^{k}(P) + D^{k}(c), \tag{2–16}$$

where

$$D^k(r) = \frac{1}{|\mathcal{V}|} \cdot \sum_{\mathrm{u}\in\mathcal{V}} D_{\mathrm{u}}^k(r), \tag{2–17}$$

$$D^k(m) = \frac{1}{|\mathcal{V}|} \cdot \sum_{\mathrm{u}\in\mathcal{V}} D_{\mathrm{u}}^k(m), \tag{2–18}$$

$$D^k(P) = \frac{1}{|\mathcal{V}|} \cdot \sum_{\mathrm{u}\in\mathcal{V}} D_{\mathrm{u}}^k(P), \tag{2–19}$$

$$D^k(c) = \frac{1}{|\mathcal{V}|} \cdot \sum_{\mathrm{u}\in\mathcal{V}} D_{\mathrm{u}}^k(c). \tag{2–20}$$

(2–16) is our proposed distortion decomposition for FTD.

Next, we present the derivation of a closed-form formula for each of the four distortion terms in Section 2.3.2 through Section 2.3.5.

2.3.2 Analysis of Distortion Caused by RCE

In this subsection, we first derive the pixel-level residual caused distortion $D_{\mathrm{u}}^k(r)$. Then we derive the frame-level residual caused distortion $D^k(r)$.

2.3.2.1 Pixel-level distortion caused by RCE

We denote S_{u}^k as the state indicator of whether there is transmission error for pixel $\mathbf{u}^k$ after channel decoding. Note that as mentioned in Section 2.2.1, both the residual channel and the MV channel contain channel decoding; hence in this chapter, the transmission error in the residual channel or the MV channel is meant to be the error uncorrectable by the channel decoding. To distinguish the residual error state and the MV error state, here we use $S_{\mathrm{u}}^k(r)$ to denote the residual error state for pixel $\mathbf{u}^k$. That is, $S_{\mathrm{u}}^k(r) = 1$ if $\hat{\mathrm{e}}_{\mathrm{u}}^k$ is received with error, and $S_{\mathrm{u}}^k(r) = 0$ if $\hat{\mathrm{e}}_{\mathrm{u}}^k$ is received without error. At the receiver, if there is no residual transmission error for pixel $\mathbf{u}$, $\widetilde{\mathrm{e}}_{\mathrm{u}}^k$ is equal to $\hat{\mathrm{e}}_{\mathrm{u}}^k$. However, if the residual packets are received with error, we need to conceal the residual error at the receiver. Denote $\check{\mathrm{e}}_{\mathrm{u}}^k$ the concealed residual when $S_{\mathrm{u}}^k(r) = 1$, and we have,

$$\widetilde{\mathrm{e}}_{\mathrm{u}}^k = \begin{cases} \check{\mathrm{e}}_{\mathrm{u}}^k, & S_{\mathrm{u}}^k(r) = 1 \\ \hat{\mathrm{e}}_{\mathrm{u}}^k, & S_{\mathrm{u}}^k(r) = 0. \end{cases} \tag{2–21}$$

Note that $\check{e}_u^k$ depends on $\hat{e}_u^k$ and the residual concealment method, but does not depend on the channel condition. From the definition of $\tilde{\varepsilon}_u^k$ and (2–21), we have

$$\begin{aligned}\tilde{\varepsilon}_u^k &= (\hat{e}_u^k - \check{e}_u^k) \cdot S_u^k(r) + (\hat{e}_u^k - \hat{e}_u^k) \cdot (1 - S_u^k(r)) \\ &= (\hat{e}_u^k - \check{e}_u^k) \cdot S_u^k(r).\end{aligned} \tag{2–22}$$

$\hat{e}_u^k$ depends on the input video sequence and the encoder structure, while $S_u^k(r)$ depends on communication system parameters such as delay bound, channel coding rate, transmission power, channel gain of the wireless channel. Under our framework shown in Fig. 2-1, the input video sequence and the encoder structure are independent of communication system parameters. Since $\hat{e}_u^k$ and $S_u^k(r)$ are solely caused by independent sources, we assume $\hat{e}_u^k$ and $S_u^k(r)$ are independent. That is, we make the following assumption.

Assumption 1. *$S_u^k(r)$ is independent of $\hat{e}_u^k$.*

Assumption 1 means that whether $\hat{e}_u^k$ will be correctly received or not, does not depend on the value of $\hat{e}_u^k$. Denote $\varepsilon_u^k \triangleq \hat{e}_u^k - \check{e}_u^k$; we have $\tilde{\varepsilon}_u^k = \varepsilon_u^k \cdot S_u^k(r)$. Denote $P_u^k(r)$ as the residual pixel error probability (XEP) for pixel $\mathbf{u}^k$, that is, $P_u^k(r) \triangleq P\{S_u^k(r) = 1\}$. Then, from (2–22) and Assumption 1, we have

$$D_u^k(r) = E[(\tilde{\varepsilon}_u^k)^2] = E[(\varepsilon_u^k)^2] \cdot E[(S_u^k(r))^2] = E[(\varepsilon_u^k)^2] \cdot (1 \cdot P_u^k(r)) = E[(\varepsilon_u^k)^2] \cdot P_u^k(r). \tag{2–23}$$

Hence, our formula for the pixel-level residual caused distortion is

$$D_u^k(r) = E[(\varepsilon_u^k)^2] \cdot P_u^k(r). \tag{2–24}$$

2.3.2.2 Frame-level distortion caused by RCE

To derive the frame-level residual caused distortion, the encoder needs to know the second moment of RCE for each pixel in that frame. However, if encoder knows the characteristics of residual process and concealment method, the formulae will be much simplified. One simple concealment method is to let $\check{e}_u^k = 0$ for all erroneous pixels.

A more general concealment method is to use the neighboring pixels to conceal an erroneous pixel. So we make the following assumption.

Assumption 2. *The residual $\hat{e}_{\mathbf{u}}^k$ is stationary with respect to (w.r.t.) 2D variable $\mathbf{u}$ in the same frame. In addition, $\check{e}_{\mathbf{u}}^k$ only depends on $\{\hat{e}_{\mathbf{v}}^k : \mathbf{v} \in \mathcal{N}_{\mathbf{u}}\}$ where $\mathcal{N}_{\mathbf{u}}$ is a fixed neighborhood of $\mathbf{u}$.*

In other words, Assumption 2 assumes that 1) $\hat{e}_{\mathbf{u}}^k$ is a 2D stationary stochastic process and the distribution of $\hat{e}_{\mathbf{u}}^k$ is the same for all $\mathbf{u} \in V^k$, and 2) $\check{e}_{\mathbf{u}}^k$ is also a 2D stationary stochastic process since it only depends on the neighboring $\hat{e}_{\mathbf{u}}^k$. Hence, $\hat{e}_{\mathbf{u}}^k - \check{e}_{\mathbf{u}}^k$ is also a 2D stationary stochastic process, and its second moment $E[(\hat{e}_{\mathbf{u}}^k - \check{e}_{\mathbf{u}}^k)^2] = E[(\varepsilon_{\mathbf{u}}^k)^2]$ is the same for all $\mathbf{u} \in V^k$. Therefore, we can drop $\mathbf{u}$ from the notation, and let $E[(\varepsilon^k)^2] = E[(\varepsilon_{\mathbf{u}}^k)^2]$ for all $\mathbf{u} \in V^k$.

Denote $N_i^k(r)$ as the number of pixels contained in the i-th residual packet of the k-th frame; denote $P_i^k(r)$ as PEP of the i-th residual packet of the k-th frame; denote $N^k(r)$ as the total number of residual packets of the k-th frame. Since for all pixels in the same packet, the residual XEP is equal to its PEP, from (2–17) and (3–16), we have

$$\begin{aligned} D^k(r) &= \frac{1}{|\mathcal{V}|} \sum_{\mathbf{u} \in \mathcal{V}^k} E[(\varepsilon_{\mathbf{u}}^k)^2] \cdot P_{\mathbf{u}}^k(r) && (2\text{–}25) \\ &= \frac{1}{|\mathcal{V}|} \sum_{\mathbf{u} \in \mathcal{V}^k} E[(\varepsilon^k)^2] \cdot P_{\mathbf{u}}^k(r) && (2\text{–}26) \\ &\overset{(a)}{=} \frac{E[(\varepsilon^k)^2]}{|\mathcal{V}|} \sum_{i=1}^{N^k(r)} (P_i^k(r) \cdot N_i^k(r)) && (2\text{–}27) \\ &\overset{(b)}{=} E[(\varepsilon^k)^2] \cdot \bar{P}^k(r). && (2\text{–}28) \end{aligned}$$

where (a) is due to $P_{\mathbf{u}}^k(r) = P_i^k(r)$ for pixel $\mathbf{u}$ in the i-th residual packet; (b) is due to

$$\bar{P}^k(r) \triangleq \frac{1}{|\mathcal{V}|} \sum_{i=1}^{N^k(r)} (P_i^k(r) \cdot N_i^k(r)). \qquad (2\text{–}29)$$

$\bar{P}^k(r)$ is a weighted average over PEPs of all residual packets in the k-th frame, in which different packets may contain different numbers of pixels. Hence, our formula for the

frame-level residual caused distortion is

$$D^k(r) = E[(\varepsilon^k)^2] \cdot \bar{P}^k(r). \tag{2–30}$$

2.3.3 Analysis of Distortion Caused by MVCE

Similar to the derivations in Section 2.3.2.1, in this subsection, we derive the formula for the pixel-level MV caused distortion $D_{\mathrm{u}}^k(m)$, and the frame-level MV caused distortion $D^k(m)$.

2.3.3.1 Pixel-level distortion caused by MVCE

Denote the MV error state for pixel $\mathbf{u}^k$ by $S_{\mathrm{u}}^k(m)$, and denote the concealed MV by $\check{\mathbf{mv}}_{\mathrm{u}}^k$ when $S_{\mathrm{u}}^k(m) = 1$. Therefore, we have

$$\widetilde{\mathbf{mv}}_{\mathrm{u}}^k = \begin{cases} \check{\mathbf{mv}}_{\mathrm{u}}^k, & S_{\mathrm{u}}^k(m) = 1 \\ \mathbf{mv}_{\mathrm{u}}^k, & S_{\mathrm{u}}^k(m) = 0. \end{cases} \tag{2–31}$$

Here, we use the temporal error concealment [21] to conceal MV errors. Denote $\xi_{\mathrm{u}}^k \triangleq \hat{f}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-1} - \hat{f}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1}$; we have $\widetilde{\xi}_{\mathrm{u}}^k = \xi_{\mathrm{u}}^k \cdot S_{\mathrm{u}}^k(m)$, where ξ_{u}^k depends on the accuracy of MV concealment, and the spatial correlation between reference pixel and concealed reference pixel at the encoder. Denote $P_{\mathrm{u}}^k(m)$ as the MV XEP for pixel $\mathbf{u}^k$, that is, $P_{\mathrm{u}}^k(m) \triangleq P\{S_{\mathrm{u}}^k(m) = 1\}$. We make the following assumption.

Assumption 3. *$S_{\mathrm{u}}^k(m)$ is independent of ξ_{u}^k.*

Following the same deriving process in Section 2.3.2.1, we can obtain

$$D_{\mathrm{u}}^k(m) = E[(\xi_{\mathrm{u}}^k)^2] \cdot P_{\mathrm{u}}^k(m). \tag{2–32}$$

Note that ξ_{u}^k depends on $\mathbf{mv}_{\mathrm{u}}^k$ and the MV concealment method, but does not depend on the channel condition. In most cases, given the concealment method, the statistics of ξ_{u}^k can be easily obtained at the encoder. From the experiments, we observe that ξ_{u}^k follows a zero-mean Laplacian distribution.

Note that in H.264 specification, there is no slice data partitioning for an instantaneous decoding refresh (IDR) frame [22]; so $S_{\mathrm{u}}^k(r)$ and $S_{\mathrm{u}}^k(m)$ are fully correlated in an IDR-frame, that is, $S_{\mathrm{u}}^k(r) = S_{\mathrm{u}}^k(m)$, and hence $P_{\mathrm{u}}^k(r) = P_{\mathrm{u}}^k(m)$. This is also true for I-MB, and P-MB without slice data partitioning. For P-MB with slice data partitioning in H.264, $S_{\mathrm{u}}^k(r)$ and $S_{\mathrm{u}}^k(m)$ are partially correlated. In other words, if the packet of slice data partition A, which contains MV information, is lost, the corresponding packet of slice data partition B, which contains residual information, cannot be decoded even if it is correctly received, since there is no slice header in the slice data partition B. Therefore, the residual channel and the MV channel in Fig. 2-1 are actually correlated if the encoder follows H.264 specification. In this chapter, we study transmission distortion in a more general case where $S_{\mathrm{u}}^k(r)$ and $S_{\mathrm{u}}^k(m)$ can be either independent or correlated.[7]

2.3.3.2 Frame-level distortion caused by MVCE

To derive the frame-level MV caused distortion, we make the following assumption.

Assumption 4. *The second moment of ξ_{u}^k is the same for all $\mathbf{u} \in V^k$.*

Under Assumption 4, we can drop $\mathbf{u}$ from the notation, and let $E[(\xi^k)^2] = E[(\xi_{\mathrm{u}}^k)^2]$ for all $\mathbf{u} \in V^k$. Denote $N_i^k(m)$ as the number of pixels contained in the i-th MV packet of the k-th frame; denote $P_i^k(m)$ as PEP of the i-th MV packet of the k-th frame; denote $N^k(m)$ as the total number of MV packets of the k-th frame. Following the same derivation process in Section 2.3.2.2, we obtain the frame-level MV caused distortion for the k-th frame as below

$$D^k(m) = E[(\xi^k)^2] \cdot \bar{P}^k(m), \tag{2–33}$$

[7] To achieve this, we change the H.264 reference code JM14.0 by allowing residual packets to be used for decoder without the corresponding MV packets being correctly received, that is, $\hat{e}_{\mathrm{u}}^k$ can be used to reconstruct $\widetilde{f}_{\mathrm{u}}^k$ even if $\mathbf{mv}_{\mathrm{u}}^k$ is not correctly received.

where $\bar{P}^k(m) \triangleq \frac{1}{|\mathcal{V}|}\sum_{i=1}^{N_m^k}(P_i^k(m) \cdot N_i^k(m))$, a weighted average over PEPs of all MV packets in the k-th frame, in which different packets may contain different numbers of pixels.

2.3.4 Analysis of Distortion Caused by Propagated Error Plus Clipping Noise

In this subsection, we derive the distortion caused by error propagation in a non-linear decoder with clipping. We first derive the pixel-level propagation and clipping caused distortion $D_{\mathrm{u}}^k(P)$. Then we derive the frame-level propagation and clipping caused distortion $D^k(P)$.

2.3.4.1 Pixel-level distortion caused by propagated error plus clipping noise

First, we analyze the pixel-level propagation and clipping caused distortion $D_{\mathrm{u}}^k(P)$ in P-MBs. $D_{\mathrm{u}}^k(P)$ depends on propagated error and clipping noise; and clipping noise depends on propagated error, RCE, and MVCE. Hence, $D_{\mathrm{u}}^k(P)$ depends on propagated error, RCE, and MVCE. Let r, m, p denote the event of occurrence of RCE, MV concealment error and propagated error, respectively, and let $\bar{r}, \bar{m}, \bar{p}$ denote logical NOT of r, m, p respectively (indicating no error). We use a triplet to denote the joint event of three types of error; e.g., $\{r, m, p\}$ denotes the event that all the three types of errors occur, and $\mathbf{u}^k\{\bar{r}, \bar{m}, \bar{p}\}$ denotes the pixel $\mathbf{u}^k$ experiencing none of the three types of errors.

When we analyze the condition that several error events may occur, the notation could be simplified by the principle of formal logic. For example, $\widetilde{\Delta}_{\mathrm{u}}^k\{\bar{r}, \bar{m}\}$ denotes the clipping noise under the condition that there is neither RCE nor MVCE for pixel $\mathbf{u}^k$, while it is not certain whether the reference pixel has error. Correspondingly, denote $P_{\mathrm{u}}^k\{\bar{r}, \bar{m}\}$ as the probability of event $\{\bar{r}, \bar{m}\}$, that is, $P_{\mathrm{u}}^k\{\bar{r}, \bar{m}\} = P\{S_{\mathrm{u}}^k(r) = 0 \text{ and } S_{\mathrm{u}}^k(m) = 0\}$. From the definition of $P_{\mathrm{u}}^k(r)$, the marginal probability $P_{\mathrm{u}}^k\{r\} = P_{\mathrm{u}}^k(r)$ and the marginal probability $P_{\mathrm{u}}^k\{\bar{r}\} = 1 - P_{\mathrm{u}}^k(r)$. The same, $P_{\mathrm{u}}^k\{m\} = P_{\mathrm{u}}^k(m)$ and $P_{\mathrm{u}}^k\{\bar{m}\} = 1 - P_{\mathrm{u}}^k(m)$.

Define $D_{\mathrm{u}}^k(p) \triangleq E[(\widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-1} + \widetilde{\Delta}_{\mathrm{u}}^k\{\bar{r}, \bar{m}\})^2]$; and define $\alpha_{\mathrm{u}}^k \triangleq \frac{D_{\mathrm{u}}^k(p)}{D_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-1}}$, which is called propagation factor for pixel $\mathbf{u}^k$. The propagation factor α_{u}^k defined in this chapter is

different from the propagation factor [11], leakage [8], or attenuation factor [16], which are modeled as the effect of spatial filtering or intra update; our propagation factor α_{u}^{k} is also different from the fading factor [9], which is modeled as the effect of using fraction of referenced pixels in the reference frame for motion prediction. Note that $D_{\mathrm{u}}^{k}(p)$ is only a special case of $D_{\mathrm{u}}^{k}(P)$ under the error event of $\{\bar{r}, \bar{m}\}$ for pixel $\mathbf{u}^{k}$. However, most existing models inappropriately use their propagation factor, obtained under the error event of $\{\bar{r}, \bar{m}\}$, to replace $D_{\mathrm{u}}^{k}(P)$ of all other error events directly without distinguishing their difference.

To calculate $E[(\widetilde{\zeta}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1} + \widetilde{\Delta}_{\mathrm{u}}^{k})^2]$ in (2–14), we need to analyze $\widetilde{\Delta}_{\mathrm{u}}^{k}$ in four different error events for pixel $\mathbf{u}^{k}$: 1) both residual and MV are erroneous, denoted by $\mathbf{u}^{k}\{r, m\}$; 2) residual is erroneous but MV is correct, denoted by $\mathbf{u}^{k}\{r, \bar{m}\}$; 3) residual is correct but MV is erroneous, denoted by $\mathbf{u}^{k}\{\bar{r}, m\}$; and 4) both residual and MV are correct, denoted by $\mathbf{u}^{k}\{\bar{r}, \bar{m}\}$. So,

$$\begin{aligned} D_{\mathrm{u}}^{k}(P) = {} & P_{\mathrm{u}}^{k}\{r, m\} \cdot E[(\widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1} + \widetilde{\Delta}_{\mathrm{u}}^{k}\{r, m\})^2] + P_{\mathrm{u}}^{k}\{r, \bar{m}\} \cdot E[(\widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-1} + \widetilde{\Delta}_{\mathrm{u}}^{k}\{r, \bar{m}\})^2] \\ & + P_{\mathrm{u}}^{k}\{\bar{r}, m\} \cdot E[(\widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1} + \widetilde{\Delta}_{\mathrm{u}}^{k}\{\bar{r}, m\})^2] + P_{\mathrm{u}}^{k}\{\bar{r}, \bar{m}\} \cdot E[(\widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-1} + \widetilde{\Delta}_{\mathrm{u}}^{k}\{\bar{r}, \bar{m}\})^2]. \end{aligned} \tag{2–34}$$

Note that the concealed pixel value should be in the clipping function range, that is, $\Gamma(\widetilde{f}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1} + \check{e}_{\mathrm{u}}^{k}) = \widetilde{f}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1} + \check{e}_{\mathrm{u}}^{k}$, so $\widetilde{\Delta}_{\mathrm{u}}^{k}\{r\} = 0$. Also note that if the MV channel is independent of the residual channel, we have $P_{\mathrm{u}}^{k}\{r, m\} = P_{\mathrm{u}}^{k}(r) \cdot P_{\mathrm{u}}^{k}(m)$. However, as mentioned in Section 2.3.3.1, in H.264 specification, these two channels are correlated. In other words, $P_{\mathrm{u}}^{k}\{\bar{r}, m\} = 0$ and $P_{\mathrm{u}}^{k}\{\bar{r}, \bar{m}\} = P_{\mathrm{u}}^{k}\{\bar{r}\}$ for P-MBs with slice data partitioning in H.264. In such a case, (2–34) is simplified to

$$D_{\mathrm{u}}^{k}(P) = P_{\mathrm{u}}^{k}\{r, m\} \cdot D_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1} + P_{\mathrm{u}}^{k}\{r, \bar{m}\} \cdot D_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-1} + P_{\mathrm{u}}^{k}\{\bar{r}\} \cdot D_{\mathrm{u}}^{k}(p). \tag{2–35}$$

In a more general case, where $P_{\mathrm{u}}^{k}\{\bar{r}, m\} \neq 0$, Eq. (2–35) is still valid. This is because $P_{\mathrm{u}}^{k}\{\bar{r}, m\} \neq 0$ only happens under slice data partitioning condition, where

$P_{\mathrm{u}}^{k}\{\bar{r}, m\} \ll P_{\mathrm{u}}^{k}\{\bar{r}, \bar{m}\}$ and $E[(\widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1} + \widetilde{\Delta}_{\mathrm{u}}^{k}\{\bar{r}, \bar{m}\})^2] \approx E[(\widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1} + \widetilde{\Delta}_{\mathrm{u}}^{k}\{\bar{r}, m\})^2]$ under UEP. Therefore, the last two terms in (2–34) is almost equal to $P_{\mathrm{u}}^{k}\{\bar{r}\} \cdot D_{\mathrm{u}}^{k}(p)$.

Note that for P-MB without slice data partitioning in H.264, we have $P_{\mathrm{u}}^{k}\{r, \bar{m}\} = P_{\mathrm{u}}^{k}\{\bar{r}, m\} = 0$, $P_{\mathrm{u}}^{k}\{r, m\} = P_{\mathrm{u}}^{k}\{r\} = P_{\mathrm{u}}^{k}\{m\} = P_{\mathrm{u}}^{k}$, and $P_{\mathrm{u}}^{k}\{\bar{r}, \bar{m}\} = P_{\mathrm{u}}^{k}\{\bar{r}\} = P_{\mathrm{u}}^{k}\{\bar{m}\} = 1 - P_{\mathrm{u}}^{k}$. Therefore, (2–35) can be further simplified to

$$D_{\mathrm{u}}^{k}(P) = P_{\mathrm{u}}^{k} \cdot D_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1} + (1 - P_{\mathrm{u}}^{k}) \cdot D_{\mathrm{u}}^{k}(p). \tag{2–36}$$

Also note that for I-MB, there will be no transmission distortion if it is correctly received, that is, $D_{\mathrm{u}}^{k}(p) = 0$. So (3–18) can be further simplified to

$$D_{\mathrm{u}}^{k}(P) = P_{\mathrm{u}}^{k} \cdot D_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1}. \tag{2–37}$$

Comparing (2–37) with (3–18), we see that I-MB is a special case of P-MB with $D_{\mathrm{u}}^{k}(p) = 0$, that is, the propagation factor $\alpha_{\mathrm{u}}^{k} = 0$ according to the definition. It is important to note that $D_{\mathrm{u}}^{k}(P) > 0$ for I-MB. In other words, I-MB also contains the distortion caused by propagation error since $P_{\mathrm{u}}^{k} \neq 0$. However, existing LTI models [8, 9] assume that there is no distortion caused by propagation error for I-MB, which under-estimates the transmission distortion.

In the following part of this subsection, we derive the propagation factor α_{u}^{k} for P-MB and prove some important properties of clipping noise. To derive α_{u}^{k}, we first give Lemma 1 as below.

Lemma 1. *Given the PMF of the random variable $\widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1}$ and the value of $\hat{f}_{\mathrm{u}}^{k}$, $D_{\mathrm{u}}^{k}(p)$ can be calculated at the encoder by $D_{\mathrm{u}}^{k}(p) = E[\Phi^2(\widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1}, \hat{f}_{\mathrm{u}}^{k})]$, where $\Phi(x, y)$ is called error reduction function and defined by*

$$\Phi(x, y) \triangleq y - \Gamma(y - x) = \begin{cases} y - \gamma_L, & y - x < \gamma_L \\ x, & \gamma_L \leq y - x \leq \gamma_H \\ y - \gamma_H, & y - x > \gamma_H. \end{cases} \tag{2–38}$$

Lemma 1 is proved in Appendix A.1. In fact, we have found in our experiments that in any error event, $\widetilde{\zeta}^{k-1}_{\mathrm{u+mv}^k_{\mathrm{u}}}$ approximately follows Laplacian distribution with zero mean. If we assume $\widetilde{\zeta}^{k-1}_{\mathrm{u+mv}^k_{\mathrm{u}}}$ follows Laplacian distribution with zero mean, the calculation for $D^k_{\mathrm{u}}(p)$ becomes simpler since the only unknown parameter for PMF of $\widetilde{\zeta}^{k-1}_{\mathrm{u+mv}^k_{\mathrm{u}}}$ is its variance. Under this assumption, we have the following proposition.

Proposition 1. *The propagation factor α for propagated error with Laplacian distribution of zero-mean and variance σ^2 is given by*

$$\alpha = 1 - \frac{1}{2}e^{-\frac{y-\gamma_L}{b}}(\frac{y-\gamma_L}{b}+1) - \frac{1}{2}e^{-\frac{\gamma_H-y}{b}}(\frac{\gamma_H-y}{b}+1), \tag{2–39}$$

where y is the reconstructed pixel value, and $b = \frac{\sqrt{2}}{2}\sigma$.

Proposition 1 is proved in Appendix A.2. In the zero-mean Laplacian case, α^k_{u} will only be a function of $\hat{f}^k_{\mathrm{u}}$ and the variance of $\widetilde{\zeta}^{k-1}_{\mathrm{u+mv}^k_{\mathrm{u}}}$, which is equal to $D^{k-1}_{\mathrm{u+mv}^k_{\mathrm{u}}}$ in this case. Since $D^{k-1}_{\mathrm{u+mv}^k_{\mathrm{u}}}$ has already been calculated during the phase of predicting the $(k-1)$-th frame transmission distortion, $D^k_{\mathrm{u}}(p)$ can be calculated by $D^k_{\mathrm{u}}(p) = \alpha^k_{\mathrm{u}} \cdot D^{k-1}_{\mathrm{u+mv}^k_{\mathrm{u}}}$ via the definition of α^k_{u}. Then we can recursively calculate $D^k_{\mathrm{u}}(P)$ in (2–35) since both $D^{k-1}_{\mathrm{u+mv}^k_{\mathrm{u}}}$ and $D^{k-1}_{\mathrm{u+mv}^k_{\mathrm{u}}}$ have been calculated previously for the $(k-1)$-th frame. (3–8) is very important for designing a low complexity algorithm to estimate propagation and clipping caused distortion in FTD, which will be presented in Chapter 3.

Next, we prove an important property of the non-linear clipping function in the following proposition.

Proposition 2. *Clipping reduces propagated error, that is, $D^k_{\mathrm{u}}(p) \leq D^{k-1}_{\mathrm{u+mv}^k_{\mathrm{u}}}$, or $\alpha^k_{\mathrm{u}} \leq 1$.*

Proof. First, from Lemma 5, which is presented and proved in Appendix A.6, we have $\Phi^2(x, y) \leq x^2$ for any $\gamma_L \leq y \leq \gamma_H$. In other words, the function $\Phi(x, y)$ reduces the energy of propagated error. This is the reason why we call it error reduction function. With Lemma 1, it is straightforward to prove that whatever the PMF of $\widetilde{\zeta}^{k-1}_{\mathrm{u+mv}^k_{\mathrm{u}}}$ is, $E[\Phi^2(\widetilde{\zeta}^{k-1}_{\mathrm{u+mv}^k_{\mathrm{u}}}, \hat{f}^k_{\mathrm{u}})] \leq E[(\widetilde{\zeta}^{k-1}_{\mathrm{u+mv}^k_{\mathrm{u}}})^2]$, that is, $D^k_{\mathrm{u}}(p) \leq D^{k-1}_{\mathrm{u+mv}^k_{\mathrm{u}}}$, which is equivalent to $\alpha^k_{\mathrm{u}} \leq 1$. □

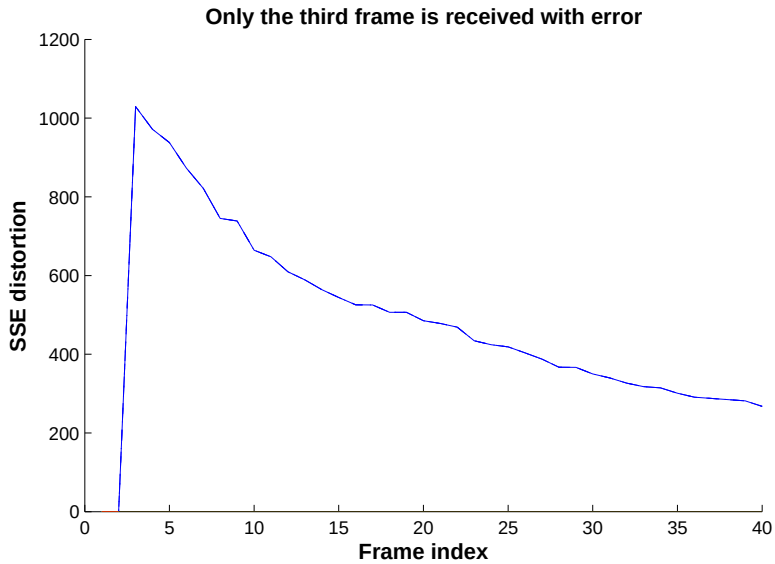


Figure 2-2. The effect of clipping noise on distortion propagation.

Proposition 2 tells us that if there is no newly induced errors in the k-th frame, transmission distortion decreases from the $(k-1)$-th frame to the k-th frame. Fig. 2-2 shows the experimental result of transmission distortion propagation for bus_cif.yuv, where transmission errors only occur in the third frame.

In fact, if we consider the more general cases where there may be new error induced in the k-th frame, we can still prove that $E[(\widetilde{\zeta}^{k-1}_{\mathrm{u}+\widetilde{\mathrm{mv}}^k_{\mathrm{u}}} + \widetilde{\Delta}^k_{\mathrm{u}})^2] \leq E[(\widetilde{\zeta}^{k-1}_{\mathrm{u}+\widetilde{\mathrm{mv}}^k_{\mathrm{u}}})^2]$ using the proof for the following corollary.

Corollary 1. *The correlation coefficient between $\widetilde{\zeta}^{k-1}_{\mathrm{u}+\widetilde{\mathrm{mv}}^k_{\mathrm{u}}}$ and $\widetilde{\Delta}^k_{\mathrm{u}}$ is non-positive. Specifically, they are negatively correlated under the condition $\{\bar{r}, p\}$, and uncorrelated under other conditions.*

Corollary 1 is proved in Appendix A.8. This property is very important for designing a low complexity algorithm to estimate propagation and clipping caused distortion in PTD, which will be presented in Chapter 3.

2.3.4.2 Frame-level distortion caused by propagated error plus clipping noise

In (2–35), $D^{k-1}_{\mathrm{u}+\check{\mathrm{mv}}^k_{\mathrm{u}}} \neq D^{k-1}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}}$ due to the non-stationarity of the error process over space. However, both the sum of $D^{k-1}_{\mathrm{u}+\check{\mathrm{mv}}^k_{\mathrm{u}}}$ over all pixels in the $(k-1)$-th frame and

the sum of $D^{k-1}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}}$ over all pixels in the $(k-1)$-th frame will converge to D^{k-1} due to the randomness of MV. The formula for frame-level propagation and clipping caused distortion is given in Lemma 2.

Lemma 2. *The frame-level propagation and clipping caused distortion in the k-th frame is*

$$D^k(P) = D^{k-1} \cdot \bar{P}^k(r) + D^k(p) \cdot (1 - \bar{P}^k(r))(1 - \beta^k), \tag{2–40}$$

where $D^k(p) \triangleq \frac{1}{|\mathcal{V}|} \sum_{\mathrm{u} \in \mathcal{V}^k} D^k_{\mathrm{u}}(p)$ and $\bar{P}^k(r)$ is defined in (2–29); β^k is the percentage of I-MBs in the k-th frame; D^{k-1} is the transmission distortion in the $(k-1)$-th frame.

Lemma 2 is proved in Appendix A.3. Define the propagation factor for the k-th frame $\alpha^k \triangleq \frac{D^k(p)}{D^{k-1}}$; then we have $\alpha^k = \frac{\sum_{\mathrm{u} \in \mathcal{V}^k} \alpha^k_{\mathrm{u}} \cdot D^{k-1}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}}}{D^{k-1}}$. Note that $D^{k-1}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}}$ may be different for different pixels in the $(k-1)$-th frame due to the non-stationarity of error process over space. However, when the number of pixels in the $(k-1)$-th frame is sufficiently large, the sum of $D^{k-1}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}}$ over all the pixels in the $(k-1)$-th frame will converge to D^{k-1}. Therefore, we have $\alpha^k = \frac{\sum_{\mathrm{u} \in \mathcal{V}^k} \alpha^k_{\mathrm{u}} \cdot D^{k-1}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}}}{\sum_{\mathrm{u} \in \mathcal{V}^k} D^{k-1}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}}}$, which is a weighted average of α^k_{u} with the weight being $D^{k-1}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}}$. As a result, $D^k(p) \leq D^k(P)$. When the number of pixels in the (k-1)-th frame is small, $\sum_{\mathrm{u} \in \mathcal{V}^k} \alpha^k_{\mathrm{u}} \cdot D^{k-1}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}}$ may be larger than D^{k-1} although its probability is small as observed in our experiments. However, most existing works directly use $D^k(P) = D^k(p)$ in predicting transmission distortion. This is another reason why LTI models [8, 9] under-estimate transmission distortion when there is no MV error. Details will be discussed in Section 2.4.2.

From Proposition 1, we know that α^k_{u} is a function of $\hat{f}^k_{\mathrm{u}}$. So, α^k depends on all samples of $\hat{f}^k_{\mathrm{u}}$ in the k-th frame. Since the samples of $\hat{f}^k_{\mathrm{u}}$ usually change over frames due to the video content variation, the propagation factor α^k also varies from frame to frame as observed in the experiments. Accurately estimating α^k for each frame is very important for instantaneous distortion estimation. However, existing models assume

propagation factor is constant over all frames, which makes the distortion estimation inaccurate. We will discuss how to accurately estimate α^k in real time in Chapter 3.

2.3.5 Analysis of Correlation Caused Distortion

In this subsection, we first derive the pixel-level correlation caused distortion $D_{\mathrm{u}}^k(c)$. Then we derive the frame-level correlation caused distortion $D^k(c)$.

2.3.5.1 Pixel-level correlation caused distortion

We analyze the correlation caused distortion $D_{\mathrm{u}}^k(c)$ at the decoder in four different cases: i) for $\mathbf{u}^k\{\bar{r}, \bar{m}\}$, both $\widetilde{\varepsilon}_{\mathrm{u}}^k = 0$ and $\widetilde{\xi}_{\mathrm{u}}^k = 0$, so $D_{\mathrm{u}}^k(c) = 0$; ii) for $\mathbf{u}^k\{r, \bar{m}\}$, $\widetilde{\xi}_{\mathrm{u}}^k = 0$ and $D_{\mathrm{u}}^k(c) = 2E[\varepsilon_{\mathrm{u}}^k \cdot (\widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-1} + \widetilde{\Delta}_{\mathrm{u}}^k\{r, \bar{m}\})]$; iii) for $\mathbf{u}^k\{\bar{r}, m\}$, $\widetilde{\varepsilon}_{\mathrm{u}}^k = 0$ and $D_{\mathrm{u}}^k(c) = 2E[\xi_{\mathrm{u}}^k \cdot (\widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{m}}\mathrm{v}_{\mathrm{u}}^k}^{k-1} + \widetilde{\Delta}_{\mathrm{u}}^k\{\bar{r}, m\})]$; iv) for $\mathbf{u}^k\{r, m\}$, $D_{\mathrm{u}}^k(c) = 2E[\varepsilon_{\mathrm{u}}^k \cdot \xi_{\mathrm{u}}^k] + 2E[\varepsilon_{\mathrm{u}}^k \cdot (\widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{m}}\mathrm{v}_{\mathrm{u}}^k}^{k-1} + \widetilde{\Delta}_{\mathrm{u}}^k\{r, m\})] + 2E[\xi_{\mathrm{u}}^k \cdot (\widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{m}}\mathrm{v}_{\mathrm{u}}^k}^{k-1} + \widetilde{\Delta}_{\mathrm{u}}^k\{r, m\})]$. From Section 2.3.4.1, we know $\widetilde{\Delta}_{\mathrm{u}}^k\{r\} = 0$. So, we obtain

$$\begin{aligned} D_{\mathrm{u}}^k(c) = {} & P_{\mathrm{u}}^k\{r, \bar{m}\} \cdot 2E[\varepsilon_{\mathrm{u}}^k \cdot \widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-1}] + P_{\mathrm{u}}^k\{\bar{r}, m\} \cdot 2E[\xi_{\mathrm{u}}^k \cdot (\widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{m}}\mathrm{v}_{\mathrm{u}}^k}^{k-1} + \widetilde{\Delta}_{\mathrm{u}}^k\{\bar{r}, m\})] \\ & + P_{\mathrm{u}}^k\{r, m\} \cdot (2E[\varepsilon_{\mathrm{u}}^k \cdot \xi_{\mathrm{u}}^k] + 2E[\varepsilon_{\mathrm{u}}^k \cdot \widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{m}}\mathrm{v}_{\mathrm{u}}^k}^{k-1}] + 2E[\xi_{\mathrm{u}}^k \cdot \widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{m}}\mathrm{v}_{\mathrm{u}}^k}^{k-1}]). \end{aligned} \tag{2–41}$$

In the experiments, we find that in the trajectory of pixel $\mathbf{u}^k$, 1) the residual $\hat{\mathrm{e}}_{\mathrm{u}}^k$ is approximately uncorrelated with the residual in all other frames $\hat{\mathrm{e}}_{\mathrm{v}}^i$, where $i \neq k$, as shown in Fig. 2-3; and 2) the residual $\hat{\mathrm{e}}_{\mathrm{u}}^k$ is approximately uncorrelated with the MVCE of the corresponding pixel ξ_{u}^k and the MVCE in all previous frames ξ_{v}^i, where $i < k$, as shown in Fig. 2-4. Based on the above observations, we further assume that for any $i < k$, $\hat{\mathrm{e}}_{\mathrm{u}}^k$ is uncorrelated with $\hat{\mathrm{e}}_{\mathrm{v}}^i$ and ξ_{v}^i if $\mathbf{v}^i$ is not in the trajectory of pixel $\mathbf{u}^k$, and make the following assumption.

Assumption 5. *$\hat{\mathrm{e}}_{\mathrm{u}}^k$ is uncorrelated with ξ_{u}^k, and is uncorrelated with both $\hat{\mathrm{e}}_{\mathrm{v}}^i$ and ξ_{v}^i for any $i < k$.*

Since $\widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-1}$ and $\widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{m}}\mathrm{v}_{\mathrm{u}}^k}^{k-1}$ are the transmission reconstructed errors accumulated from all the frames before the k-th frame, $\varepsilon_{\mathrm{u}}^k$ is uncorrelated with $\widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-1}$ and $\widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{m}}\mathrm{v}_{\mathrm{u}}^k}^{k-1}$ due to

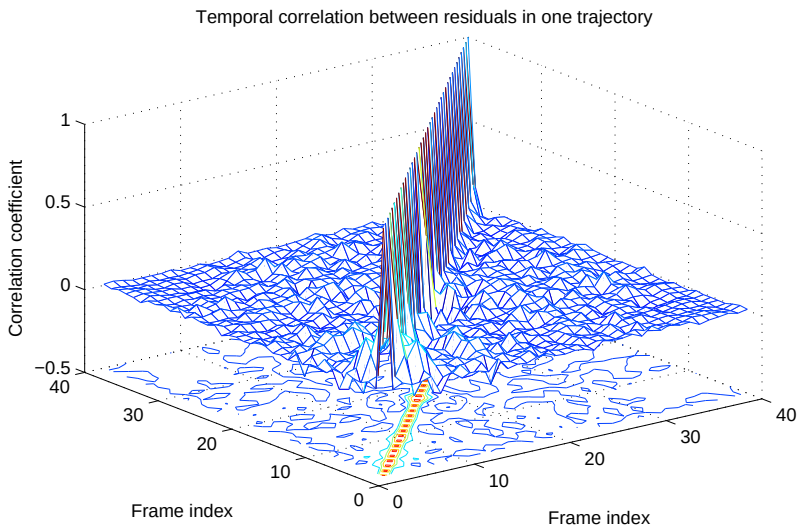


Figure 2-3. Temporal correlation between the residuals in one trajectory.

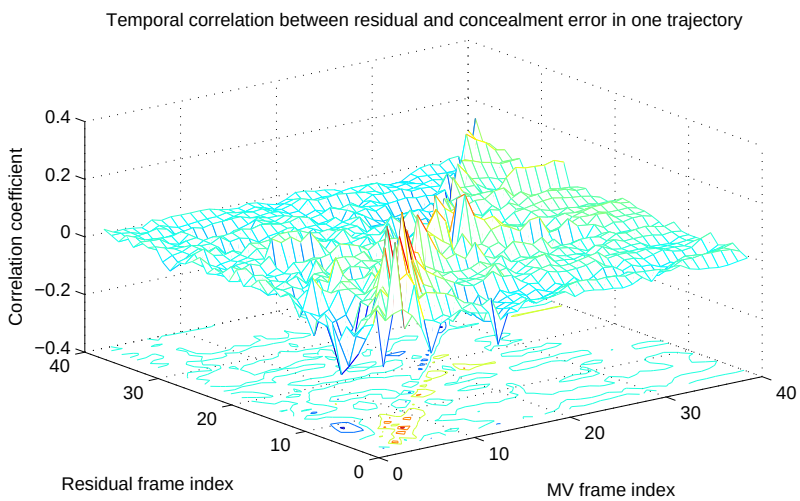


Figure 2-4. Temporal correlation matrix between residual and MVCE in one trajectory.

Assumption 5. Thus, (2–41) becomes

$$D_{\mathrm{u}}^{k}(c) = 2P_{\mathrm{u}}^{k}\{m\} \cdot E[\xi_{\mathrm{u}}^{k} \cdot \widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{m}}\mathrm{v}_{\mathrm{u}}^{k}}^{k-1}] + 2P_{\mathrm{u}}^{k}\{\bar{r}, m\} \cdot E[\xi_{\mathrm{u}}^{k} \cdot \widetilde{\Delta}_{\mathrm{u}}^{k}\{\bar{r}, m\}]. \tag{2–42}$$

However, we observe that in the trajectory of pixel $\mathbf{u}^k$, 1) the residual $\hat{e}_{\mathrm{u}}^{k}$ is correlated with the MVCE ξ_{v}^{i}, where $i > k$, as seen in Fig. 2-4; and 2) the MVCE ξ_{u}^{k} is highly correlated with the MVCE ξ_{v}^{i} as shown in Fig. 2-5. This interesting phenomenon could be exploited by an error concealment algorithm and is subject to our future study.

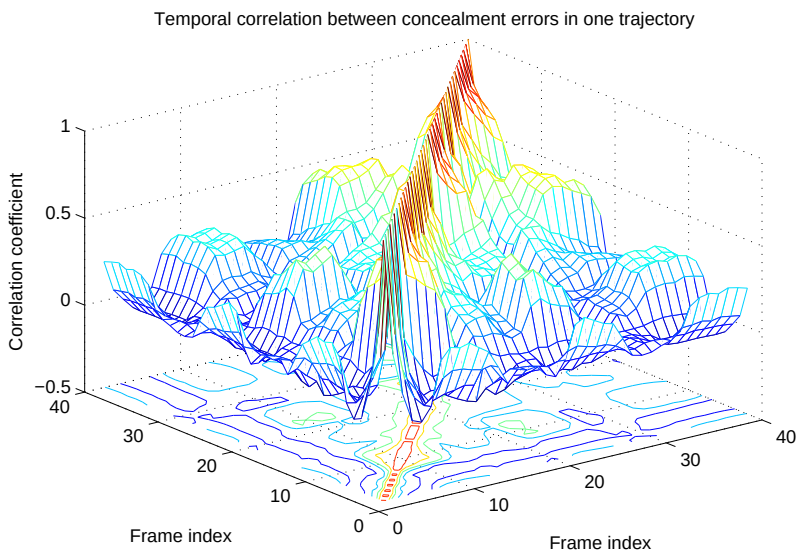


Figure 2-5. Temporal correlation matrix between MVCEs in one trajectory.

As mentioned in Section 2.3.4.1, for P-MBs with slice data partitioning in H.264, $P_{\mathrm{u}}^{k}\{\bar{r}, m\} = 0$. So, (2–42) becomes

$$D_{\mathrm{u}}^{k}(c) = 2P_{\mathrm{u}}^{k}\{m\} \cdot E[\xi_{\mathrm{u}}^{k} \cdot (\hat{f}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1} - \widetilde{f}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1})]. \tag{2–43}$$

Note that in the more general case that $P_{\mathrm{u}}^{k}\{\bar{r}, m\} \neq 0$, Eq. (2–43) is still valid since ξ_{u}^{k} is almost uncorrelated with $\widetilde{\Delta}_{\mathrm{u}}^{k}\{\bar{r}, m\}$ as observed in the experiment.

For I-MBs or P-MBs without slice data partitioning in H.264, since $P_{\mathrm{u}}^{k}\{r, \bar{m}\} = P_{\mathrm{u}}^{k}\{\bar{r}, m\} = 0$ and $P_{\mathrm{u}}^{k}\{r, m\} = P_{\mathrm{u}}^{k}\{r\} = P_{\mathrm{u}}^{k}\{m\} = P_{\mathrm{u}}^{k}$ as mentioned in Section 2.3.4.1, (2–41) can be simplified to

$$D_{\mathrm{u}}^{k}(c) = 2P_{\mathrm{u}}^{k} \cdot (2E[\varepsilon_{\mathrm{u}}^{k} \cdot \xi_{\mathrm{u}}^{k}] + 2E[\varepsilon_{\mathrm{u}}^{k} \cdot \widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1}] + 2E[\xi_{\mathrm{u}}^{k} \cdot \widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1}]). \tag{2–44}$$

Under Assumption 5, (3–19) reduces to (2–43).

Define $\lambda_{\mathrm{u}}^{k} \triangleq \frac{E[\xi_{\mathrm{u}}^{k} \cdot \widetilde{f}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1}]}{E[\xi_{\mathrm{u}}^{k} \cdot \hat{f}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1}]}$; λ_{u}^{k} is a correlation ratio, that is, the ratio of the correlation between MVCE and concealed reference pixel value at the receiver, to the correlation between MVCE and concealed reference pixel value at the transmitter. λ_{u}^{k} quantifies the effect of the correlation between the MVCE and propagated error on transmission

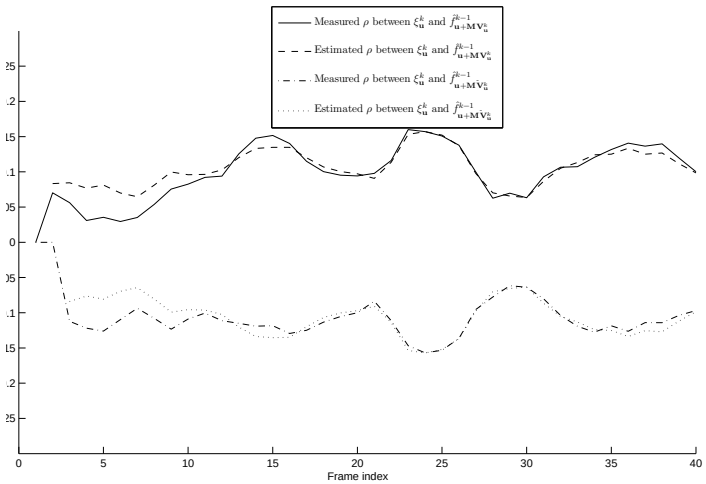


Figure 2-6. Comparison between measured and estimated correlation coefficients.

distortion. Since λ_{u}^k is a stable statistics of MV, estimating λ_{u}^k is much simpler and more accurate than estimating $E[\xi_{\mathrm{u}}^k \cdot \widetilde{f}^{k-1}_{\mathrm{u}+\check{\mathrm{m}}\mathrm{v}_{\mathrm{u}}^k}]$ directly, thereby resulting in more accurate distortion estimate. The details on how to estimate λ_{u}^k will be presented in Chapter 3.

Although we do not know the exact value of λ_{u}^k at the encoder, its range is

$$\prod_{i=1}^{k-1} P^i_{\mathrm{T}(i)}\{\bar{r}, \bar{m}\} \leq \lambda_{\mathrm{u}}^k \leq 1, \tag{2–45}$$

where $\mathbf{T}(i)$ is the pixel position of the i-th frame in the trajectory, for example, $\mathbf{T}(k-1) = \mathbf{u}^k + \mathbf{mv}_{\mathrm{u}}^k$ and $\mathbf{T}(k-2) = \mathbf{v}^{k-1} + \mathbf{mv}_{\mathrm{v}}^{k-1}$. The left inequality in (2–45) holds in the extreme case that any error in the trajectory will cause ξ_{u}^k and $\widetilde{f}^{k-1}_{\mathrm{u}+\check{\mathrm{m}}\mathrm{v}_{\mathrm{u}}^k}$ to be uncorrelated, which is usually true for high motion video. The right inequality in (2–45) holds in another extreme case that all errors in the trajectory do not affect the correlation between ξ_{u}^k and $\widetilde{f}^{k-1}_{\mathrm{u}+\check{\mathrm{m}}\mathrm{v}_{\mathrm{u}}^k}$, which is usually true for low motion video.

Using the definition of λ_{u}^k, (2–43) becomes

$$D_{\mathrm{u}}^k(c) = 2P_{\mathrm{u}}^k\{m\} \cdot (1 - \lambda_{\mathrm{u}}^k) \cdot E[\xi_{\mathrm{u}}^k \cdot \hat{f}^{k-1}_{\mathrm{u}+\check{\mathrm{m}}\mathrm{v}_{\mathrm{u}}^k}]. \tag{2–46}$$

In our experiments, we observe an interesting phenomenon that ξ_{u}^k is always positively correlated with $\hat{f}^{k-1}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}$, and negatively correlated with $\hat{f}^{k-1}_{\mathrm{u}+\check{\mathrm{m}}\mathrm{v}_{\mathrm{u}}^k}$. This is theoretically

proved in Lemma 3 under Assumption 6; and this is also verified by our experiments as shown in Fig. 2-6.

Assumption 6. $E[(\hat{f}^{k-1}_{\mathrm{u}+\check{\mathrm{m}}\mathrm{v}^k_{\mathrm{u}}})^2] = E[(\hat{f}^{k-1}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}})^2]$.

Assumption 6 is valid under the condition that the distance between $\mathbf{mv}^k_{\mathrm{u}}$ and $\check{\mathbf{m}}\mathbf{v}^k_{\mathrm{u}}$ is small; this is also verified by our experiments.

Lemma 3. *Under Assumption 6,* $E[\xi^k_{\mathrm{u}} \cdot \hat{f}^{k-1}_{\mathrm{u}+\check{\mathrm{m}}\mathrm{v}^k_{\mathrm{u}}}] = -\frac{E[(\xi^k_{\mathrm{u}})^2]}{2}$ *and* $E[\xi^k_{\mathrm{u}} \cdot \hat{f}^{k-1}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}}] = \frac{E[(\xi^k_{\mathrm{u}})^2]}{2}$.

Lemma 3 is proved in Appendix A.4. Under Assumption 6, using Lemma 3, we further simplify (2–46) as below.

$$D^k_{\mathrm{u}}(c) = (\lambda^k_{\mathrm{u}} - 1) \cdot E[(\xi^k_{\mathrm{u}})^2] \cdot P^k_{\mathrm{u}}(m). \tag{2–47}$$

From (3–17), we know that $E[(\xi^k_{\mathrm{u}})^2] \cdot P^k_{\mathrm{u}}(m)$ is exactly equal to $D^k_{\mathrm{u}}(m)$. Therefore, (2–47) is further simplified to

$$D^k_{\mathrm{u}}(c) = (\lambda^k_{\mathrm{u}} - 1) \cdot D^k_{\mathrm{u}}(m). \tag{2–48}$$

As mentioned in Section 2.3.3.1, we observe that ξ^k_{u} follows a zero-mean Laplacian distribution in the experiment. Denote ρ the correlation coefficient between ξ^k_{u} and $\hat{f}^{k-1}_{\mathrm{u}+\check{\mathrm{m}}\mathrm{v}^k_{\mathrm{u}}}$. If we assume $E[\xi^k_{\mathrm{u}}] = 0$, we have $\rho = \frac{E[\xi^k_{\mathrm{u}} \cdot \hat{f}^{k-1}_{\mathrm{u}+\check{\mathrm{m}}\mathrm{v}^k_{\mathrm{u}}}] - E[\xi^k_{\mathrm{u}}] \cdot E[\hat{f}^{k-1}_{\mathrm{u}+\check{\mathrm{m}}\mathrm{v}^k_{\mathrm{u}}}]}{\sigma_{\xi^k_{\mathrm{u}}} \cdot \sigma_{\hat{f}^k_{\mathrm{u}}}} = -\frac{\sigma_{\xi^k_{\mathrm{u}}}}{2\sigma_{\hat{f}^k_{\mathrm{u}}}}$. Similarly, it is easy to prove that the correlation coefficient between ξ^k_{u} and $\hat{f}^{k-1}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}}$ is $\frac{\sigma_{\xi^k_{\mathrm{u}}}}{2\sigma_{\hat{f}^k_{\mathrm{u}}}}$. This agrees well with the experimental results shown in Fig. 2-6. Via the same derivation process, one can obtain the correlation coefficient between $\hat{\mathrm{e}}^k_{\mathrm{u}}$ and $\hat{f}^{k-1}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}}$, and between $\hat{\mathrm{e}}^k_{\mathrm{u}}$ and $\hat{f}^k_{\mathrm{u}}$. One possible application of these correlation properties is error concealment with partial information available.

2.3.5.2 Frame-Level correlation caused distortion

Denote $V_i^k(m)$ the set of pixels in the i-th MV packet of the k-th frame. From (2–20), (2–47) and Assumption 4, we obtain

$$\begin{aligned} D^k(c) &= \frac{E[(\xi^k)^2]}{|\mathcal{V}|} \sum_{\mathrm{u}\in\mathcal{V}^k} (\lambda_{\mathrm{u}}^k - 1) \cdot P_{\mathrm{u}}^k(m) \\ &= \frac{E[(\xi^k)^2]}{|\mathcal{V}|} \sum_{i=1}^{N^k(m)} \{P_i^k(m) \sum_{\mathrm{u}\in\mathcal{V}_i^k(m)} (\lambda_{\mathrm{u}}^k - 1)\}. \end{aligned} \tag{2–49}$$

Define $\lambda^k \triangleq \frac{1}{|\mathcal{V}|}\sum_{\mathrm{u}\in V^k} \lambda_{\mathrm{u}}^k$; due to the randomness of $\mathbf{mv}_{\mathrm{u}}^k$, $\frac{1}{N_i^k(m)}\sum_{\mathrm{u}\in\mathcal{V}_i^k\{m\}} \lambda_{\mathrm{u}}^k$ will converge to λ^k for any packet that contains a sufficiently large number of pixels. By rearranging (2–49), we obtain

$$\begin{aligned} D^k(c) &= \frac{E[(\xi^k)^2]}{|\mathcal{V}|} \sum_{i=1}^{N^k(m)} \{P_i^k(m) \cdot N_i^k(m) \cdot (\lambda^k - 1)\} \\ &= (\lambda^k - 1) \cdot E[(\xi^k)^2] \cdot \bar{P}^k(m). \end{aligned} \tag{2–50}$$

From (3–3), we know that $E[(\xi^k)^2] \cdot \bar{P}^k(m)$ is exactly equal to $D^k(m)$. Therefore, (2–50) is further simplified to

$$D^k(c) = (\lambda^k - 1) \cdot D^k(m). \tag{2–51}$$

2.3.6 Summary

In Section 2.3.1, we decomposed transmission distortion into four terms; we derived a formula for each term in Sections 2.3.2 through 2.3.5. In this section, we combine the formulae for the four terms into a single formula.

2.3.6.1 Pixel-Level transmission distortion

Theorem 2.1. *Under single-reference prediction, the PTD of pixel* $\mathbf{u}^k$ *is*

$$D_{\mathrm{u}}^k = D_{\mathrm{u}}^k(r) + \lambda_{\mathrm{u}}^k \cdot D_{\mathrm{u}}^k(m) + P_{\mathrm{u}}^k\{r, m\} \cdot D_{\mathrm{u}+\tilde{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1} + P_{\mathrm{u}}^k\{r, \bar{m}\} \cdot D_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-1} + P_{\mathrm{u}}^k\{\bar{r}\} \cdot \alpha_{\mathrm{u}}^k \cdot D_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-1}. \tag{2–52}$$

Proof. (2–52) can be obtained by plugging (3–16), (3–17), (2–35), and (2–47) into (3–15). □

Corollary 2. *Under single-reference prediction and no slice data partitioning,* (2–52) *is simplified to*

$$D_{\mathrm{u}}^{k} = P_{\mathrm{u}}^{k} \cdot (E[(\varepsilon_{\mathrm{u}}^{k})^2] + \lambda_{\mathrm{u}}^{k} \cdot E[(\xi_{\mathrm{u}}^{k})^2] + D_{\mathrm{u}+\check{\mathrm{m}}\mathrm{v}_{\mathrm{u}}^{k}}^{k-1}) + (1 - P_{\mathrm{u}}^{k}) \cdot \alpha_{\mathrm{u}}^{k} \cdot D_{\mathrm{u}+\mathrm{m}\mathrm{v}_{\mathrm{u}}^{k}}^{k-1}. \tag{2–53}$$

2.3.6.2 Frame-Level transmission distortion

Theorem 2.2. *Under single-reference prediction, the FTD of the k-th frame is*

$$D^{k} = D^{k}(r) + \lambda^{k} \cdot D^{k}(m) + \bar{P}^{k}(r) \cdot D^{k-1} + (1 - \bar{P}^{k}(r)) \cdot D^{k}(p) \cdot (1 - \beta^{k}). \tag{2–54}$$

Proof. (2–54) can be obtained by plugging (3–2), (3–3), (2–40) and (2–51) into (2–16). □

Corollary 3. *Under single-reference prediction and no slice data partitioning, the FTD of the k-th frame is given by* (2–54)[8] *.*

2.4 Relationship between Theorem 2.2 and Existing Transmission Distortion Models

As mentioned previously, some existing works have addressed the problem of transmission distortion prediction, and they proposed several different models [8], [9], [16], [11] to estimate transmission distortion. In this section, we will identify the relationship between Theorem 2.2 and their models, and specify the conditions, under which those models are accurate. Note that in order to demonstrate the effect of non-linear clipping on transmission distortion propagation, we disable intra update, that is, $\beta^{k} = 0$ for all the following cases.

[8] The same formula for both cases is because both mean of $D_{\mathrm{u}+\check{\mathrm{m}}\mathrm{v}_{\mathrm{u}}^{k}}^{k-1}$ and mean of $D_{\mathrm{u}+\mathrm{m}\mathrm{v}_{\mathrm{u}}^{k}}^{k-1}$ converge to D^{k-1} when the number of pixels in the k-th frame is sufficiently large, as seen in Appendix A.3.

2.4.1 Case 1: Only the $(k-1)$-th Frame Has Error, and the Subsequent Frames are All Correctly Received

In this case, the models proposed in Ref. [8] [11] state that when there is no intra coding and spatial filtering, the propagation distortion will be the same for all the frames after the $(k-1)$-th frame, i.e., $D^n(p) = D^{n-1}$ ($\forall\ n \geq k$). However, this is not true as we proved in Proposition 2. Due to the clipping function, we have $\alpha^n \leq 1$ ($\forall\ n \geq k$), i.e., $D^n \leq D^{n-1}$ ($\forall\ n \geq k$) in case the n-th frame is error-free. Actually, from Appendix A.6, we know that the equality only holds under a very special case that $\hat{f}^k_{\mathrm{u}} - \gamma_H \leq \widetilde{\zeta}^{k-1}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}} \leq \hat{f}^k_{\mathrm{u}} - \gamma_L$ for all pixel $\mathbf{u} \in V^k$.

2.4.2 Case 2: Burst Errors in Consecutive Frames

In Ref. [16], authors observe that the transmission distortion caused by accumulated errors from consecutive frames is generally larger than the sum of those distortions caused by individual frame errors. This is also observed in our experiment when there is no MV error. To explain this phenomenon, let us first look at a simple case that residuals in the k-th frame are all erroneous, while the MVs in the k-th frame are all correctly received. In this case, we obtain from (2–54) that $D^k = D^k(r) + \bar{P}^k(r) \cdot D^{k-1} + (1 - \bar{P}^k(r)) \cdot D^k(p)$, which is larger than the simple sum $D^k(r) + D^k(p)$ as in the LTI model; the under-estimation caused by the LTI model is due to $D^k - (D^k(r) + D^k(p)) = (1-\alpha^k) \cdot \bar{P}^k(r) \cdot D^{k-1}$.

However, when MV is erroneous, the experimental result is quite different from that claimed in Ref. [16] especially for the high motion video. In other words, the LTI model now causes over-estimation for a burst error channel. In this case, the predicted transmission distortion can be calculated via (2–54) in Theorem 2.2 as $D^k_1 = D^k(r) + \lambda^k \cdot D^k(m) + \bar{P}^k(r) \cdot D^{k-1}_1 + (1 - \bar{P}^k(r)) \cdot \alpha^k \cdot D^{k-1}_1$, and by the LTI model as $D^k_2 = D^k(r) + D^k(m) + \alpha^k \cdot D^{k-1}_2$. So, the prediction difference between Theorem 2.2 and the

LTI model is

$$D_1^k - D_2^k = (1-\alpha^k)\cdot \bar{P}^k(r)\cdot D_1^{k-1} - (1-\lambda^k)\cdot \bar{P}^k(m)\cdot E[(\xi^k)^2] + \alpha^k\cdot(D_1^{k-1} - D_2^{k-1}). \tag{2–55}$$

At the beginning, $D_1^0 = D_2^0 = 0$, and $D^{k-1} << E[(\xi^k)^2]$ when k is small. Therefore, the transmission distortion caused by accumulated errors from consecutive frames will be smaller than the sum of the distortions caused by individual frame errors, that is, $D_1^k < D_2^k$. We may see from (2–55) that, due to the propagation of over-estimation $D_1^{k-1} - D_2^{k-1}$ from the $(k-1)$-th frame to the k-th frame, the accumulated difference between D_1^k and D_2^k will become larger and larger as k increases.

2.4.3 Case 3: Modeling Transmission Distortion as an Output of an LTI System with PEP as input

In Ref. [9], authors propose an LTI transmission distortion model based on their observations from experiments. This LTI model ignores the effects of correlation between the newly induced error and the propagated error, that is, $\lambda^k = 1$. This is only valid for low motion video. From (2–54), we obtain

$$D^k = D^k(r) + D^k(m) + (\bar{P}^k(r) + (1-\bar{P}^k(r))\cdot\alpha^k)\cdot D^{k-1}. \tag{2–56}$$

Let $\eta^k = \bar{P}^k(r) + (1-\bar{P}^k(r))\cdot\alpha^k$. If 1) there is no slice data partitioning, i.e., $P^k(m) = P^k(r) = P^k$, and 2) $\bar{P}^k(r) = P^k(r)$ (which means one frame is transmitted in one packet, or different packets experience the same channel condition), then (2–56) becomes $D^k = \{E[(\xi^k)^2] + E[(\varepsilon^k)^2]\}\cdot P^k + \eta^k\cdot D^{k-1}$. Let $E^k \triangleq E[(\xi^k)^2] + E[(\varepsilon^k)^2]$. Then the recursive formula results in

$$D^k = \sum_{l=k-L}^{k}[(\prod_{i=l+1}^{k}\eta^i)\cdot(E^l\cdot P^l)], \tag{2–57}$$

where L is the time interval between the k-th frame and the latest correctly received frame.

Denote the system by an operator H that maps the error input sequence $\{P^k\}$, as a function of frame index k, to the distortion output sequence $\{D^k\}$. Since generally $D^k(p)$ is a nonlinear function of D^{k-1}, as a ratio of $D^k(p)$ and D^{k-1}, α^k is still a function of D^{k-1}. As a result, η^k is a function of D^{k-1}. That means the operator H is non-linear, i.e., the system is non-linear. In addition, since α^k varies from frame to frame as mentioned in Section 2.3.4.2, the system is time-variant. In summary, H is generally a non-linear time-variant system.

The LTI model assumes that 1) the operator H is linear, that is, $H(a \cdot P_1^k + b \cdot P_2^k) = a \cdot H(P_1^k) + b \cdot H(P_2^k)$, which is valid only when η^k does not depend on D^{k-1}; and 2) the operator H is time-invariant, that is, $D^{k+\delta} = H(P^{k+\delta})$, which is valid only when η^k is constant, i.e., both $P^k(r)$ and α^k are constant. Under these two assumptions, we have $\eta^i = \eta$, and we obtain $\prod_{i=l+1}^{k} \eta^i = (\eta)^{k-l}$. Let $h[k] = (\eta)^k$, where $h[k]$ is the impulse response of the LTI model; then we obtain

$$D^k = \sum_{l=k-L}^{k} [h[k-l] \cdot (E^l \cdot P^l)]. \tag{2–58}$$

From Proposition 2, it is easy to prove that $0 \leq \eta \leq 1$; so $h[k]$ is a decreasing function of time. We see that (2–58) is a convolution between the error input sequence and the system impulse response. Actually, if we let $h[k] = e^{-\gamma k}$, where $\gamma = -\log \eta$, it is exactly the formula proposed in Ref. [9]. Note that (2–58) is a very special case of (2–54) with the following limitations: 1) the video content has to be of low motion; 2) there is no slice data partitioning or all pixels in the same frame experience the same channel condition; 3) η^k is a constant, that is, both $\bar{P}^k(r)$ and the propagation factor α^k are constant, which requires the probability distributions of reconstructed pixel values in all frames should be the same. Note that the physical meaning of η^k is not the actual propagation factor, but it is just a notation for simplifying the formula.

2.5 PTD and FTD under Multi-Reference Prediction

The PTD and FTD formulae in Section 2.3 are for single-reference prediction. In this section, we extend the formulae to multi-reference prediction.

2.5.1 Pixel-level Distortion under Multi-Reference Prediction

If multiple frames are allowed to be the references for motion estimation, the reconstructed pixel value at the encoder in (2–1) becomes

$$\hat{f}_{\mathrm{u}}^{k} = \Gamma(\hat{f}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-j} + \hat{e}_{\mathrm{u}}^{k}). \tag{2–59}$$

For the reconstructed pixel value at the decoder in (2–3), it is a bit different as below.

$$\widetilde{f}_{\mathrm{u}}^{k} = \Gamma(\widetilde{f}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-j'} + \widetilde{e}_{\mathrm{u}}^{k}). \tag{2–60}$$

If $\mathbf{mv}_{\mathrm{u}}^{k}$ is correctly received, $\widetilde{\mathbf{mv}}_{\mathrm{u}}^{k} = \mathbf{mv}_{\mathrm{u}}^{k}$ and $\widetilde{f}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-j'} = \widetilde{f}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-j}$. However, if $\mathbf{mv}_{\mathrm{u}}^{k}$ is received with error, the concealed MV has no difference from the single-reference case, that is, $\widetilde{\mathbf{mv}}_{\mathrm{u}}^{k} = \check{\mathbf{mv}}_{\mathrm{u}}^{k}$ and $\widetilde{f}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-j'} = \widetilde{f}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1}$.

As a result, (2–12) becomes

$$\begin{aligned}\widetilde{\zeta}_{\mathrm{u}}^{k} &= (\hat{e}_{\mathrm{u}}^{k} + \hat{f}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-j} - \hat{\Delta}_{\mathrm{u}}^{k}) - (\widetilde{e}_{\mathrm{u}}^{k} + \widetilde{f}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-j'} - \widetilde{\Delta}_{\mathrm{u}}^{k}) \\ &= (\hat{e}_{\mathrm{u}}^{k} - \widetilde{e}_{\mathrm{u}}^{k}) + (\hat{f}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-j} - \hat{f}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-j'}) + (\hat{f}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-j'} - \widetilde{f}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-j'}) - (\hat{\Delta}_{\mathrm{u}}^{k} - \widetilde{\Delta}_{\mathrm{u}}^{k}).\end{aligned} \tag{2–61}$$

Following the same deriving process from Section 2.3.1 to Section 2.3.5, the formulae for PTD under multi-reference prediction are the same as those under single-reference prediction except the following changes: 1) MVCE $\widetilde{\xi}_{\mathrm{u}}^{k} \triangleq \hat{f}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-j} - \hat{f}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-j'}$ and clipping noise $\widetilde{\Delta}_{\mathrm{u}}^{k} \triangleq (\widetilde{f}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-j'} + \widetilde{e}_{\mathrm{u}}^{k}) - \Gamma(\widetilde{f}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-j'} + \widetilde{e}_{\mathrm{u}}^{k})$; 2) $D_{\mathrm{u}}^{k}(m)$ and $D_{\mathrm{u}}^{k}(c)$ are given by (3–17) and (2–48), respectively, with a new definition of $\xi_{\mathrm{u}}^{k} \triangleq \hat{f}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-j} - \hat{f}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1}$; 3) $D_{\mathrm{u}}^{k}(p) \triangleq E[(\widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-j} + \widetilde{\Delta}_{\mathrm{u}}^{k}\{\bar{r}, \bar{m}\})^2]$, $\alpha_{\mathrm{u}}^{k} \triangleq \frac{D_{\mathrm{u}}^{k}(p)}{D_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-j}}$ and

$$D_{\mathrm{u}}^{k}(P) = P_{\mathrm{u}}^{k}\{r, m\} \cdot D_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1} + P_{\mathrm{u}}^{k}\{r, \bar{m}\} \cdot D_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-j} + P_{\mathrm{u}}^{k}\{\bar{r}\} \cdot D_{\mathrm{u}}^{k}(p), \tag{2–62}$$

compared to (2–35). The generalization of PTD formulae to multi-reference prediction is straightforward since the multi-reference prediction case just has a larger set of reference pixels than the single-reference case. Therefore, we have the following general theorem for PTD.

Theorem 2.3. *Under multi-reference prediction, the PTD of pixel* $\mathbf{u}^k$ *is*

$$D_{\mathrm{u}}^k = D_{\mathrm{u}}^k(r) + \lambda_{\mathrm{u}}^k \cdot D_{\mathrm{u}}^k(m) + P_{\mathrm{u}}^k\{r, m\} \cdot D_{\mathrm{u}+\check{\mathrm{m}}\mathrm{v}_{\mathrm{u}}^k}^{k-1} + P_{\mathrm{u}}^k\{r, \bar{m}\} \cdot D_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-j} + P_{\mathrm{u}}^k\{\bar{r}\} \cdot \alpha_{\mathrm{u}}^k \cdot D_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-j}. \tag{2–63}$$

Corollary 4. *Under multi-reference prediction and no slice data partitioning,* (2–63) *is simplified to*

$$D_{\mathrm{u}}^k = P_{\mathrm{u}}^k \cdot (E[(\varepsilon_{\mathrm{u}}^k)^2] + \lambda_{\mathrm{u}}^k \cdot E[(\xi_{\mathrm{u}}^k)^2] + D_{\mathrm{u}+\check{\mathrm{m}}\mathrm{v}_{\mathrm{u}}^k}^{k-1}) + (1 - P_{\mathrm{u}}^k) \cdot \alpha_{\mathrm{u}}^k \cdot D_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-j}. \tag{2–64}$$

2.5.2 Frame-level Distortion under Multi-Reference Prediction

Under multi-reference prediction, each block typically is allowed to choose its reference block independently; hence, different pixels in the same frame may have different reference frames. Define $V^k(j) \triangleq \{\mathbf{u}^k : \mathbf{u}^k = \mathbf{v}^{k-j} - \mathbf{mv}_{\mathrm{u}}^k\}$, where $j \in \{1, 2, ..., J\}$ and J is the number of reference frames; i.e., $V^k(j)$ is the set of the pixels in the k-th frame, whose reference pixels are in the $(k-j)$-th frame. Obviously, $\bigcup_{j=1}^{J} V^k(j) = V^k$ and $\bigcap_{j=1}^{J} V^k(j) = \varnothing$. Define $w^k(j) \triangleq \frac{|\mathcal{V}^k(j)|}{|\mathcal{V}^k|}$. Note that V^k and $V^k(j)$ have the similar physical meanings but only the different cardinalities.

$D^k(m)$ and $D^k(c)$ are given by (3–3) and (2–51), respectively, with a new definition of $\xi^k(j) \triangleq \{\xi_{\mathrm{u}}^k : \xi_{\mathrm{u}}^k = \hat{f}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-j} - \hat{f}_{\mathrm{u}+\check{\mathrm{m}}\mathrm{v}_{\mathrm{u}}^k}^{k-1}\}$. and $\xi^k = \sum_{j=1}^{J} w^k(j) \cdot \xi^k(j)$. Define the propagation factor of $V^k(j)$ by $\alpha^k(j) \triangleq \frac{\sum_{\mathrm{u} \in V^k(j)} \alpha_{\mathrm{u}}^k \cdot D_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-j}}{\sum_{\mathrm{u} \in V^k(j)} D_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-j}}$. The following lemma gives the formula for $D^k(P)$.

Lemma 4. *The frame-level propagation and clipping caused distortion in the k-th frame for the multi-reference case is*

$$D^k(P) = D^{k-1} \cdot \bar{P}^k\{r, m\} + \sum_{j=1}^{J} (\bar{P}^k(j)\{r, \bar{m}\} \cdot w^k(j) \cdot D^{k-j}) \\ + (1 - \beta^k) \cdot \sum_{j=1}^{J} (\bar{P}^k(j)\{\bar{r}\} \cdot w^k(j) \cdot \alpha^k(j) \cdot D^{k-j}), \tag{2–65}$$

where β^k is the percentage of I-MBs in the k-th frame; $\bar{P}^k(j)\{r, \bar{m}\}$ is the weighted average of joint PEPs of event $\{r, \bar{m}\}$ for the j-th sub-frame in the k-th frame. $\bar{P}^k(j)\{\bar{r}\}$ is the weighted average of PEP of event $\{\bar{r}\}$ for the j-th sub-frame in the k-th frame.

Lemma 4 is proved in Appendix A.5. With Lemma 4, we have the following general theorem for FTD.

Theorem 2.4. *Under multi-reference prediction, the FTD of the k-th frame is*

$$D^k = D^k(r) + \lambda^k \cdot D^k(m) + D^{k-1} \cdot \bar{P}^k\{r, m\} \\ + \sum_{j=1}^{J} (\bar{P}^k(j)\{r, \bar{m}\} \cdot w^k(j) \cdot D^{k-j}) + (1 - \beta^k) \cdot \sum_{j=1}^{J} (\bar{P}^k(j)\{\bar{r}\} \cdot w^k(j) \cdot \alpha^k(j) \cdot D^{k-j}). \tag{2–66}$$

Proof. (2–66) can be obtained by plugging (3–2), (3–3), (2–65) and (2–51) into (2–16).

□

It is easy to prove that (2–54) in Theorem 2.2 is a special case of (2–66) with $J = 1$ and $w^k(j) = 1$. It is also easy to prove that (2–63) in Theorem 2.3 is a special case of (2–66) with $|\mathcal{V}| = 1$.

Corollary 5. *Under multi-reference prediction and no slice data partitioning,* (2–66) *is simplified to*

$$D^k = D^k(r) + \lambda^k \cdot D^k(m) + D^{k-1} \cdot \bar{P}^k\{r\} + (1 - \beta^k) \cdot \sum_{j=1}^{J} (\bar{P}^k(j)\{\bar{r}\} \cdot w^k(j) \cdot \alpha^k(j) \cdot D^{k-j}). \tag{2–67}$$

CHAPTER 3
PREDICTION OF TRANSMISSION DISTORTION FOR WIRELESS VIDEO COMMUNICATION: ALGORITHM AND APPLICATION

In this chapter, we design the algorithms to estimate the transmission distortion based on the analysis in Chapter 2. We also apply the algorithm in the rate-distortion optimized mode decision problem and achieve a remarkable performance gain than existing solutions.

3.1 A Literature Review on Estimation Algorithms of Transmission Distortion

Transmitting video over wireless with good quality or low end-to-end distortion is particularly challenging since the received video is subject to not only quantization distortion but also transmission distortion (i.e., video distortion caused by packet errors). The capability of predicting transmission distortion can assist in designing video encoding and transmission schemes that achieve maximum video quality or minimum end-to-end video distortion. In Chapter 2, we have theoretically derived formulae for transmission distortion. In this chapter, we leverage the analytical results in Chapter 2 to design algorithms for estimating transmission distortion; we also develop an algorithm for estimating end-to-end distortion, and apply it to prediction mode decision in H.264 encoder.

To estimate frame-level transmission distortion (FTD), several linear model based algorithms [8–11] are proposed. These algorithms use the sum of the newly induced distortion in the current frame and the propagated distortion from previous frames, to estimate transmission distortion. The linear model based algorithms simplify the analysis of transmission distortion at the cost of sacrificing the prediction accuracy by neglecting the correlation between the newly induced error and the propagated error. Liang et al. [16] extend the result in Ref. [8] by addressing the effect of correlation. However, they do not consider the effect of motion vector (MV) error on transmission distortion and their algorithm is not tested with high motion video content. Under this condition, they claim that the LTI models [8, 9] under-estimate transmission distortion

due to positive correlation between two adjacent erroneous frames. In Chapter 2, we identify that the MV concealment error is negatively correlated with the propagated error and this correlation dominates over all other types of correlation especially for high motion video. As long as MV transmission errors exist, the transmission distortion estimated by LTI models becomes over-estimated. In Chapter 2, we also quantify the effects of those correlations on transmission distortion by a system parameter called correlation ratio. On the other hand, none of existing works analyzes the impact of clipping noise on transmission distortion. In Chapter 2, we prove that clipping noise reduces the propagated error and quantify its effect by another system parameter called propagation factor. In this chapter, we design algorithms to estimate correlation ratio and propagation factor, which facilitates the design of a low complexity algorithm called **RMPC-FTD algorithm** for estimating frame-level transmission distortion. Experimental results demonstrate that our RMPC-FTD algorithm is more accurate and more robust than existing algorithms. Another advantage of our RMPC-FTD algorithm is that all parameters in the formula derived in Chapter 2 can be estimated by using the instantaneous video frame statistics and channel conditions, which allows the frame statistics to be time-varying and the error processes to be non-stationary. However, existing algorithms estimate their parameters by using the statistics averaged over multiple frames and assume these statistics do not change over time; their models all assume the error process is stationary. As a result, our RMPC-FTD algorithm is more suitable for real-time video communication.

For pixel-level transmission distortion (PTD), the estimation algorithm is similar to the FTD estimation algorithm since the PTD formula is a special case of the FTD formula as discussed in Chapter 2. However, in some existing video encoders, e.g., H.264 reference code JM14.0[1] , motion estimation and prediction mode decision are

[1] http://iphome.hhi.de/suehring/tml/download/old_jm/jm14.0.zip

separately considered. Therefore, the MV and corresponding residual are known for distortion estimation in mode decision. In such a case, the PTD estimation algorithm can be simplified with known values of the MV and corresponding residual, compared to using their statistics. In this chapter, we design a PTD estimation algorithm, called **RMPC-PTD** for such a case; we also extend RMPC-PTD to estimate pixel-level end-to-end distortion (PEED).

PEED estimation is important for designing optimal encoding and transmission schemes. Some existing PEED estimation algorithms are proposed in Refs. [4, 5]. In Ref. [4], the recursive optimal per-pixel estimate (ROPE) algorithm is proposed to estimate the PEED by recursively calculating the first and second moments of the reconstructed pixel value. However, the ROPE algorithm neglects the significant effect of clipping noise on transmission distortion, resulting in inaccurate estimate. Furthermore, the ROPE algorithm requires intensive computation of correlation coefficients when pixel averaging operations (e.g., in interpolation filter and deblocking filter) are involved [23], which reduces its applicability in H.264 video encoder. Stockhammer et al. [5] propose a distortion estimation algorithm by simulating K independent decoders at the encoder side during the encoding process and averaging the distortions of these K decoders. This algorithm is based on the Law of Large Number (LLN), i.e., the estimated distortion will asymptotically approach the expected distortion as K goes to infinity. For this reason, we call the algorithm in Ref. [5] as LLN algorithm. However, for LLN algorithm, the larger number of decoders simulated, the higher computational complexity and the larger memory required. As a result, LLN algorithm is not suitable for real-time video communication. To enhance estimation accuracy, reduce complexity and improve extensibility, in this chapter, we extend RMPC-PTD algorithm to PEED estimation; the resulting algorithm is called **RMPC-PEED**. Compared to ROPE algorithm, RMPC-PEED algorithm is more accurate since the significant effect of clipping noise on transmission distortion is considered. Another advantage over ROPE algorithm is that RMPC-PEED

algorithm is much easier to be extended to support averaging operations, e.g., interpolation filter. Compared to LLN algorithm, the computational complexity and memory requirement of RMPC-PEED algorithm are much lower and the estimated distortion has smaller variance.

In existing video encoders, prediction mode decision is to choose the best prediction mode in the sense of minimizing the Rate-Distortion (R-D) cost for each Macroblock (MB) or sub-MB. Estimation of the MB level or sub-MB level end-to-end distortion for different prediction modes is needed. In inter-prediction, the reference pixels of the same encoding block may belong to different blocks in the reference frame; therefore, PEED estimation is needed for calculating R-D cost in prediction mode decision. In this chapter, we apply our RMPC-PEED algorithm to prediction mode decision in H.264; the resulting algorithm is called **RMPC-MS**. Experimental results show that, for prediction mode decision in H.264 encoder, our RMPC-MS algorithm achieves an average PSNR gain of 1.44dB over ROPE algorithm for 'foreman' sequence under $PEP = 5\%$; and it achieves an average PSNR gain of 0.89dB over LLN algorithm for 'foreman' sequence under $PEP = 1\%$.

The rest of chapter is organized as follows. Section 3.2 presents our algorithms for estimating FTD under two scenarios: one without acknowledgement feedback and one with acknowledgement feedback. In Section 3.3, we develop algorithms for estimating PTD. In Section 3.4, we extend our PTD estimation algorithm to PEED estimation. In Section 3.5, we apply our PEED estimation algorithm to prediction mode decision in H.264 encoder and compare its complexity with existing algorithms. Section 5.5 shows the experimental results that demonstrate accuracy and robustness of our distortion estimation algorithm and superior R-D performance of our mode decision scheme over existing schemes.

3.2 Algorithms for Estimating FTD

In this section, we develop our algorithms for estimating FTD under two scenarios: one without acknowledgement feedback and one with acknowledgement feedback, which are presented in Sections 3.2.1 and 3.2.2, respectively.

3.2.1 FTD Estimation without Feedback Acknowledgement

Chapter 2 derives a formula for FTD under single-reference prediction, i.e.,

$$D^k = D^k(r) + D^k(m) + D^k(P) + D^k(c), \tag{3–1}$$

where

$$D^k(r) = E[(\varepsilon^k)^2] \cdot \bar{P}^k(r); \tag{3–2}$$

$$D^k(m) = E[(\xi^k)^2] \cdot \bar{P}^k(m); \tag{3–3}$$

$$D^k(P) = \bar{P}^k(r) \cdot D^{k-1} + (1-\beta^k) \cdot (1-\bar{P}^k(r)) \cdot \alpha^k \cdot D^{k-1}; \tag{3–4}$$

$$D^k(c) = (\lambda^k - 1) \cdot D^k(m); \tag{3–5}$$

ε^k is the residual concealment error and $\bar{P}^k(r)$ is the weighted average PEP of all residual packets in the k-th frame; ξ^k is the MV concealment error and $\bar{P}^k(m)$ is the weighted average PEP of all residual packets in the k-th frame; β^k is the percentage of encoded I-MBs in the k-th frame; both the propagation factor α^k and the correlation ratio λ^k depend on video content, channel condition and codec structure, and are therefore called **system parameters**; D^{k-1} is the transmission distortion in the $k-1$ frame, which can be iteratively calculated by (3–1).

Next, Sections 3.2.1.1 through 3.2.1.4 present methods to estimate each of the four distortion terms in (3–1), respectively.

3.2.1.1 Estimation of residual caused distortion

From the analysis in Chapter 2, $E[(\varepsilon^k)^2] = E[(\varepsilon_{\mathbf{u}}^k)^2] = E[(\hat{e}_{\mathbf{u}}^k - \check{e}_{\mathbf{u}}^k)^2]$ for all $\mathbf{u}$ in the k-th frame; $\hat{e}_{\mathbf{u}}^k$ is the transmitted residual for pixel $\mathbf{u}^k$; and $\check{e}_{\mathbf{u}}^k$ is the concealed residual

for pixel $\mathbf{u}^k$ at the decoder. $E[(\varepsilon^k)^2]$ can be estimated from the finite samples of $\varepsilon_{\mathrm{u}}^k$ in the k-th frame, i.e., $\hat{E}[(\varepsilon^k)^2] = \frac{1}{|\mathcal{V}|}\sum_{\mathrm{u}\in\mathcal{V}^k}(\hat{e}_{\mathrm{u}}^k - \widehat{\breve{e}_{\mathrm{u}}^k})^2$, where $\widehat{\breve{e}_{\mathrm{u}}^k}$ is the estimate of $\breve{e}_{\mathrm{u}}^k$.

From the analysis in Chapter 2, $\bar{P}^k(r) = \frac{1}{|\mathcal{V}|}\sum_{i=1}^{N^k(r)}(P_i^k(r)\cdot N_i^k(r))$, where $P_i^k(r)$ is the PEP of the i-th residual packet in the k-th frame; $N_i^k(r)$ is the number of pixels contained in the i-th residual packet of the k-th frame; $N^k(r)$ is the number of residual packets in the k-th frame. $P_i^k(r)$ can be estimated from channel state statistics. Denote the estimated PEP by $\hat{P}_i^k(r)$ for all $i \in \{1, 2, ..., N^k(r)\}$; then $\bar{P}^k(r)$ can be estimated by $\hat{P}^k(r) = \frac{1}{|\mathcal{V}|}\sum_{i=1}^{N^k(r)}(\hat{P}_i^k(r)\cdot N_i^k(r))$. As a result, $D^k(r)$ can be estimated by

$$\hat{D}^k(r) = \hat{E}[(\varepsilon^k)^2]\cdot\hat{P}^k(r) = \frac{1}{(|\mathcal{V}|)^2}\sum_{i=1}^{N^k(r)}(\hat{P}_i^k(r)\cdot N_i^k(r))\sum_{\mathrm{u}\in\mathcal{V}^k}(\hat{e}_{\mathrm{u}}^k - \widehat{\breve{e}_{\mathrm{u}}^k})^2. \tag{3–6}$$

Next, we discuss how to 1) conceal $\hat{e}_{\mathrm{u}}^k$ at the decoder; 2) estimate $\breve{e}_{\mathrm{u}}^k$ at the encoder; and 3) estimate $P_i^k(r)$ at the encoder.

Concealment of $\hat{e}_{\mathrm{u}}^k$ at the decoder: At the decoder, if $\hat{e}_{\mathrm{u}}^k$ is received with error and its neighboring pixels are correctly received, its neighboring pixels could be utilized to conceal $\hat{e}_{\mathrm{u}}^k$. However, this is possible only if the pixel $\mathbf{u}^k$ is at the slice boundary and the pixels at the other side of this slice boundary is correctly received. In H.264, most pixels in a slice do not locate at the slice boundary. Therefore, if one slice is lost, most of pixels in that slice will be concealed without the information from neighboring pixels. If the same method is used to conceal $\breve{e}_{\mathrm{u}}^k$ of all pixels, it is not difficult to prove that the minimum of $E[(\varepsilon^k)^2]$ is achieved when $\breve{e}_{\mathrm{u}}^k = E[\hat{e}_{\mathrm{u}}^k]$.

Note that when $\breve{e}_{\mathrm{u}}^k$ is concealed by $E[\hat{e}_{\mathrm{u}}^k]$ at the decoder, $E[(\varepsilon^k)^2]$ is the variance of $\hat{e}_{\mathrm{u}}^k$, that is, $E[(\varepsilon^k)^2] = \sigma^2_{\hat{e}_{\mathrm{u}}^k}$. In our experiment, we find that the histogram of $\hat{e}^k$ in each frame approximately follows a Laplacian distribution with zero mean. As proved in Ref. [24], the variance of e^k depends on the spatio-temporal correlation of the input video sequence and the accuracy of motion estimation. Since $\hat{e}^k$ is a function of e^k, $E[(\varepsilon^k)^2]$ also depends on the accuracy of motion estimation. So, for a given video sequence, more accurate residual concealment and more accurate motion

estimation produce a smaller $D^k(r)$. This could be used as a criterion for the design of the encoding algorithm at the encoder and residual concealment method at the decoder.

Estimation of $\check{e}_u^k$ at the encoder: If the encoder has knowledge of the concealment method at the decoder as well as the feedback acknowledgement of some packets, $\check{e}_u^k$ can be estimated by the same concealed methods at the decoder. That means the methods to estimate $\check{e}_u^k$ of pixels at the slice boundary are different from other pixels. However, if no feedback acknowledgement of which packets are correctly received, the same method may be used to estimate $\check{e}_u^k$ of all pixels, that is, $\widehat{\check{e}_u^k} = \frac{1}{|\mathcal{V}|}\sum_{u\in\mathcal{V}^k}\hat{e}_u^k$. Note that even if the feedback acknowledgement of some packets are correctly received before the estimation, the estimate obtained by this method at the encoder is still quite accurate since most pixels in a slice do not locate at the slice boundary.

In most cases, for a standard hybrid codec such as H.264, $\frac{1}{|\mathcal{V}|}\sum_{u\in\mathcal{V}^k}\hat{e}_u^k$ approximately equals zero[2] for P-MBs and B-MBs. Therefore, one simple concealment method is to let $\widehat{\check{e}_u^k} = 0$ as in most transmission distortion models. In this chapter, we still use $\widehat{\check{e}_u^k}$ in case $\frac{1}{|\mathcal{V}|}\sum_{u\in\mathcal{V}^k}\hat{e}_u^k \neq 0$ due to the imperfect predictive coding, or in the general case, that is, some feedback acknowledgements may have been received before the estimation. Note that when $\widehat{\check{e}_u^k} = \frac{1}{|\mathcal{V}|}\sum_{u\in\mathcal{V}^k}\hat{e}_u^k$ at the encoder, $\hat{E}[(\varepsilon^k)^2]$ is the sample variance of $\hat{e}_u^k$ and in fact a biased estimator of $\sigma^2_{\hat{e}_u^k}$ [25]. In other words, $\hat{E}[(\varepsilon^k)^2]$ is a sufficient statistic of all individual samples $\hat{e}_u^k$. If the sufficient statistic $\hat{E}[(\varepsilon^k)^2]$ is known, the FTD estimator does not need the values of $\hat{e}_u^k$ of all pixels. Therefore, such an FTD estimator incurs much lower complexity than using the values of $\hat{e}_u^k$ of all pixels.

Estimation of $P_i^k(r)$: In wired communication, application layer PEP is usually estimated by Packet Error Rate (PER), which is the ratio of the number of incorrectly received packets to the number of transmitted packets, that is, $\hat{P}_i^k(r) = PER_i^k(r)$. In a wireless fading channel, instantaneous physical layer PEP is a function of the

[2] This is actually an objective of predictive coding.

instantaneous channel gain $g(t)$ at time t [18], which is denoted by $p(g(t))$. At an encoder, there are two cases: 1) the transmitter has perfect knowledge of $g(t)$, and 2) the transmitter has no knowledge of $g(t)$ but knows the probability density function (pdf) of $g(t)$. For Case 1, the estimated PEP $\hat{P}_i^k(r) = p(g(t))$ since $g(t)$ is known. Note that since the channel gain is time varying, the estimated instantaneous PEP is also time varying.[3] For Case 2, $p(g(t))$ is a random variable since only pdf of $g(t)$ is known. Hence, we should use the expected value of $p(g(t))$ to estimate $P_i^k(r)$, that is, $\hat{P}_i^k(r) = E[p(g(t))]$, where the expectation is taken over the pdf of $g(t)$.

3.2.1.2 Estimation of MV caused distortion

From the analysis in Chapter 2, $E[(\xi^k)^2] = E[(\xi_{\mathbf{u}}^k)^2] = E[(\hat{f}^{k-1}_{\mathbf{u}+\mathbf{mv}_{\mathbf{u}}^k} - \hat{f}^{k-1}_{\mathbf{u}+\check{\mathbf{mv}}_{\mathbf{u}}^k})^2]$ for all $\mathbf{u}$ in the k-th frame; $\mathbf{mv}_{\mathbf{u}}^k$ is the transmitted MV for pixel $\mathbf{u}^k$; and $\check{\mathbf{mv}}_{\mathbf{u}}^k$ is the concealed MV for pixel $\mathbf{u}^k$ at the decoder. $E[(\xi^k)^2]$ can be estimated from the finite samples of $\xi_{\mathbf{u}}^k$ in the k-th frame, i.e., $\hat{E}[(\xi^k)^2] = \frac{1}{|\mathcal{V}|}\sum_{\mathbf{u}\in\mathcal{V}^k}(\hat{f}^{k-1}_{\mathbf{u}+\mathbf{mv}_{\mathbf{u}}^k} - \hat{f}^{k-1}_{\mathbf{u}+\widehat{\check{\mathbf{mv}}_{\mathbf{u}}^k}})^2$, where $\widehat{\check{\mathbf{mv}}_{\mathbf{u}}^k}$ is the estimate of $\check{\mathbf{mv}}_{\mathbf{u}}^k$.

Similar to Section 3.2.1.1, $\bar{P}^k(m) = \frac{1}{|\mathcal{V}|}\sum_{i=1}^{N^k(m)}(P_i^k(m)\cdot N_i^k(m))$, where $P_i^k(m)$ is the PEP of the i-th MV packet in the k-th frame; $N_i^k(m)$ is the number of pixels contained in the i-th MV packet of the k-th frame; $N^k(m)$ is the number of MV packets in the k-th frame. $P_i^k(m)$ can be estimated from channel state statistics. Denote the estimated PEP by $\hat{P}_i^k(m)$ for all $i \in \{1, 2, ..., N^k(m)\}$; then $\bar{P}^k(r)$ can be estimated by $\hat{P}^k(m) = \frac{1}{|\mathcal{V}|}\sum_{i=1}^{N^k(m)}(\hat{P}_i^k(m)\cdot N_i^k(m))$. As a result, $D^k(m)$ can be estimated by

$$\hat{D}^k(m) = \hat{E}[(\xi^k)^2]\cdot\hat{P}^k(m) = \frac{1}{(|\mathcal{V}|)^2}\sum_{i=1}^{N^k(m)}(\hat{P}_i^k(m)\cdot N_i^k(m))\sum_{\mathbf{u}\in\mathcal{V}^k}(\hat{f}^{k-1}_{\mathbf{u}+\mathbf{mv}_{\mathbf{u}}^k} - \hat{f}^{k-1}_{\mathbf{u}+\widehat{\check{\mathbf{mv}}_{\mathbf{u}}^k}})^2. \quad (3\text{–}7)$$

Next, we discuss how to 1) conceal $\mathbf{mv}_{\mathbf{u}}^k$ at the decoder; 2) estimate $\check{\mathbf{mv}}_{\mathbf{u}}^k$ at the encoder; and 3) estimate $P_i^k(m)$ at the encoder.

[3] This implies that the pixel error process is non-stationary over both time and space.

Concealment of $\mathbf{mv}_{\mathrm{u}}^{k}$ at the decoder: Different from residual, MV are highly correlated in both temporal and spatial domains. Hence, the decoder may conceal the MV by temporally neighboring block if its spatially neighboring blocks are not available. Depending on whether the neighboring blocks are correctly received or not, there may be several options of MV error concealment methods for each block, or each pixel to make it more general. If the neighboring blocks are correctly received, $\check{\mathbf{mv}}_{\mathrm{u}}^{k}$ can be concealed by the median or average of those neighboring blocks. Interested readers may refer to Ref. [21], [26], [27] for discussions on different MV concealment methods. In our experiment, we also observe that the histogram of ξ^k in one frame approximately follows a Laplacian distribution with zero mean. For different concealment methods, the variance of ξ^k will be different. The more accurate concealed motion estimation, the smaller $D^k(m)$.

Estimation of $\check{\mathbf{mv}}_{\mathrm{u}}^{k}$ at the encoder: If the encoder knows the concealment methods of current block and the PEP of neighboring blocks, we can estimate the MV caused distortion by assigning different concealment methods with different probabilities at the encoder as in Ref. [4]. However, if the encoder does not know what concealment methods are used by the decoder or no neighboring blocks can be utilized for error concealment (e.g., both temporal and spatial neighboring blocks are in error), a simple estimation algorithm [9], [10] is to let $\widehat{\check{\mathbf{mv}}_{\mathrm{u}}^{k}} = 0$, that is, using the pixel value from the same position of the previous frame. In this chapter, we still use $\widehat{\check{\mathbf{mv}}_{\mathrm{u}}^{k}}$ to denote the estimate of concealed motion vector for the general case.

Estimation of $P_i^k(m)$: The estimation of $P_i^k(m)$ is similar to the estimation of $P_i^k(r)$. Note that in H.264 specification, there is no slice data partitioning for an instantaneous decoding refresh (IDR) frame [22], so $P_i^k(r) = P_i^k(m)$ for all pixels in an IDR-frame. This is also true for I-MB, and P-MB without slice data partitioning. For P-MB with slice data partitioning in H.264, the error state of residual and the error state of MV of the same pixel are partially correlated. To be more specific, if the MV packet is lost, the

corresponding residual packet cannot be decoded even if it is correctly received, since there is no slice header in the residual packet. As a result, $P_i^k(r_{H.264}) = P_i^k(r) + (1 - P_i^k(r))P_i^k(m)$.

3.2.1.3 Estimation of propagation and clipping caused distortion

To estimate $D^k(P)$, we only need to estimate α^k since $\bar{P}^k(r)$ has been estimated in Section 3.2.1.1. In Chapter 2, we theoretically derive the propagation factor α_{u}^k of pixel $\mathbf{u}^k$ for propagated error with a zero-mean Laplacian distribution, i.e.,

$$\alpha = 1 - \frac{1}{2}e^{-\frac{y-\gamma_L}{b}}(\frac{y-\gamma_L}{b}+1) - \frac{1}{2}e^{-\frac{\gamma_H-y}{b}}(\frac{(\gamma_H-y)}{b}+1), \tag{3–8}$$

where γ_L and γ_H are user-specified low threshold and high threshold, respectively; y is the reconstructed pixel value; $b = \frac{\sqrt{2}}{2}\sigma$; and σ is the standard deviation of the propagated error.

Here, we provide three methods to estimate the propagation factor α^k as below.

Estimation of α^k by α_{u}^k: As defined in Chapter 2, $\alpha^k = \frac{\sum_{\mathrm{u}\in\mathcal{V}^k}\alpha_{\mathrm{u}}^k \cdot D_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-1}}{\sum_{\mathrm{u}\in\mathcal{V}^k} D_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-1}}$. Therefore, we may first estimate α_{u}^k by (3–8) and then estimate α^k by its definition. However, this method requires to compute exponentiations and divisions in (3–8) for each pixel, and needs large memory to store $\hat{D}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-1}$ for all pixels in all reference frames.

Estimate the average of a function by the function of an average: If we estimate α^k directly by the frame statistics instead of pixel values, both the computational complexity and memory requirement will be decreased by a factor of N_{V^k}. If only $\hat{D}^{k-1}$ instead of $\hat{D}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-1}$ is stored in memory, we may simplify estimating α^k by $\hat{\alpha}^k = \frac{\sum_{\mathrm{u}\in\mathcal{V}^k}\hat{\alpha}_{\mathrm{u}}^k \cdot \hat{D}^{k-1}}{\sum_{\mathrm{u}\in\mathcal{V}^k}\hat{D}^{k-1}} = \frac{1}{|\mathcal{V}|}\sum_{\mathrm{u}\in\mathcal{V}^k}\hat{\alpha}_{\mathrm{u}}^k$. This is accurate if all packets in the same frame experience the same channel condition. We see from (3–8) that α_{u}^k is a function of the reconstructed pixel value $\hat{f}_{\mathrm{u}}^k$ and the variance of propagated error $\sigma^2_{\hat{f}^{k-1}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}}$, which is equal to D^{k-1} in this case. Denote $\alpha_{\mathrm{u}}^k = g(\hat{f}_{\mathrm{u}}^k, D^{k-1})$; we have $\alpha^k = \frac{1}{|\mathcal{V}|}\sum_{\mathrm{u}\in\mathcal{V}^k} g(\hat{f}_{\mathrm{u}}^k, D^{k-1})$. One simple and intuitive method is to use the function of an average to estimate the average of a function, that is, $\hat{\alpha}^k = g(\frac{1}{|\mathcal{V}|}\sum_{\mathrm{u}\in\mathcal{V}^k}\hat{f}_{\mathrm{u}}^k, \hat{D}^{k-1})$.

Improve estimation accuracy by using the property of (3–8): Although the above method dramatically reduces the estimation complexity and memory requirement, that simple approximation is only accurate if α_{u}^k is a linear function of $\hat{f}_{\mathrm{u}}^k$. In other words, such approximation causes underestimation for the convex function or overestimation for the concave function [28]. Although (3–8) is neither a convex function nor a concave function, it is interesting to see that 1) α_{u}^k is symmetric about $\hat{f}_{\mathrm{u}}^k = \frac{\gamma_H+\gamma_L}{2}$; 2) α_{u}^k is a monotonically increasing function of $\hat{f}_{\mathrm{u}}^k$ when $\gamma_L < \hat{f}_{\mathrm{u}}^k < \frac{\gamma_H+\gamma_L}{2}$, and α_{u}^k is a monotonically decreasing function of $\hat{f}_{\mathrm{u}}^k$ when $\frac{\gamma_H+\gamma_L}{2} < \hat{f}_{\mathrm{u}}^k < \gamma_H$; 3) both half sides are much more linear than the whole function. So, we propose to use $\frac{1}{|\mathcal{V}|}\sum_{\mathrm{u}\in\mathcal{V}^k}|\hat{f}_{\mathrm{u}}^k - \frac{\gamma_H+\gamma_L}{2}| + \frac{\gamma_H+\gamma_L}{2}$ instead of $\frac{1}{|\mathcal{V}|}\sum_{\mathrm{u}\in\mathcal{V}^k}\hat{f}_{\mathrm{u}}^k$ to estimate α^k. Since the symmetry property is exploited, such algorithm gives much more accurate estimate $\hat{\alpha}^k$.

From the analysis in Chapter 2, we have $D^k(p) = \alpha^k \cdot D^{k-1}$; so we can estimate $D^k(p)$ by $\hat{D}^k(p) = \hat{D}^{k-1} \cdot \hat{\alpha}^k$. To compensate the accuracy loss of using frame statistics, we may use the following algorithm to estimate $D^k(p)$ without the exponentiation and division for each pixel:

$$\hat{D}^k(p) = (\hat{D}^{k-1} - \hat{D}^{k-1}(r) - \hat{D}^{k-1}(m)) \cdot \hat{\alpha}^k + \overline{\Phi^2(\varepsilon^{k-1},\ \hat{f}^k)} + \overline{\Phi^2(\xi^{k-1},\ \hat{f}^k)}, \tag{3–9}$$

where $\hat{D}^{k-1}(r)$ can be estimated by (3–6); $\hat{D}^{k-1}(m)$ can be estimated by (3–7); $\hat{\alpha}^k$ can be estimated by (3–8); $\overline{\Phi^2(\varepsilon^{k-1},\ \hat{f}^k)} = \frac{1}{|\mathcal{V}|}\sum_{\mathrm{u}\in\mathcal{V}^k}\Phi^2(\varepsilon_{\mathrm{u}}^{k-1},\ \hat{f}_{\mathrm{u}}^k)$ and $\overline{\Phi^2(\xi^{k-1},\ \hat{f}^k)} = \frac{1}{|\mathcal{V}|}\sum_{\mathrm{u}\in\mathcal{V}^k}\Phi^2(\xi_{\mathrm{u}}^{k-1},\ \hat{f}_{\mathrm{u}}^k)$, while both of them can be easily calculated by

$$\Phi(x,\ y) \triangleq y - \Gamma(y-x) = \begin{cases} y-\gamma_L, & y-x<\gamma_L \\ x, & \gamma_L \le y-x \le \gamma_H \\ y-\gamma_H, & y-x>\gamma_H. \end{cases} \tag{3–10}$$

Our experimental results in Section 5.5 show that the proposed algorithm provides accurate estimate. Finally, it is straightforward to estimate $D^k(P)$ by

$$\hat{D}^k(P) = \hat{P}^k(r) \cdot \hat{D}^{k-1} + (1 - \beta^k) \cdot (1 - \hat{P}^k(r)) \cdot \hat{D}^k(p). \tag{3–11}$$

3.2.1.4 Estimation of correlation-caused distortion

To estimate $D^k(c)$, the only parameter needs to be estimated is λ^k since $D^k(m)$ has been estimated in Section 3.2.1.2. As defined in Chapter 2, $\lambda^k = \frac{1}{|\mathcal{V}|}\sum_{\mathrm{u}\in\mathcal{V}^k} \lambda^k_{\mathrm{u}}$, where $\lambda^k_{\mathrm{u}} = \frac{E[\xi^k_{\mathrm{u}} \cdot \widetilde{f}^{k-1}_{\mathrm{u}+\check{\mathrm{mv}}^k_{\mathrm{u}}}]}{E[\xi^k_{\mathrm{u}} \cdot \hat{f}^{k-1}_{\mathrm{u}+\check{\mathrm{mv}}^k_{\mathrm{u}}}]}$. λ^k_{u} depends on the motion activity of the video content according to Chapter 2.

In our experiment, we find that λ^k is small when the average MV length over the set in the k-th frame is larger than half of the block length, and $\lambda^k \approx 1$ when the average MV length in the k-th frame is smaller than half of the block length, or when the propagated error from the reference frames is small. An intuitive explanation for this phenomenon is as below: 1) if the average MV length is large and the MV packets are received with error, most concealed reference pixels will be in some block different from the block where the corresponding true reference pixels locate; 2) if the average MV length is small, most concealed reference pixels and the corresponding true reference pixels will still be in the same block even if the MV packet is received with error; 3) since the correlation between two pixels inside the same block is much higher than the correlation between two pixels located in different blocks, hence λ^k is small when the average MV length is large and vice versa; 4) if there is no propagated error from the reference frames, according to the definition, it is easy to prove that $\lambda^k = 1$.

Therefore, we propose a low complexity algorithm to estimate λ^k by video frame statistics as below

$$\hat{\lambda}^k = \begin{cases} (1 - \bar{P}^{k-1}(m))(1 - \bar{P}^{k-1}(r)), & \overline{|\mathbf{mv}^k|} > \frac{\text{block\_size}}{2} \\ 1, & \text{otherwise}, \end{cases} \tag{3–12}$$

where $\bar{P}^{k-1}(r)$ is defined in (3–4); $\bar{P}^{k-1}(m)$ is defined in (3–5); $\overline{|\mathbf{mv}^k|} = \frac{1}{|\mathcal{V}|}\sum_{\mathrm{u}\in\mathcal{V}^k}|\mathbf{mv}_{\mathrm{u}}^k|$, and $|\mathbf{mv}_{\mathrm{u}}^k|$ is the length of $\mathbf{mv}_{\mathrm{u}}^k$. As a result,

$$\hat{D}^k(c) = (\hat{\lambda}^k - 1)\cdot\hat{D}^k(m). \tag{3–13}$$

3.2.1.5 Summary

Without feedback acknowledgement, the transmission distortion of the k-th frame can be estimated by

$$\hat{D}^k = \hat{D}^k(r) + \hat{\lambda}^k\cdot\hat{D}^k(m) + \hat{P}^k(r)\cdot\hat{D}^{k-1} + (1-\beta^k)\cdot(1-\hat{P}^k(r))\cdot\hat{D}^k(p), \tag{3–14}$$

where $\hat{D}^k(r)$ can be estimated by (3–6); $\hat{D}^k(m)$ can be estimated by (3–7); $\hat{D}^k(p)$ can be estimated by (3–9); $\hat{\lambda}^k$ can be estimated by (3–12); $\hat{P}^k(r)$ can be estimated by the estimated PEP of all residual packets in the k-th frame as discussed in Section 3.2.1.1. We call the resulting algorithm in (3–14) as **RMPC-FTD algorithm**.

3.2.2 FTD Estimation with Feedback Acknowledgement

In some wireless video communication systems, the receiver may send the transmitter a notification about whether packets are correctly received. This feedback acknowledgement mechanism can be utilized to improve FTD estimation accuracy as shown in Algorithm 1.

Algorithm 1. *FTD estimation at the transmitter under feedback acknowledgement.*

1) ***Input****: $\hat{P}_i^k(r)$ and $\hat{P}_i^k(m)$ for all $i \in \{1, 2, ..., N^k\}$.*

2) Initialization and update.

If $k = 1$, do initialization.

If $k > 1$, update with feedback information.

If there are acknowledgements for packets in the $(k-1)$-th frame,

For $j = 1 : N^{k-1}$

if ACK for the j-th residual packet is received, update $\hat{P}_j^{k-1}(r) = 0$.

if NACK for the j-th residual packet is received, update $\hat{P}_j^{k-1}(r) = 1$.

if ACK for the j-th MV packet is received, update $\hat{P}_j^{k-1}(m) = 0$.

if NACK for the j-th MV packet is received, update $\hat{P}_j^{k-1}(m) = 1$.

End

Update $\hat{D}^{k-1}$.

Else (neither ACK nor NACK is received), go to 3).

3) Estimate D^k via

$$\hat{D}^k = \hat{D}^k(r) + \hat{\lambda}^k \cdot \hat{D}^k(m) + \hat{P}^k(r) \cdot \hat{D}^{k-1} + (1 - \beta^k) \cdot (1 - \hat{P}^k(r)) \cdot \hat{D}^k(p),$$

which is (3–14).

4) ***Output****: $\hat{D}^k$.*

Algorithm 1 has a low computational complexity since $\hat{D}^{k-1}$ is updated based on whether packets in the $(k-1)$-th frame are correctly received or not. In a more general case that the encoder can tolerate a feedback delay of d frames, we could update $\hat{D}^{k-1}$ based on the feedback acknowledgements for the $(k-d)$-th frame through the $(k-1)$-th frame. However, this requires extra memory for the encoder to store all the system parameters from the $(k-d)$-th frame to the $(k-1)$-th frame in order to update $\hat{D}^{k-1}$.

3.3 Pixel-level Transmission Distortion Estimation Algorithm

The PTD estimation algorithm is similar to the FTD estimation algorithm presented in Section 3.2. However, the values of some variables in the PTD formula derived in Chapter 2 may be known at the encoder. Taking $\mathbf{u}^k$ as an example, before the prediction mode is selected, the best motion vector $\mathbf{mv}_{\mathrm{u}}^k$ of each prediction mode is known after motion estimation is done; hence the residual $\hat{e}_{\mathrm{u}}^k$ and reconstructed pixel value $\hat{f}_{\mathrm{u}}^k$ of each mode are also known. In such a case, these known values could be used to replace the statistics of the corresponding random variables to simplify the PTD estimation. In this section, we discuss how to use the known values to improve the estimation accuracy and reduce the algorithm complexity.

3.3.1 Estimation of PTD

In this section, we consider the case with no data partitioning; hence, $P_{\mathrm{u}}^k = P_{\mathrm{u}}^k(r) = P_{\mathrm{u}}^k(m)$. For the case with slice data partitioning, the derivation process is similar to that in this section.

From Chapter 2, we know that PTD can be calculated by

$$D_{\mathrm{u}}^k = D_{\mathrm{u}}^k(r) + D_{\mathrm{u}}^k(m) + D_{\mathrm{u}}^k(P) + D_{\mathrm{u}}^k(c), \tag{3–15}$$

where

$$D_{\mathrm{u}}^k(r) = E[(\varepsilon_{\mathrm{u}}^k)^2] \cdot P_{\mathrm{u}}^k(r); \tag{3–16}$$

$$D_{\mathrm{u}}^k(m) = E[(\xi_{\mathrm{u}}^k)^2] \cdot P_{\mathrm{u}}^k(m); \tag{3–17}$$

$$D_{\mathrm{u}}^k(P) = P_{\mathrm{u}}^k \cdot D_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1} + (1 - P_{\mathrm{u}}^k) \cdot D_{\mathrm{u}}^k(p); \tag{3–18}$$

$$D_{\mathrm{u}}^k(c) = 2P_{\mathrm{u}}^k \cdot (2E[\varepsilon_{\mathrm{u}}^k \cdot \xi_{\mathrm{u}}^k] + 2E[\varepsilon_{\mathrm{u}}^k \cdot \widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1}] + 2E[\xi_{\mathrm{u}}^k \cdot \widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1}]); \tag{3–19}$$

where $D_{\mathrm{u}}^k(p) \triangleq E[(\widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-j} + \widetilde{\Delta}_{\mathrm{u}}^k\{\bar{r}, \bar{m}\})^2]$ for $j \in \{1, ..., J\}$; J is the number of previous encoded frames used for inter motion search; $\widetilde{\Delta}_{\mathrm{u}}^k\{\bar{r}, \bar{m}\}$ denotes the clipping noise under the error event that the packet is correctly received.

If the values for $\mathbf{mv}_{\mathrm{u}}^k$, $\hat{e}_{\mathrm{u}}^k$ and $\hat{f}_{\mathrm{u}}^k$ are known, given the error concealment at the encoder, the values for $\varepsilon_{\mathrm{u}}^k = \hat{e}_{\mathrm{u}}^k - \check{e}_{\mathrm{u}}^k$ and $\xi_{\mathrm{u}}^k = \hat{f}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-1} - \hat{f}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1}$ are also known. Then, $D_{\mathrm{u}}^k(r) = (\varepsilon_{\mathrm{u}}^k)^2 \cdot P_{\mathrm{u}}^k$, $D_{\mathrm{u}}^k(m) = (\xi_{\mathrm{u}}^k)^2 \cdot P_{\mathrm{u}}^k$, and $D_{\mathrm{u}}^k(c) = P_{\mathrm{u}}^k \cdot (2\varepsilon_{\mathrm{u}}^k \cdot \xi_{\mathrm{u}}^k + 2\varepsilon_{\mathrm{u}}^k \cdot E[\widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1}] + 2\xi_{\mathrm{u}}^k \cdot E[\widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1}])$. Hence, the formula for PTD can be simplified to

$$D_{\mathrm{u}}^k = E[(\widetilde{\zeta}_{\mathrm{u}}^k)^2] = P_{\mathrm{u}}^k \cdot ((\varepsilon_{\mathrm{u}}^k + \xi_{\mathrm{u}}^k)^2 + 2(\varepsilon_{\mathrm{u}}^k + \xi_{\mathrm{u}}^k) \cdot E[\widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1}] + D_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1}) + (1 - P_{\mathrm{u}}^k) \cdot D_{\mathrm{u}}^k(p). \tag{3–20}$$

Denote $\hat{D}(\cdot)$ the estimate of $D(\cdot)$, and denote $\hat{E}(\cdot)$ as the estimate of $E(\cdot)$. Therefore, D_{u}^k can be estimated by $\hat{D}_{\mathrm{u}}^k = \hat{P}_{\mathrm{u}}^k \cdot ((\varepsilon_{\mathrm{u}}^k + \xi_{\mathrm{u}}^k)^2 + 2(\varepsilon_{\mathrm{u}}^k + \xi_{\mathrm{u}}^k) \cdot \hat{E}[\widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1}] + \hat{D}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1}) + (1 - \hat{P}_{\mathrm{u}}^k) \cdot \hat{D}_{\mathrm{u}}^k(p)$, where $\hat{P}_{\mathrm{u}}^k$ can be obtained by the PEP estimation algorithm in Section 3.2. $\hat{D}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1}$ is the estimate in the $(k-1)$-th frame and is stored for calculating

D_{u}^k. Therefore, the only unknowns are $\hat{E}[\widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1}]$ and $\hat{D}_{\mathrm{u}}^k(p)$, which can be calculated by the methods in Sections 3.3.2 and 3.3.3.

3.3.2 Calculation of $\hat{E}[\widetilde{\zeta}_{\mathrm{u}}^k]$

Since $\hat{E}[\widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1}]$ from the $(k-1)$-th frame is required for calculating $\hat{D}_{\mathrm{u}}^k$, we should estimate the first moment of $\widetilde{\zeta}_{\mathrm{u}}^k$ and store it for the subsequent frame. From Chapter 2, we know $\widetilde{\zeta}_{\mathrm{u}}^k = \widetilde{\varepsilon}_{\mathrm{u}}^k + \widetilde{\xi}_{\mathrm{u}}^k + \widetilde{\zeta}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^k}^{k-j'} + \widetilde{\Delta}_{\mathrm{u}}^k$. For P-MBs, when MV packet is correctly received, $\widetilde{\varepsilon}_{\mathrm{u}}^k = \widetilde{\xi}_{\mathrm{u}}^k = 0$ and $\widetilde{\zeta}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^k}^{k-j'} = \widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-j}$; when MV packet is received with error, $\widetilde{\zeta}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^k}^{k-j'} = \widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1}$, and since residual and MV are in the same packet, $\widetilde{\Delta}_{\mathrm{u}}^k\{r, m\} = \widetilde{\Delta}_{\mathrm{u}}^k\{r\} = 0$ as proved in Chapter 2. Therefore, the first moment of ζ_{u}^k can be recursively calculated by

$$E[\widetilde{\zeta}_{\mathrm{u}}^k] = P_{\mathrm{u}}^k \cdot (\varepsilon_{\mathrm{u}}^k + \xi_{\mathrm{u}}^k + E[\widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1}]) + (1 - P_{\mathrm{u}}^k) \cdot E[\widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-j} + \widetilde{\Delta}_{\mathrm{u}}^k\{\bar{r}, \bar{m}\}]. \tag{3–21}$$

Consequently, $E[\widetilde{\zeta}_{\mathrm{u}}^k]$ can be estimated by $\hat{E}[\widetilde{\zeta}_{\mathrm{u}}^k] = \hat{P}_{\mathrm{u}}^k \cdot (\varepsilon_{\mathrm{u}}^k + \xi_{\mathrm{u}}^k + \hat{E}[\widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1}]) + (1 - \hat{P}_{\mathrm{u}}^k) \cdot \hat{E}[\widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-j} + \widetilde{\Delta}_{\mathrm{u}}^k\{\bar{r}, \bar{m}\}]$.

For I-MBs, when the packet is correctly received, $\widetilde{\zeta}_{\mathrm{u}}^k = 0$; when MV packet is received with error, the result is the same as for P-MBs. Therefore, the first moment of ζ_{u}^k can be recursively calculated by

$$E[\widetilde{\zeta}_{\mathrm{u}}^k] = P_{\mathrm{u}}^k \cdot (\varepsilon_{\mathrm{u}}^k + \xi_{\mathrm{u}}^k + E[\widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1}]), \tag{3–22}$$

and $E[\widetilde{\zeta}_{\mathrm{u}}^k]$ can be estimated by $\hat{E}[\widetilde{\zeta}_{\mathrm{u}}^k] = \hat{P}_{\mathrm{u}}^k \cdot (\varepsilon_{\mathrm{u}}^k + \xi_{\mathrm{u}}^k + \hat{E}[\widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1}])$.

3.3.3 Calculation of $\hat{E}[\widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-j} + \widetilde{\Delta}_{\mathrm{u}}^k\{\bar{r}, \bar{m}\}]$ and $\hat{D}_{\mathrm{u}}^k(p)$

From Chapter 2, we know that for I-MBs, $D_{\mathrm{u}}^k(p) = 0$; for P-MBs, $D_{\mathrm{u}}^k(p) = \alpha_{\mathrm{u}}^k \cdot D_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-j}$ and it can be estimated by $\hat{D}_{\mathrm{u}}^k(p) = \hat{\alpha}_{\mathrm{u}}^k \cdot \hat{D}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-j}$, where $\hat{\alpha}_{\mathrm{u}}^k$ is estimated by (3–8) with $y = \hat{f}_{\mathrm{u}}^k$ and $\sigma^2 = \hat{D}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-j}$.

However, such complexity is too high to be used in prediction mode decision since every pixel requires such a computation for each mode. To address this, we leverage the property proved in Proposition 3 to design a low-complexity and high-accuracy algorithm to recursively calculate $\hat{E}[\widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-j} + \widetilde{\Delta}_{\mathrm{u}}^k\{\bar{r}, \bar{m}\}]$ and $\hat{D}_{\mathrm{u}}^k(p)$ for P-MBs.

Proposition 3. *Assume* $\gamma_H = 255$ *and* $\gamma_L = 0$*. The propagated error* $\widetilde{\zeta}^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}}$ *and the clipping noise* $\widetilde{\Delta}^k_{\mathrm{u}}\{\bar{r}, \bar{m}\}$ *satisfy*

$$\widetilde{\zeta}^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}} + \widetilde{\Delta}^k_{\mathrm{u}}\{\bar{r}, \bar{m}\} = \begin{cases} \hat{f}^k_{\mathrm{u}} - 255, & \widetilde{\zeta}^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}} < \hat{f}^k_{\mathrm{u}} - 255 \\ \hat{f}^k_{\mathrm{u}}, & \widetilde{\zeta}^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}} > \hat{f}^k_{\mathrm{u}} \\ \widetilde{\zeta}^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}}, & \textit{otherwise}. \end{cases} \tag{3–23}$$

Proposition 3 is proved in Appendix B.1. Using Proposition 3, $\hat{E}[\widetilde{\zeta}^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}} + \widetilde{\Delta}^k_{\mathrm{u}}\{\bar{r}, \bar{m}\}]$ in (4–1) and $\hat{D}^k_{\mathrm{u}}(p)$ in (4–2) can be estimated under the following three cases.

Case 1: If $\hat{E}[\widetilde{\zeta}^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}}] < \hat{f}^k_{\mathrm{u}} - 255$, we have $\hat{E}[\widetilde{\zeta}^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}} + \widetilde{\Delta}^k_{\mathrm{u}}\{\bar{r}, \bar{m}\}] = \hat{f}^k_{\mathrm{u}} - 255$, and $\hat{D}^k_{\mathrm{u}}(p) = \hat{E}[(\widetilde{\zeta}^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}} + \widetilde{\Delta}^k_{\mathrm{u}}\{\bar{r}, \bar{m}\})^2] = (\hat{f}^k_{\mathrm{u}} - 255)^2$.

Case 2: If $\hat{E}[\widetilde{\zeta}^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}}] > \hat{f}^k_{\mathrm{u}}$, we have $\hat{E}[\widetilde{\zeta}^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}} + \widetilde{\Delta}^k_{\mathrm{u}}\{\bar{r}, \bar{m}\}] = \hat{f}^k_{\mathrm{u}}$, and $\hat{D}^k_{\mathrm{u}}(p) = (\hat{f}^k_{\mathrm{u}})^2$.

Case 3: If $\hat{f}^k_{\mathrm{u}} - 255 \leq \hat{E}[\widetilde{\zeta}^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}}] \leq \hat{f}^k_{\mathrm{u}}$, we have $\hat{E}[\widetilde{\zeta}^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}} + \widetilde{\Delta}^k_{\mathrm{u}}\{\bar{r}, \bar{m}\}] = \hat{E}[\widetilde{\zeta}^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}}]$, and $\hat{D}^k_{\mathrm{u}}(p) = \hat{E}[(\widetilde{\zeta}^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}})^2]$.

3.3.4 Summary

PTD can be recursively estimated by (4–2) and (4–1) or (3–22); and $\hat{E}[\widetilde{\zeta}^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}} + \widetilde{\Delta}^k_{\mathrm{u}}\{\bar{r}, \bar{m}\}]$ and $\hat{D}^k_{\mathrm{u}}(p)$ can be calculated by the methods in Section 3.3.3. The resulting algorithm is called **RMPC-PTD** algorithm.

3.4 Pixel-level End-to-end Distortion Estimation Algorithm

The pixel-level end-to-end distortion (PEED) for each pixel $\mathbf{u}$ in the k-th frame is defined by $D^k_{\mathrm{u},ETE} \triangleq E[(f^k_{\mathrm{u}} - \widetilde{f}^k_{\mathrm{u}})^2]$, where f^k_{u} is the input pixel value at the encoder and $\widetilde{f}^k_{\mathrm{u}}$ is the reconstructed pixel value at the decoder. Then we have

$$\begin{aligned} D^k_{\mathrm{u},ETE} &= E[(f^k_{\mathrm{u}} - \widetilde{f}^k_{\mathrm{u}})^2] \\ &= E[(f^k_{\mathrm{u}} - \hat{f}^k_{\mathrm{u}} + \hat{f}^k_{\mathrm{u}} - \widetilde{f}^k_{\mathrm{u}})^2] \\ &= E[(f^k_{\mathrm{u}} - \hat{f}^k_{\mathrm{u}} + \widetilde{\zeta}^k_{\mathrm{u}})^2] \\ &= (f^k_{\mathrm{u}} - \hat{f}^k_{\mathrm{u}})^2 + E[(\widetilde{\zeta}^k_{\mathrm{u}})^2] + 2(f^k_{\mathrm{u}} - \hat{f}^k_{\mathrm{u}}) \cdot E[\widetilde{\zeta}^k_{\mathrm{u}}]. \end{aligned} \tag{3–24}$$

We call $f_u^k - \hat{f}_u^k$ quantization error and $\widetilde{\zeta}_u^k$ transmission error. While $f_u^k - \hat{f}_u^k$ depends only on the quantization parameter (QP), $\widetilde{\zeta}_u^k$ mainly depends on the PEP and the error concealment scheme. If the value of $\hat{f}_u^k$ is known, then the only unknowns in (3–24) are $E[(\widetilde{\zeta}_u^k)^2]$ and $E[\widetilde{\zeta}_u^k]$, which can be estimated by the methods in Section 3.3. We call the algorithm in (3–24) as **RPMC-PEED** algorithm.

Compared to ROPE algorithm [4], which estimates the first moment and second moment of the reconstructed pixel value $\widetilde{f}_u^k$, we have the following observations. First, RPMC-PEED algorithm estimates the first moment and the second moment of reconstructed error $\widetilde{\zeta}_u^k$; therefore, RPMC-PEED algorithm is much easier to be enhanced to support the averaging operations in H.264, such as interpolation filter. Second, estimating the first moment and the second moment of $\widetilde{\zeta}_u^k$ in RMPC-PEED produces lower distortion estimation error than estimating both moments of $\widetilde{f}_u^k$ in ROPE. Third, our experimental results show that ROPE may produce a negative value as the estimate for distortion, which violates the requirement that (true) distortion must be non-negative; our experimental results also show that the negative distortion estimate is caused by not considering clipping, which results in inaccurate distortion estimation by ROPE.

Note that in Chapter 2, we assume the clipping noise at the encoder is zero, that is, $\hat{\Delta}_u^k = 0$. If we use $\hat{f}_{u+mv_u^k}^{k-j} + \hat{e}_u^k$ to replace $\hat{f}_u^k$ in (3–24), we may calculate the quantization error by $f_u^k - (\hat{f}_{u+mv_u^k}^{k-j} + \hat{e}_u^k)$ and calculate the transmission error by

$$\begin{aligned}\widetilde{\zeta}_u^k &= (\hat{f}_{u+mv_u^k}^{k-j} + \hat{e}_u^k) - \widetilde{f}_u^k \\ &= (\hat{f}_{u+mv_u^k}^{k-j} + \hat{e}_u^k) - (\widetilde{f}_{u+\widetilde{mv}_u^k}^{k-1} + \widetilde{e}_u^k - \widetilde{\Delta}_u^k) \\ &= \widetilde{\varepsilon}_u^k + \widetilde{\xi}_u^k + \widetilde{\zeta}_{u+\widetilde{mv}_u^k}^{k-1} + \widetilde{\Delta}_u^k,\end{aligned} \tag{3–25}$$

which is exactly the formula for transmission error decomposition in Chapter 2. Therefore, $\hat{\Delta}_u^k$ does not affect the end-to-end distortion $D_{u,ETE}^k$ if we use $\hat{f}_{u+mv_u^k}^{k-j} + \hat{e}_u^k$ to replace $\hat{f}_u^k$ in calculating both the quantization error and the transmission error.

3.5 Applying RMPC-PEED Algorithm to H.264 Prediction Mode Decision

3.5.1 Rate-distortion Optimized Prediction Mode Decision

In H.264 specification, there are two types of prediction modes, i.e., inter prediction and intra prediction[4] . In inter prediction, there are 7 modes, i.e., modes for 16x16, 16x8, 8x16, 8x8, 8x4, 4x8, and 4x4 luma blocks. In intra prediction, there are 9 modes for 4x4 luma blocks and 4 modes for 16x16 luma blocks. Hence, there are a total of $7 + 9 + 4 = 20$ modes to be selected in mode decision. For each MB, our proposed Error-Resilient Rate Distortion Optimized (ERRDO) mode decision consists of two steps. First, R-D cost is calculated by

$$J(\omega_m) = D^k_{ETE}(\omega_m) + \lambda \cdot R(\omega_m), \tag{3–26}$$

where $D^k_{ETE} = \sum_{\mathrm{u}\in\mathcal{V}^k_i} D^k_{\mathrm{u},ETE}$; $\mathcal{V}^k_i$ is the set of pixels in the i-th MB (or sub-MB) of the k-th frame; ω_m is the prediction mode, and ω_m ($\omega_m \in \{1, 2, \cdots, 20\}$); $R(\omega_m)$ is the encoded bit rate for mode ω_m; λ is the preset Lagrange multiplier. Then, the optimal prediction mode that minimizes the rate-distortion (R-D) cost is found by

$$\hat{\omega}_m = \arg\min_{\omega_m}\{J(\omega_m)\}. \tag{3–27}$$

If $D^k_{ETE}(\omega_m)$ in (3–26) is replaced by source coding distortion or quantization distortion, we call it Source-Coding Rate Distortion Optimized (SCRDO) mode decision.

Using (3–26) and (3–27), we design Algorithm 2 for ERRDO mode decision in H.264; Algorithm 2 is called **RMPC-MS** algorithm.

Algorithm 2. *ERRDO Mode decision for an MB in the k-th frame ($k \geq 1$).*

*1) **Input**: QP, PEP.*

2) Initialization of $\hat{E}[\widetilde{\zeta}^0_{\mathrm{u}}]$ and $\hat{E}[(\widetilde{\zeta}^0_{\mathrm{u}})^2]$ for all pixel $\mathbf{u}$.

[4] There are two other encoding modes for P-MB defined in H.264, i.e., skip mode and I_PCM mode. However, they are usually not involved in the PEED estimation process.

3) For mode = 1 : 20 (9+4 intra, 7 inter).

3a) If intra mode,

calculate $\hat{E}[\widetilde{\zeta}_{\mathrm{u}}^{k}]$ *by (3–22) for all pixels in the MB,*

go to 3b),

Else if $\hat{E}[\widetilde{\zeta}_{\mathrm{u+mv}_{\mathrm{u}}^{k}}^{k-j}] < \hat{f}_{\mathrm{u}}^{k} - 255$,

$\hat{E}[\widetilde{\zeta}_{\mathrm{u+mv}_{\mathrm{u}}^{k}}^{k-j} + \widetilde{\Delta}_{\mathrm{u}}^{k}\{\bar{r}, \bar{m}\}] = \hat{f}_{\mathrm{u}}^{k} - 255$,

$\hat{E}[(\widetilde{\zeta}_{\mathrm{u+mv}_{\mathrm{u}}^{k}}^{k-j} + \widetilde{\Delta}_{\mathrm{u}}^{k}\{\bar{r}, \bar{m}\})^2] = (\hat{f}_{\mathrm{u}}^{k} - 255)^2$,

Else if $\hat{E}[\widetilde{\zeta}_{\mathrm{u+mv}_{\mathrm{u}}^{k}}^{k-j}] > \hat{f}_{\mathrm{u}}^{k}$

$\hat{E}[\widetilde{\zeta}_{\mathrm{u+mv}_{\mathrm{u}}^{k}}^{k-j} + \widetilde{\Delta}_{\mathrm{u}}^{k}\{\bar{r}, \bar{m}\}] = \hat{f}_{\mathrm{u}}^{k}$,

$\hat{E}[(\widetilde{\zeta}_{\mathrm{u+mv}_{\mathrm{u}}^{k}}^{k-j} + \widetilde{\Delta}_{\mathrm{u}}^{k}\{\bar{r}, \bar{m}\})^2] = (\hat{f}_{\mathrm{u}}^{k})^2$,

Else

$\hat{E}[\widetilde{\zeta}_{\mathrm{u+mv}_{\mathrm{u}}^{k}}^{k-j} + \widetilde{\Delta}_{\mathrm{u}}^{k}\{\bar{r}, \bar{m}\}] = \hat{E}[\widetilde{\zeta}_{\mathrm{u+mv}_{\mathrm{u}}^{k}}^{k-j}]$,

$\hat{E}[(\widetilde{\zeta}_{\mathrm{u+mv}_{\mathrm{u}}^{k}}^{k-j} + \widetilde{\Delta}_{\mathrm{u}}^{k}\{\bar{r}, \bar{m}\})^2] = \hat{E}[(\widetilde{\zeta}_{\mathrm{u+mv}_{\mathrm{u}}^{k}}^{k-j})^2]$,

End

calculate $\hat{E}[\widetilde{\zeta}_{\mathrm{u}}^{k}]$ *by (4–1) for all pixels in the MB,*

3b) calculate $\hat{D}_{\mathrm{u}}^{k}$ *by (4–2) for all pixels in the MB,*

3c) estimate $D_{\mathrm{u},ETE}^{k}$ *by (3–24) for all pixels in the MB,*

3d) calculate R-D cost via (3–26) for each mode,

End

Via (3–27), select the mode with minimum R-D cost as the optimal mode for the MB.

5) ***Output****: the best mode for the MB.*

Using Theorem 3.1, we can design another ERRDO mode decision algorithm that produces the same solution as that of Algorithm 2, as Proposition 4 states.

Theorem 3.1. *(Decomposition Theorem) If there is no slice data partitioning, end-to-end distortion can be decomposed into a mode-dependent term and a mode-independent*

term, i.e.,

$$D^k_{\mathrm{u},ETE}(\omega_m) = D^k_{\mathrm{u},\overline{ETE}}(\omega_m) + C^k_{\mathrm{u}}. \tag{3–28}$$

where C^k_{u} *is independent of* ω_m *and*

$$D^k_{\mathrm{u},\overline{ETE}}(\omega_m) = (1 - P^k_{\mathrm{u}}) \cdot \{(f^k_{\mathrm{u}} - \hat{f}^k_{\mathrm{u}})^2 + D^k_{\mathrm{u}}(p) + 2(f^k_{\mathrm{u}} - \hat{f}^k_{\mathrm{u}}) \cdot E[\widetilde{\zeta}^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}} + \widetilde{\Delta}^k_{\mathrm{u}}\{\bar{r}, \bar{m}\}]\}. \tag{3–29}$$

Theorem 3.1 is proved in Appendix B.2.

Using Theorem 3.1, we only need to change two places in Algorithm 2 to obtain a new algorithm, which we call Algorithm A: first, replace Step 3c) in Algorithm 2 by "estimate $D^k_{\mathrm{u},\overline{ETE}}$ by (3–29) for all pixels in the MB"; second, replace 3d) in Algorithm 2 by "calculate R-D cost via $D^k_{\overline{ETE}}(\omega_m) + \lambda \cdot R(\omega_m)$ for each mode", where $D^k_{\overline{ETE}} = \sum_{\mathrm{u}\in\mathcal{V}^k_i} D^k_{\mathrm{u},\overline{ETE}}$.

Proposition 4. *If there is no slice data partitioning, Algorithm A and Algorithm 2 produce the same solution, i.e.,* $\hat{\omega}_m = \arg\min_{\omega_m}\{\hat{D}^k_{ETE}(\omega_m) + \lambda \cdot R(\omega_m)\} = \arg\min_{\omega_m}\{\hat{D}^k_{\overline{ETE}}(\omega_m) + \lambda \cdot R(\omega_m)\}$.

Proposition 4 is proved in Appendix B.3.

Note that $D^k_{\overline{ETE}}$ in (3–29) is not exactly the end-to-end distortion; but the decomposition in (3–28) can help reduce the complexity of some estimation algorithms, for example, LLN algorithm [29].

3.5.2 Complexity of RMPC-MS, ROPE, and LLN Algorithm

In this subsection, we compare the complexity of RMPC-MS algorithm with that of two popular mode decision algorithms, namely, ROPE algorithm and LLN algorithm, which are also based on pixel-level distortion estimation. To make a fair comparison, the same conditions should be used for all the three algorithms. Assume all the three algorithms use an error concealment scheme that conceals an erroneous pixel by the

pixel in the same position of the previous frame; then, $\check{e}_u^k = 0$ and $\check{\mathbf{mv}}_u^k = 0$; hence, $\varepsilon_u^k + \xi_u^k = f_u^k - \hat{f}_u^{k-1}$.

Here, the complexity is quantified by the number of additions (ADDs) and multiplications (MULs)[5] . If a subroutine (or the same set of operations) is invoked multiple times, it is counted only once since the temporary result is saved in the memory; for example, $\varepsilon_u^k + \xi_u^k$ in (4–2) and (4–1) is counted as one ADD. A substraction is counted as an addition. We only consider pixel-level operations; block-level operations, for example MV addition, are neglected. We ignore the complexity of those basic operations since their complexity is the same for all the three algorithms, such as motion compensation.

3.5.2.1 RMPC-MS algorithm

Let us first consider the complexity of RMPC-MS algorithm, i.e. Algorithm 2, for inter modes. In Algorithm 2, the worst case is $\hat{E}[\widetilde{\zeta}_{u+mv_u^k}^{k-j}] < \hat{f}_u^k - 255$. Under this case, there is one ADD and one square, i.e. MUL. The other two cases require only two copy operations, and so are neglected. Note that $\hat{f}_u^k - \hat{E}[\widetilde{\zeta}_{u+mv_u^k}^{k-j}] \leq 255$ with high probability, that is, $\hat{E}[\widetilde{\zeta}_{u+mv_u^k}^{k-j}] < \hat{f}_u^k - 255$ is relatively rare. Therefore, in most cases, there are only two copy operations in the loop. Calculating the second moment of $\widetilde{\zeta}_u^k$ needs 4 ADDs and 4 MULs as in (4–2). Similarly, calculating the first moment of $\widetilde{\zeta}_u^k$ needs 2 ADDs and 2 MULs as in (4–1). Finally, calculating the end-to-end distortion needs 3 ADDs and 2 MULs as in (3–24). Hence, the worst case of calculating the end-to-end distortion for each pixel is 10 ADDs and 9 MULs. Note that in most cases, the complexity is 9 ADDs and 8 MULs for inter modes as shown in Table 3-1.

Note that since P_u^k is the same for all pixels in one MB, we do not need to calculate $1 - P_u^k$ for each pixel. Multiplying by 2 can be achieved by a shift operation; so it is not counted as one MUL.

[5] Those minor operations, such as memory copy, shift, and conditional statement, are neglected for all algorithms.

For Intra modes, we know that $\widetilde{\zeta}^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}} + \widetilde{\Delta}^k_{\mathrm{u}}\{\bar{r}, \bar{m}\} = 0$ from Chapter 2. Therefore, the complexity of intra mode is reduced to 3 ADDs and 3 MULs in (4–2), 1 ADDs and 1 MULs in (3–22). As a result, the end-to-end distortion for each pixel is 7 ADDs and 6 MULs for each intra mode.

In H.264, there are 7 inter modes and 13 intra modes; therefore there are a total of 154 ADDs and 134 MULs for each pixel in most cases. In the worst case, there are a total of 161 additions and 141 MULs for each pixel, where the additional computation comes from the consideration of clipping effect.

Memory Requirement Analysis: To estimate the end-to-end distortion by Algorithm 2, the first moment and the second moment of the reconstructed error of the best mode should be stored after the mode decision. Therefore, 2 units of memory are required to store those two moments for each pixel. Note that the first moment takes values in $\{-255, -254, \cdots, 255\}$, i.e., 8 bits plus 1 sign bit per pixel, and the second moment takes values in $\{0, 1, \cdots, 255^2\}$, i.e., 16 bits per pixel.

3.5.2.2 ROPE algorithm

In ROPE algorithm, the moment estimation formulae for inter prediction and intra prediction are different. For inter modes, calculating the first moment needs 2 ADDs and 2 MULs; calculating the second moment needs 3 ADDs and 4 MULs; calculating the end-to-end distortion needs 2 ADDs and 2 MULs. For intra modes, calculating the first moment needs 1 ADD and 2 MULs; calculating the second moment needs 1 ADD and 3 MULs. Hence, an inter mode needs 7 ADDs and 8 MULs; an intra mode needs 4 ADDs and 7 MULs. For H.264, the total complexity for each pixel is 101 ADDs and 147 MULs.

Note that when we implement ROPE in JM16.0, we find that ROPE algorithm causes out-of-range values for both the first moment and the second moment due to the neglect of clipping noise. Experimental results show that ROPE may produce a negative value as the estimate for distortion, which violates the requirement that (true) distortion

must be non-negative. Hence, in a practical system the estimated result from ROPE algorithm needs to be clipped into a legitimate value; this will incur a higher complexity.

Memory Requirement Analysis: To estimate the end-to-end distortion by ROPE algorithm, the first moment and the second moment of the reconstructed pixel value of the best mode should be stored after the mode decision. Therefore, 2 units of memory are required to store the two moments for each pixel. The first moment takes values in $\{0, 1, \cdots, 255\}$, i.e., 8 bits per pixel; the second moment takes values in $\{0, 1, \cdots, 255^2\}$, i.e., 16 bits per pixel. Note that in the original ROPE algorithm [4], the values of the two moments are not bounded since the propagated errors are never clipped.

3.5.2.3 LLN algorithm

In JM16.0, LLN algorithm uses the same decomposition method as Theorem 3.1 for mode decision [29]. In such a case, for inter modes, reconstructing the pixel value in one simulated decoder needs 1 ADD; calculating the end-to-end distortion needs 1 ADD and one MUL. For intra modes, there is no additional reconstruction for all simulated decoders since the newly induced errors are not considered; therefore, calculating the end-to-end distortion needs 1 ADD and 1 MUL. Suppose the number of simulated decoders at the encoder is N_d, the complexity for LLN algorithm is $27N_d$ ADDs and $20N_d$ MULs. The default number of simulated decoders in JM16.0 is 30, which means the complexity for LLN algorithm is 810 ADDs and 600 MULs. Thirty simulated decoders is suggested in Ref. [6]. In our experiment, we find that if the number of simulated decoders is less than 30, the estimated distortion exhibits high degree of randomness (i.e., having a large variance); however, if the number of simulated decoders is larger than 50, the estimated distortion is quite stable (i.e., having a small variance).

Note that the error concealment operations in LLN algorithm are required but not counted in the complexity since it is done after the mode decision. However, even without considering the extra error concealment operations, the complexity of LLN algorithm is still much higher than RMPC-MS and ROPE. Increasing the number of

Table 3-1. Complexity Comparison

		computational complexity	memory requirement
RMPC-MS	inter mode	9 ADDs, 8 MULs (worst 10 ADDs, 9 MULs)	25 bits/pixel
	intra mode	7 ADDs, 6 MULs	
	total complexity	154 ADDs, 134 MULs (worst 161 ADDs, 141 MULs)	
ROPE	inter mode	7 ADDs, 8 MULs (more with clipping)	24 bits/pixel
	intra mode	4 ADDs, 7 MULs	
	total complexity	101 ADDs, 147 MULs (more with clipping)	
LLN	inter mode	$2N_d$ ADDs, N_d MULs (more with error concealment)	$8N_d$ bits/pixel
	intra mode	N_d ADDs, N_d MULs	
	total complexity	$27N_d$ ADDs, $20N_d$ MULs (more with error concealment)	

simulated decoders at the encoder can improve estimation accuracy but at the cost of linear increase of computational complexity.

Memory Requirement Analysis: To estimate the end-to-end distortion by LLN algorithm, for each simulated decoder, each reconstructed pixel value of the best mode should be stored after the mode decision. Therefore, the encoder needs N_d units of memory to store the reconstructed pixel value. A reconstructed pixel takes values in $\{0, 1, \cdots, 255\}$, i.e., $8N_d$ bits per pixel.

Table 3-1 shows the complexity of the three algorithms.

3.6 Experimental Results

In Section 3.6.1, we compare the estimation accuracy of RMPC-FTD algorithm to that of the existing models under different channel conditions; we also compare their robustness against imperfect estimate of PEP. In Section 3.6.2, we compare the R-D

performance of RMPC-MS and existing mode decision algorithms for H.264; we also compare them under interpolation filter and deblocking filter.

To collect the statistics and test the algorithms, all possible channel conditions should be tested for every video sequence[6] . However, estimating transmission distortion and collecting the statistics of each video sequence under all possible error events are tedious tasks since only command line interface or configuration file is available in current open source H.264 reference code, such as JM and x264. To analyze the statistics and verify our algorithm, our lab has developed a software tool, called Video Distortion Analysis Tool (VDAT), which provides a friendly Graphical User Interface (GUI). VDAT implements channel simulator, supports different video codec, computes the statistics, and supports several distortion estimation algorithms[7] . VDAT is used in all the experiments in this section.

3.6.1 Estimation Accuracy and Robustness

In this section, we use Algorithm 1 to estimate FTD and compare it with Stuhlmuller's model [8] and Dani's model [9]. To evaluate estimation accuracy, we compare the estimated distortion of different algorithms with true distortion for 50 frames under the case of no acknowledgement feedback.

3.6.1.1 Experiment setup

To implement the estimation algorithms, all transmission distortion related statistics should be collected for all random variables, such as residual, motion vector, reconstructed pixel value, residual concealment error, MV concealment error, propagated error, clipping noise. All such statistics are collected from video codec

[6] http://trace.eas.asu.edu/yuv/index.html

[7] Interested readers can download all the VDAT source codes at: http://users.ecel.ufl.edu/ zhifeng/project/VDAT/index.htm.

JM14.0[8] . All tested video sequences are in CIF format, and each frame is divided into three slices. To support the slice data partitioning, we use the extended profile as defined in H.264 specification Annex A [22]. To provide unequal error protection (UEP), we let MV packets experience lower PEP than residual packets. The first frame of each coded video sequence is an I-frame, and the subsequent frames are all P-frames. In the experiment, we let the first I-frame be correctly received, and all the following P-frames go through an error-prone channel with controllable PEP. We set QP=28 for all the frames.

Each video sequence is tested under several channel conditions with UEP. Due to the space limit, we only present the experimental results for video sequences 'foreman' and 'stefan'. Experimental results for other video sequences can be found online[9] . For each sequence, two wireless channel conditions are tested: for good channel condition, residual PEP is 2% and MV PEP is 1%; for poor channel condition, residual PEP is 10% and MV PEP is 5%. For each PEP setting of each frame, we do 600 simulations and take the average to mitigate the effect of randomness of simulated channels on instantaneous distortion.

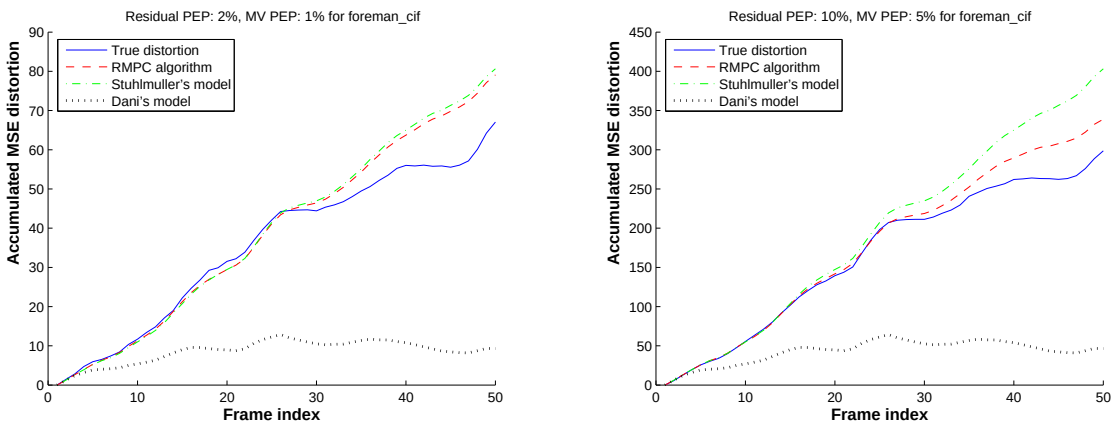
3.6.1.2 Estimation accuracy of different estimation algorithms

Fig. 3-1 shows the estimation accuracy of RMPC-FTD algorithm, Stuhlmuller's model in Ref. [8] and Dani's model in Ref. [9] for sequence 'foreman'. Fig. 3-2 shows their estimation accuracy for sequence 'stefan'. We can see that RMPC-FTD algorithm achieves the most accurate estimate. Since the superposition algorithm in Stuhlmuller's model neglects the effect of clipping noise and negative correlation between MV concealment error and propagated error, it over-estimates transmission distortion as shown in Fig. 3-2. However, since the clipping effect and the correlation caused

[8] http://iphome.hhi.de/suehring/tml/download/old_jm/jm14.0.zip

[9] http://www.mcn.ece.ufl.edu/public/zhifeng/project/VDAT/journal/

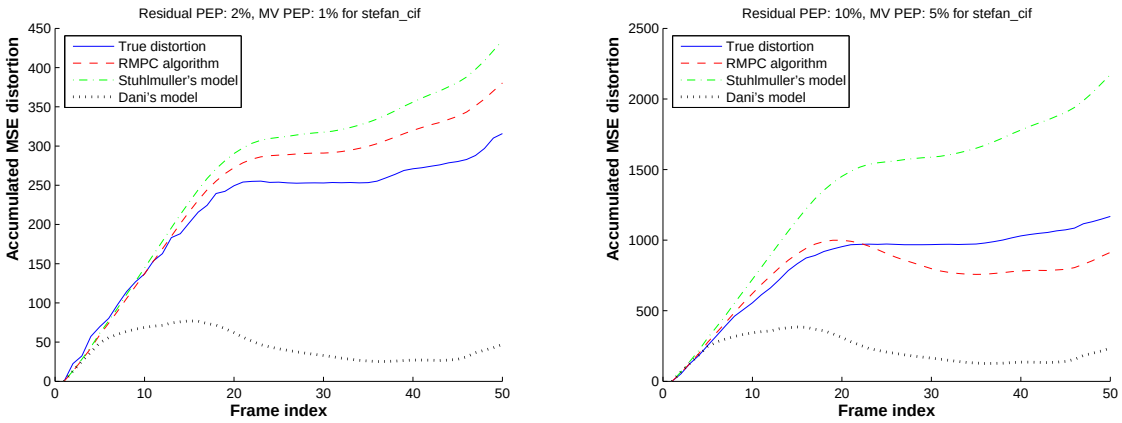
distortion is small for low motion sequence under low PEP as proved in Chapter 2, linear model is quite accurate as shown in Fig. 3-1(a). Notice that in foreman sequence under good channel, the estimated distortion different from ground truth is only about $MSE = 12$ after accumulated with 50 frames without feedback. In Ref. [9], authors claim that the larger the fraction of pixels in the reference frame to be used as reference pixels, the larger the transmission errors propagated from the reference frame. However, due to randomness of motion vectors, the probability that a pixel with error is used as reference is the same as the probability that a pixel without error is used as reference. Therefore, the number of pixels in the reference frame being used for motion prediction has nothing to do with the fading factor. As a result, the algorithm in Ref. [9] under-estimates transmission distortion as shown in Fig. 3-1 and Fig. 3-2.



(a) (b)

Figure 3-1. Transmission distortion D^k vs. frame index k for 'foreman': (a) good channel, (b) poor channel.

In our experiment, we observe that 1) the higher the propagated distortion, the smaller the propagation factor; and 2) the higher percentage of reconstructed pixel values near 0 or 255, the smaller the propagation factor. These two phenomena once more verify that the propagation factor is a function of all samples of reconstructed pixel value and sample variance of propagated error as proved in Chapter 2. These phenomena could be explained by (3–8) in that 1) α is a decreasing function of b for



(a) (b)

Figure 3-2. Transmission distortion D^k vs. frame index k for 'stefan': (a) good channel, (b) poor channel.

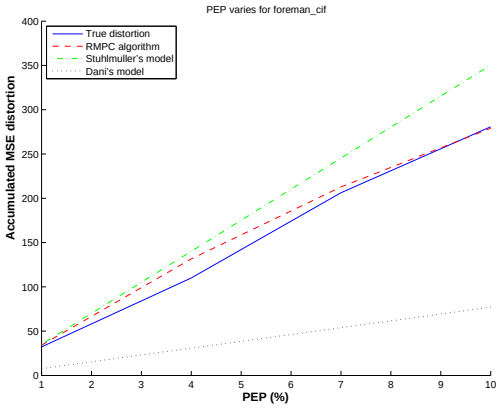


Figure 3-3. Transmission distortion D^k vs. PEP for 'foreman'.

$b > 0$; 2) α is an increasing function of y for $0 \leq y \leq 127$ and a decreasing function of y for $128 \leq y \leq 255$. We also note that a larger sample variance of propagated error causes α to be less sensitive to the change of reconstructed pixel value, while a larger deviation of reconstructed pixel value from 128 causes α to be less sensitive to the change of sample variance of propagated error.

To further study estimation accuracy, we test the estimation algorithms under many different channel conditions. Fig. 3-3 and Fig. 3-4 show the estimation accuracy under PEP varying from 1% to 10%. In both figures, RMPC-FTD algorithm achieves the most accurate distortion estimation under all channel conditions.

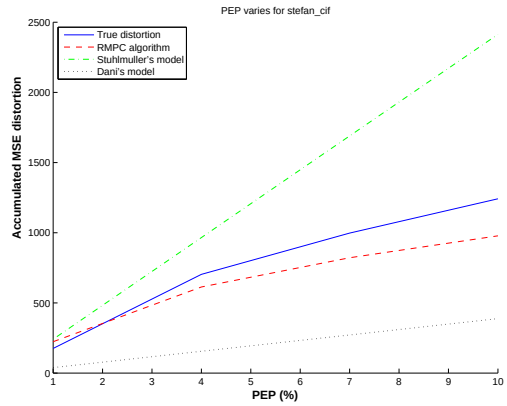


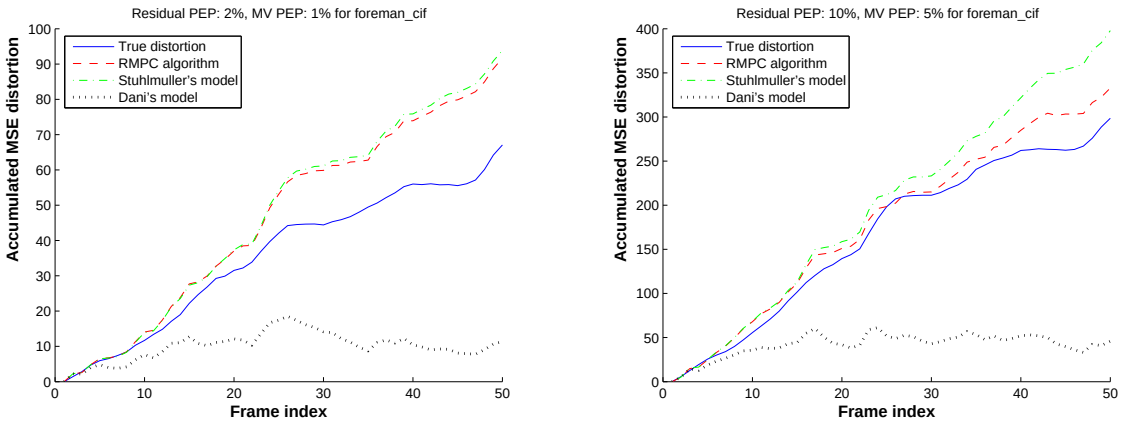
Figure 3-4. Transmission distortion D^k vs. PEP for 'stefan'.

3.6.1.3 Robustness of different estimation algorithms

In Section 3.6.1.2, we assume PEP is perfectly known at the encoder. However, in a real wireless video communication system, PEP is usually not perfectly known at the encoder; i.e., there is a random estimation error between the true PEP and the estimated PEP. Hence, it is important to evaluate the robustness of the estimation algorithms against PEP estimation error. To simulate imperfect PEP estimation, for a given true PEP denoted by P_{true}, we assume the estimated PEP is a random variable and is uniformly distributed in $[0, 2 \times P_{true}]$; e.g., if $P_{true} = 10\%$, the estimated PEP is uniformly distributed in $[0, 20\%]$.

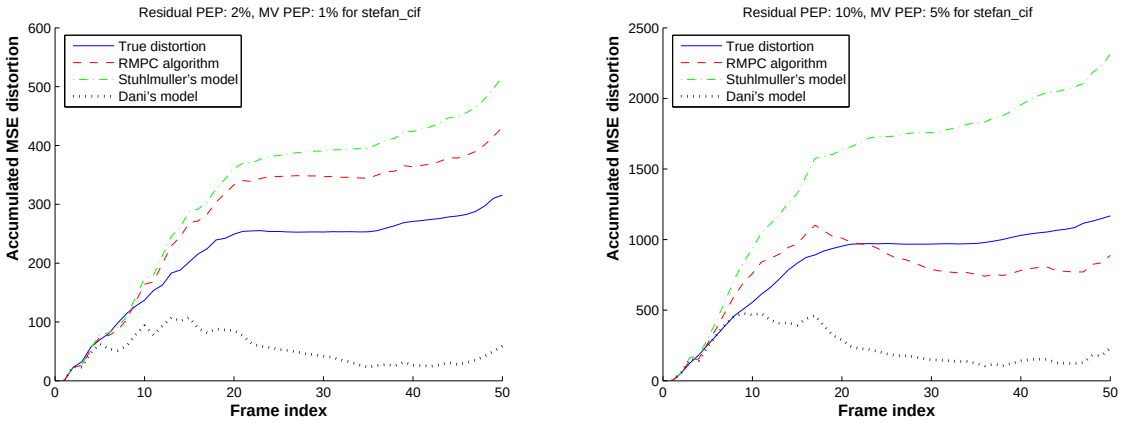
Figs. 3-5 and 3-6 show the estimation accuracy of the three algorithms for 'foreman' and 'stefan', respectively, under imperfect knowledge of PEP at the encoder. From the two figures, it is observed that compared to the case under perfect knowledge of PEP at the encoder, for both Stuhlmuller's model and Dani's model, imperfect knowledge of PEP may cause increase or decrease of the gap between the estimated distortion and the true distortion. Specifically, for Stuhlmuller's model, if the PEP is under-estimated, the gap between the estimated distortion and the true distortion decreases, compared to the case under perfect knowledge of PEP; for Dani's model, if the PEP is over-estimated, the gap between the estimated distortion and the true distortion decreases, compared to the case under perfect knowledge of PEP. In contrast, RMPC-FTD algorithm is more

robust against PEP estimation error, and provides more accurate distortion estimate than Stuhlmuller's model and Dani's model.



(a) (b)

Figure 3-5. Transmission distortion D^k vs. frame index k for 'foreman' under imperfect knowledge of PEP: (a) good channel, (b) poor channel.



(a) (b)

Figure 3-6. Transmission distortion D^k vs. frame index k for 'stefan' under imperfect knowledge of PEP: (a) good channel, (b) poor channel.

3.6.2 R-D Performance of Mode Decision Algorithms

In this subsection, we compare the R-D performance of Algorithm 2 with that of ROPE and LLN algorithms for mode decision in H.264. To compare all algorithms under the multi-reference picture motion compensated prediction, we also enhance the original ROPE algorithm [4] with multi-reference capability.

3.6.2.1 Experiment setup

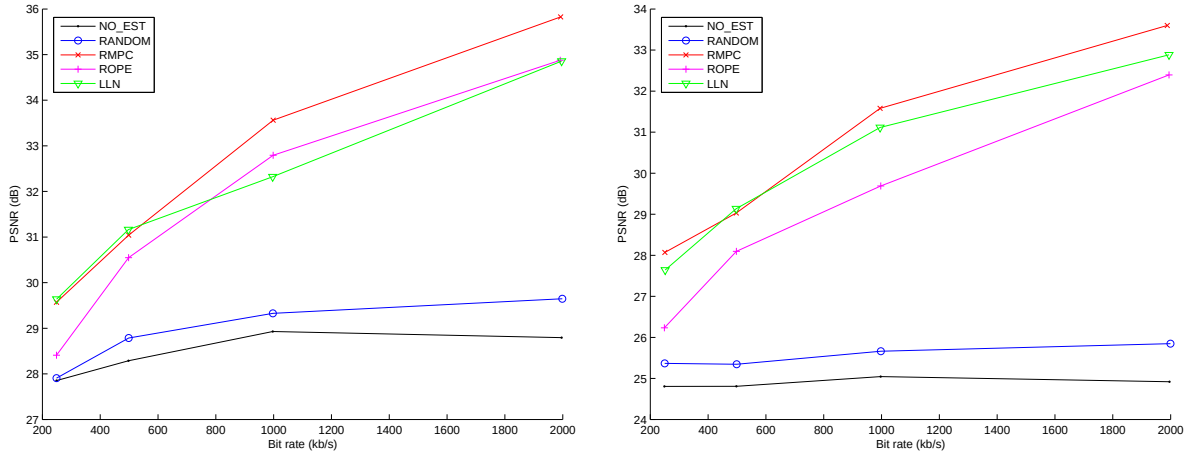
JM16.0 encoder and decoder is used in the experiments. To support more advanced techniques in H.264, we use the high profile defined in H.264 specification Annex A [22]. We conduct experiments for five schemes, that is, three ERRDO schemes, i.e., RMPC-MS, LLN, ROPE; random intra update; and default SCRDO scheme with no transmission distortion estimation. All the tested video sequences are in CIF resolution with 30fps. Each coded video sequence is tested under different PEP settings from 0.5% to 5%. Each video sequence is coded for its first 100 frames with 3 slices per frame. The error concealment method used is to copy the pixel value in the same position of the previous frame. The first frame is assumed to be correctly received.

The encoder setting is given as below. No slice data partitioning is used; constrained intra prediction is enabled; the number of reference frames is 3; B-MBs are not included; only 4x4 transform is used; CABAC is enabled for entropy coding; in LLN algorithm, the number of simulated decoders is 30.

3.6.2.2 R-D performance under no interpolation filter and no deblocking filter

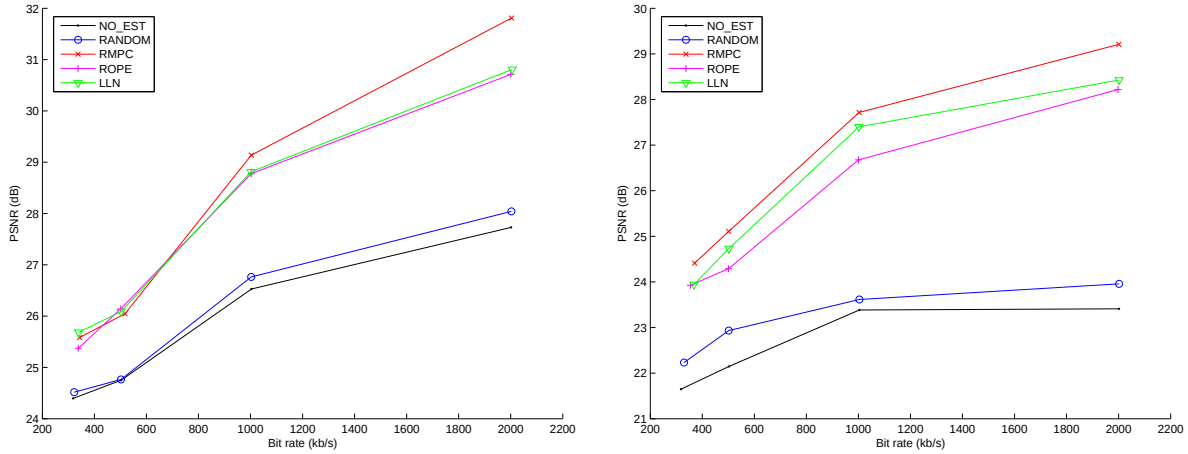
In the experiments of this subsection, both the deblocking filter and the interpolation filter with fractional MV in H.264 are disabled. Due to the space limit, we only show the plot of PSNR vs. bit rate for video sequences 'foreman' and 'football' under $PEP = 2\%$ and $PEP = 5\%$, with rate control enabled. Figs. 3-7 and 3-8 show PSNR vs. bit rate for 'foreman' and 'football', respectively. The experimental results show that RMPC-MS achieves the best R-D performance; LLN and ROPE achieves similar performance and the second best R-D performance; the random intra update scheme (denoted by 'RANDOM') achieves the third best R-D performance; the SCRDO scheme (denoted by 'NO_EST') achieves the worst R-D performance.

LLN has poorer R-D performance than RMPC-MS; this may be because 30 simulated decoders are still not enough to achieve reliable distortion estimate although LLN with 30 simulated decoders already incurs much higher complexity than RMPC-MS.



(a) (b)

Figure 3-7. PSNR vs. bit rate for 'foreman', with no interpolation filter and no deblocking filter: (a) PEP=2%, (b) PEP=5%.



(a) (b)

Figure 3-8. PSNR vs. bit rate for 'football', with no interpolation filter and no deblocking filter: (a) PEP=2%, (b) PEP=5%.

The reason why RMPC-MS achieves better R-D performance than ROPE, is due to the consideration of clipping noise in RMPC-MS. Debug messages show that, without considering the clipping noise, ROPE over-estimates the end-to-end distortion for inter modes; hence ROPE tends to select intra modes more often than RMPC-MS and LLN, which leads to higher encoding bit rate in ROPE; as a result, the PSNR gain achieved by ROPE is compromised by its higher bit rate. To verify this conjecture, we test all

sequences under the same Quantization Parameter (QP) settings without rate control. we observe that ROPE algorithm always produces higher bit rate than other schemes.

Table 3-2 shows the average PSNR gain (in dB) of RMPC-MS over ROPE and LLN for different video sequences and different PEP. The average PSNR gain is obtained by the method in Ref. [30], which measures average distance (in PSNR) between two R-D curves. From Table 3-2, we see that RMPC-MS achieves an average PSNR gain of 1.44dB over ROPE for 'foreman' under $PER = 5\%$; and it achieves an average PSNR gain of 0.89dB over LLN for 'foreman' sequence under $PEP = 1\%$.

Table 3-2. Average PSNR gain (in dB) of RMPC-MS over ROPE and LLN

Sequence	PEP	RMPC vs. ROPE	RMPC vs. LLN
coastguard	5%	0.75	0.04
	2%	0.26	0.23
	1%	0.15	0.20
	0.5%	0.16	0.17
football	5%	0.88	0.42
	2%	0.26	0.22
	1%	0.19	0.28
	0.5%	0.22	0.15
foreman	5%	**1.44**	0.28
	2%	0.74	0.53
	1%	0.61	**0.89**
	0.5%	0.30	0.65
mobile	5%	0.51	0.29
	2%	0.14	0.23
	1%	0.15	0.28
	0.5%	0.10	0.19

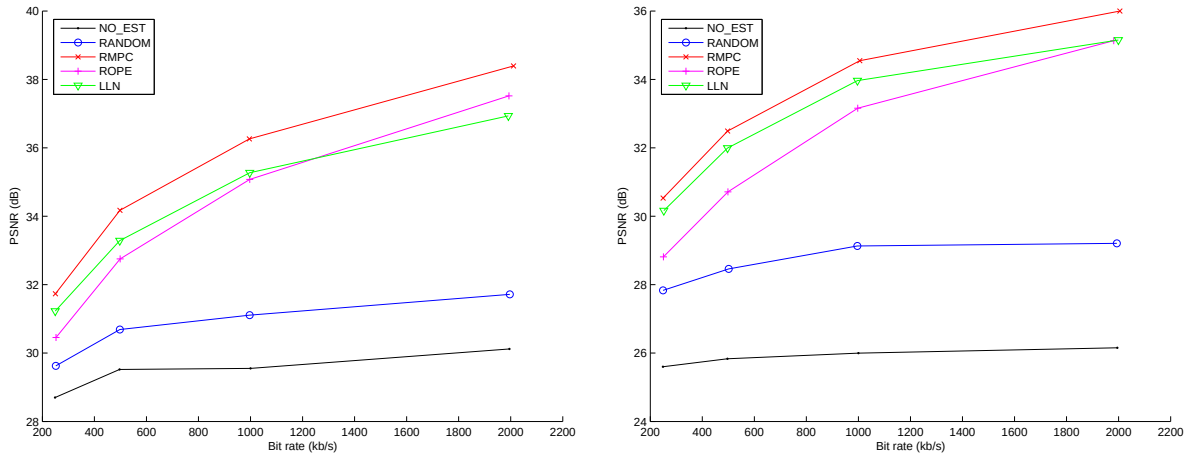
3.6.2.3 R-D performance with interpolation filter and deblocking filter

In H.264, interpolation filter provides notable objective (PSNR) gain and deblocking filter provides notable subjective gain. To support the interpolation filter with fractional MV in H.264 [20], we extend Algorithm 2 by using the nearest neighbor to approximate

the reference pixel pointed by a fractional MV. In addition, deblocking filter is also enabled in JM16.0 to compare RMPC-MS, ROPE and LLN algorithms.

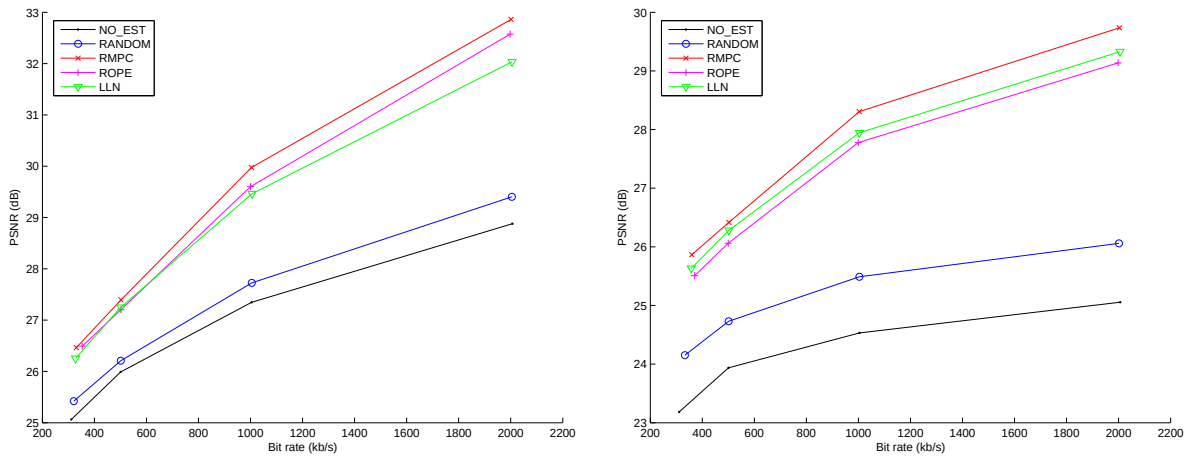
Note that both RMPC-MS and ROPE are derived without considering filtering operations. Due to high spatial correlation between adjacent pixels, the averaging operation induced by a filter will produce many cross-correlation terms for estimating distortion in a subpixel position. Yang et al. [17] enhance the original ROPE algorithm with interpolation filter in H.264. However, their algorithm requires 1 square root operation, 1 exponentiation operation, and 2 multiplication operations for calculating each cross-correlation term. Since a six-tap interpolation filter is used in H.264 for subpixel accuracy of motion vector, there are 15 cross-correlation terms for calculating each subpixel distortion. Therefore, the complexity of their algorithm is very high and may not be suitable for real-time encoding. In this subsection, we use a very simple but R-D efficient method to estimate subpixel distortion. Specifically, we choose the nearest integer pixel around the subpixel, and use the distortion of the nearest integer pixel as the estimated distortion for the subpixel. Note that this simple method is not aimed at extending RMPC-MS and ROPE algorithms, but just to compare the R-D performances of these two algorithms for H.264 with fractional MV for motion compensation.

We first show the experiment results with interpolation filter but with no deblocking filter as in Figs. 4-1 and 3-10. From Figs. 4-1 and 3-10, we observe the same result as shown in Section 4.3.2: RMPC-MS achieves better R-D performance than LLN and ROPE algorithms. From Figs. 4-1 and 3-10, we also can see that each of the five algorithms achieves higher PSNR than its corresponding scheme with no interpolation filter; this means the simple method is valid. We also observe from Table 4-1 that in this case, RMPC-MS achieves an average PSNR gain of 2.97dB over ROPE for sequence 'mobile' under $PEP = 0.5\%$; and it achieves an average PSNR gain of 1.13dB over LLN for 'foreman' under $PEP = 1\%$.



(a) (b)

Figure 3-9. PSNR vs. bit rate for ‘foreman’, with interpolation and no deblocking: (a) PEP=2%, (b) PEP=5%.



(a) (b)

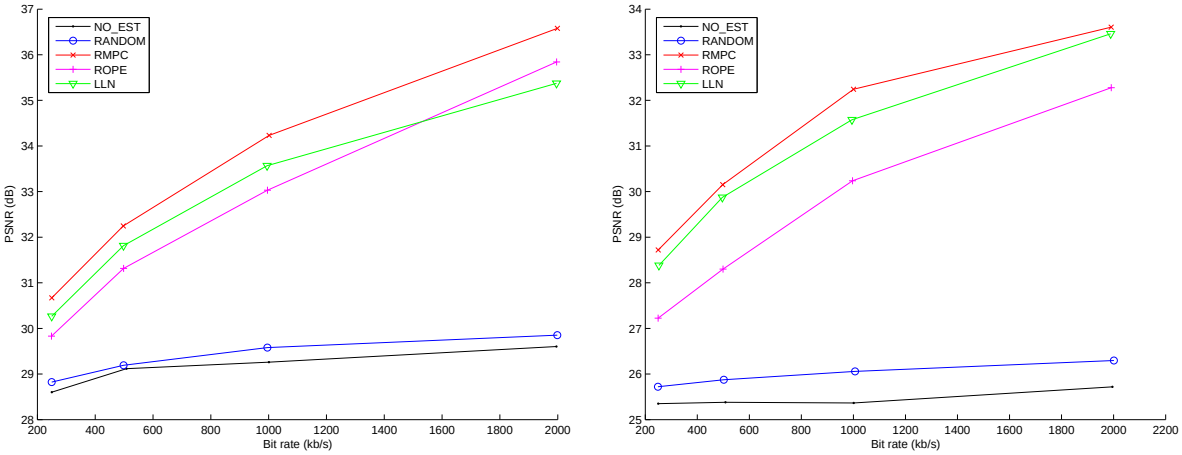
Figure 3-10. PSNR vs. bit rate for ‘football’, with interpolation and no deblocking: (a) PEP=2%, (b) PEP=5%.

We also show the experiment results with both interpolation filter and deblocking filter as shown in Figs. 3-11 and 3-12. It is interesting to see that each of the five algorithms with interpolation filter and deblocking filter achieves poorer R-D performance than the corresponding one with interpolation filter and no deblocking filter. That is, adding deblocking filter degrades the R-D performance of each algorithm since their estimated distortions become less accurate. In this case, ROPE sometimes performs better than RMPC-MS; this can be seen in Fig. 3-12, which is also the only case we

Table 3-3. Average PSNR gain (in dB) of RMPC-MS over ROPE and LLN under interpolation filtering

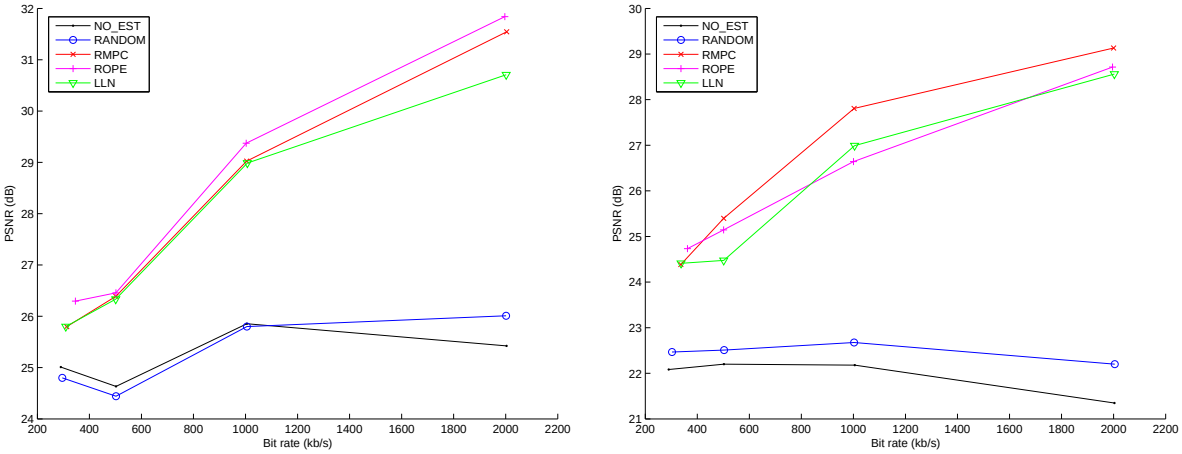
Sequence	PEP	RMPC vs. ROPE	RMPC vs. LLN
coastguard	5%	0.49	0.23
	2%	0.38	0.28
	1%	0.43	0.38
	0.5%	0.56	0.31
football	5%	0.45	0.27
	2%	0.24	0.38
	1%	0.25	0.35
	0.5%	0.30	0.23
foreman	5%	1.51	0.56
	2%	1.25	0.95
	1%	1.20	**1.13**
	0.5%	1.25	1.07
mobile	5%	0.92	0.30
	2%	1.64	0.57
	1%	2.58	0.35
	0.5%	**2.97**	0.33

have observed that ROPE performs better than RMPC-MS. This may be because RMPC-MS has a higher percentage of inter modes than ROPE. Since the deblocking operation is executed after the error concealment as in JM16.0, for intra prediction, deblocking filter only affects the estimated distortion if the packet is lost; for inter prediction, deblocking filter always impacts the estimated distortion. Therefore, the estimation accuracy for inter prediction suffers from deblocking filter more than that for intra prediction. Thus, it is likely that more inter modes in RMPC-MS cause higher PSNR drop in Fig. 3-12.



(a) (b)

Figure 3-11. PSNR vs. bit rate for ‘foreman’, with interpolation and deblocking: (a) PEP=2%, (b) PEP=5%.



(a) (b)

Figure 3-12. PSNR vs. bit rate for ‘football’, with interpolation and deblocking: (a) PEP=2%, (b) PEP=5%.

CHAPTER 4
THE EXTENDED RMPC ALGORITHM FOR ERROR RESILIENT RATE DISTORTION OPTIMIZED MODE DECISION

In this chapter, We first prove a new theorem for calculating the second moment of a weighted sum of correlated random variables without the requirement of their probability distribution. Then, we apply the theorem to extend the RMPC-MS algorithm in Chapter 3 to support the subpixel-level Mean Square Error (MSE) distortion estimation.

4.1 An Overview on Subpixel-level End-to-end Distortion Estimation for a Practical Video Codec

Existing pixel-level algorithms, e.g., the RMPC algorithm, are based on the integer pixel MV assumption to derive an estimate of D_{u}^{k}. Therefore, their application in state-of-the-art encoders is limited due to the possible use of fractional motion compensation. For the RMPC algorithms, if the MV of one block for encoding is fractional, the MV has to be rounded to the nearest integer. This block will use the reference block pointed to by the rounded MV as a reference. However, in state-of-the-art codecs, such as H.264 [22] and HEVC proposals [31], an interpolation filter is used to interpolate a reference block if the MV is fractional. Therefore, the distortion of nearest neighbor approximation is not optimal for such an encoder. As a result, we need to extend the existing RMPC algorithm to optimally estimate the distortion for blocks with interpolation filtering.

Some subpixel-level end-to-end distortion estimation algorithms have been proposed to assist mode decision as in Ref. [5, 17, 32]. In the H.264/AVC JM reference software [33], the LLN algorithm proposed in Ref. [5] is adopted to estimate the end-to-end distortion for mode decision. However, in the LLN algorithm more decoders lead to higher computational complexity and larger memory requirements. Also for the same video sequence and the same PEP, different encoders may have different estimated distortions due to the randomly produced error events at each encoder. In Ref. [32], the authors extend ROPE for the H.264 encoder by using the upper bound,

obtained from the Cauchy-Schwarz approximation, to approximate the cross-correlation terms. However, such an approximation requires very high complexity. For example, for an N-tap filter interpolation, each subpixel requires N integer multiplications[1] for calculating the second moment terms; $N(N-1)/2$ floating-point multiplications and $N(N-1)/2$ square root operations for calculating the cross-correlation terms; and $N(N-1)/2+N-1$ additions and 1 shift for calculating the estimated distortion. On the other hand, the upper bound approximation is not accurate for practical video sequences since it assumes that correlation coefficient is 1, for any two neighboring pixels. In Ref. [17], authors propose some correlation coefficient models to approximate the correlation coefficient of two pixels as functions, e.g., an exponentially decaying function, of their distance. However, due to the random behavior of individual pixel samples, the statistical model does not produce an accurate pixel-level distortion estimate. In addition, such correlation coefficient model approximations incur extra complexity compared to the Cauchy-Schwarz upper bound approximation, i.e., they need additional $N(N-1)/2$ exponential operations and $N(N-1)/2$ floating-point multiplications for each subpixel. Therefore, the complexity incurred is prohibitively high for real-time video encoders. On the other hand, since both the Cauchy-Schwarz upper bound approximation and the correlation coefficient model approximation need the floating-point multiplications, additional round-off errors are unavoidable, which further reduce their estimation accuracy.

In Chapter 2, we propose a divide-and-conquer method to quantify the effects of 1) residual concealment error, 2) Motion Vector (MV) concealment error, 3) propagation error and clipping noise, and 4) correlations between any two of them, on transmission

[1] One common method to simplify the multiplication of an integer variable and a fractional constant is as below: first scale up the fractional constant by a certain factor; round it off to an integer; then do integer multiplication; finally scale down the product.

distortion. Based on our theoretical results, we proposed the RMPC algorithm in Chapter 3 for rate-distortion optimized mode decision with pixel-level end-to-end distortion estimation. Since the correlation between the transmission errors of neighboring pixels is much smaller and more stable than the correlation between the reconstructed values of neighboring pixels, the RMPC algorithm is easier than ROPE to be extended for supporting subpixel-level end-to-end distortion estimation. In this chapter, we first theoretically derive the second moment of a weighted sum of correlated random variables as a closed-form function of the second moments of those individual random variables. Then we apply this result to design a very low complexity but accurate algorithm for mode decision. This algorithm is referred to as Extended RMPC (ERMPC). The ERMPC algorithm only requires N integer multiplications, $N-1$ additions, and 1 shift to calculate the second moment for each subpixel. Experimental results show that, ERMPC achieves an average PSNR gain of 0.25dB over the existing RMPC algorithm for the 'mobile' sequence when PEP equals 2%; and ERMPC achieves an average PSNR gain of 1.34dB over the the LLN algorithm for the 'foreman' sequence when PEP equals 1%.

The rest of this chapter is organized as follows. In Section 4.2, we first derive the general theorem for the second moment of a weighted sum of correlated random variables, and then apply this theorem to design a low-complexity and high-accuracy algorithm for mode decision. Section 5.5 shows the experimental results, which demonstrates the better R-D performance and subjective performance of the ERMPC algorithm over existing algorithms for H.264 mode decision in error prone environments.

4.2 The Extended RMPC Algorithm for Mode Decision

In this section, we first state the problem of pixel-level distortion estimation in a practical video codec. Then we derive a general theorem for any second moment of a weighted sum of correlated random variables for helping solve the problem. At last, we

apply the theorem in designing a low-complexity and high-accuracy distortion estimation algorithm for mode decision.

4.2.1 Subpixel-level Distortion Estimation

In Chapter 3, we know that $E[\widetilde{\zeta}_{\mathrm{u}}^{k}]$ and $E[(\widetilde{\zeta}_{\mathrm{u}}^{k})^2]$ can be recursively calculated by

$$E[\widetilde{\zeta}_{\mathrm{u}}^{k}] = P_{\mathrm{u}}^{k} \cdot (\varepsilon_{\mathrm{u}}^{k} + \xi_{\mathrm{u}}^{k} + E[\widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1}]) + (1 - P_{\mathrm{u}}^{k}) \cdot E[\widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-j} + \widetilde{\Delta}_{\mathrm{u}}^{k}\{\bar{r}, \bar{m}\}], \tag{4–1}$$

and

$$\begin{aligned} E[(\widetilde{\zeta}_{\mathrm{u}}^{k})^2] = P_{\mathrm{u}}^{k} \cdot ((\varepsilon_{\mathrm{u}}^{k} + \xi_{\mathrm{u}}^{k})^2 + 2(\varepsilon_{\mathrm{u}}^{k} + \xi_{\mathrm{u}}^{k}) \cdot E[\widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1}] + E[(\widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1})^2]) \\ + (1 - P_{\mathrm{u}}^{k}) \cdot E[(\widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-j} + \widetilde{\Delta}_{\mathrm{u}}^{k}\{\bar{r}, \bar{m}\})^2], \end{aligned} \tag{4–2}$$

where $\varepsilon_{\mathrm{u}}^{k} \triangleq \hat{e}_{\mathrm{u}}^{k} - \check{e}_{\mathrm{u}}^{k}$ is the residual concealment error when the residual packet is lost; $\xi_{\mathrm{u}}^{k} \triangleq \hat{f}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-1} - \hat{f}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1}$ is the MV concealment error when the MV packet is lost; $E[\widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1}]$ and $E[(\widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1})^2]$ in the $k-1$-th frame can be recursively calculated by (4–1) and (4–2). P_{u}^{k} is the pixel error probability. Denote $\hat{E}(\cdot)$ as the estimate of $E(\cdot)$; $E[\widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-j} + \widetilde{\Delta}_{\mathrm{u}}^{k}\{\bar{r}, \bar{m}\}]$ and $E[(\widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-j} + \widetilde{\Delta}_{\mathrm{u}}^{k}\{\bar{r}, \bar{m}\})^2]$ can be estimated by

$$\hat{E}[\widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-j} + \widetilde{\Delta}_{\mathrm{u}}^{k}\{\bar{r}, \bar{m}\}] = \begin{cases} \hat{f}_{\mathrm{u}}^{k} - 255, & \hat{E}[\widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-j}] < \hat{f}_{\mathrm{u}}^{k} - 255 \\ \hat{f}_{\mathrm{u}}^{k}, & \hat{E}[\widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-j}] > \hat{f}_{\mathrm{u}}^{k} \\ \hat{E}[\widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-j}], & \hat{f}_{\mathrm{u}}^{k} - 255 \le \hat{E}[\widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-j}] \le \hat{f}_{\mathrm{u}}^{k}, \end{cases} \tag{4–3}$$

and

$$\hat{E}[(\widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-j} + \widetilde{\Delta}_{\mathrm{u}}^{k}\{\bar{r}, \bar{m}\})^2] = \begin{cases} (\hat{f}_{\mathrm{u}}^{k} - 255)^2, & \hat{E}[\widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-j}] < \hat{f}_{\mathrm{u}}^{k} - 255 \\ (\hat{f}_{\mathrm{u}}^{k})^2, & \hat{E}[\widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-j}] > \hat{f}_{\mathrm{u}}^{k} \\ \hat{E}[(\widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-j})^2], & \hat{f}_{\mathrm{u}}^{k} - 255 \le \hat{E}[\widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-j}] \le \hat{f}_{\mathrm{u}}^{k}, \end{cases} \tag{4–4}$$

In H.264, the accuracy of motion compensation is in units of one quarter of the distance between luma samples. [2] The prediction values at half-sample positions are obtained by applying a one-dimensional 6-tap Finite Impulse Response (FIR) filter horizontally and vertically. The prediction values at quarter-sample positions are generated by averaging samples at integer- and half-sample positions [20]. In such a case, some variables in the pixel-level distortion estimation now include a fractional MV and those variables should be re-estimated. As a result, $E[\widetilde{\zeta}^{k-1}_{\mathrm{u}+\check{\mathrm{mv}}^k_{\mathrm{u}}}]$, $E[(\widetilde{\zeta}^{k-1}_{\mathrm{u}+\check{\mathrm{mv}}^k_{\mathrm{u}}})^2]$, $E[\widetilde{\zeta}^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}} + \widetilde{\Delta}^k_{\mathrm{u}}\{\bar{r}, \bar{m}\}]$ and $E[(\widetilde{\zeta}^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}} + \widetilde{\Delta}^k_{\mathrm{u}}\{\bar{r}, \bar{m}\})^2]$ in (4–1) and (4–2) should be estimated based on the fractional MV. Since $\hat{E}[\widetilde{\zeta}^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}} + \widetilde{\Delta}^k_{\mathrm{u}}\{\bar{r}, \bar{m}\}]$ can be calculated by $\hat{E}[\widetilde{\zeta}^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}}]$ as in (4–3) and $\hat{E}[(\widetilde{\zeta}^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}} + \widetilde{\Delta}^k_{\mathrm{u}}\{\bar{r}, \bar{m}\})^2]$ can be calculated by $\hat{E}[(\widetilde{\zeta}^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}})^2]$ as in (4–4), we only need to determine the first moment and second moment of $\widetilde{\zeta}^{k-1}_{\mathrm{u}+\check{\mathrm{mv}}^k_{\mathrm{u}}}$ and $\widetilde{\zeta}^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}}$ from their neighboring integer pixel positions.

Take $\widetilde{\zeta}^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}}$ for example. Denote $\mathbf{v}^{k-j} = \mathbf{u} + \mathbf{mv}^k_{\mathrm{u}}$ and $\mathbf{v}$ is in a subpixel position in the $k-j$-th frame. All neighboring pixels in the integer position, used to interpolate the reconstructed pixel value at $\mathbf{v}$, are denoted by $\mathbf{u}_i$ and with a weight w_i, $i \in 1, 2, ..., N$, where $N = 6$ for the half-sample interpolation, and $N = 2$ for the quarter-sample interpolation in H.264. Therefore, the interpolated reconstructed pixel value at the encoder is

$$\hat{f}^{k-j}_{\mathrm{v}} = \sum_{i=1}^{N} w_i \cdot \hat{f}^{k-j}_{\mathrm{u}_i}, \tag{4–5}$$

and the interpolated reconstructed pixel value at the decoder is

$$\widetilde{f}^{k-j}_{\mathrm{v}} = \sum_{i=1}^{N} w_i \cdot \widetilde{f}^{k-j}_{\mathrm{u}_i}. \tag{4–6}$$

[2] Note that considering the chroma distortion does not always improve the R-D performance but induces more complexity. Therefore, we only consider luma components in this chapter.

As a result, we have

$$E[\widetilde{\zeta}_{\mathrm{v}}^{k-j}] = E[\sum_{i=1}^{N} w_i \cdot (\hat{f}_{\mathrm{u}_i}^{k-j} - \widetilde{f}_{\mathrm{u}_i}^{k-j})] = \sum_{i=1}^{N} (w_i \cdot E[\widetilde{\zeta}_{\mathrm{u}_i}^{k-j}]), \tag{4–7}$$

and

$$\begin{aligned} E[(\widetilde{\zeta}_{\mathrm{v}}^{k-j})^2] &= E[(\sum_{i=1}^{N} w_i \cdot \hat{f}_{\mathrm{u}_i}^{k-j} - \sum_{i=1}^{N} w_i \cdot \widetilde{f}_{\mathrm{u}_i}^{k-j})^2] \\ &= E\{[\sum_{i=1}^{N} w_i \cdot (\hat{f}_{\mathrm{u}_i}^{k-j} - \widetilde{f}_{\mathrm{u}_i}^{k-j})]^2\} \\ &= E[(\sum_{i=1}^{N} w_i \cdot \widetilde{\zeta}_{\mathrm{u}_i}^{k-j})^2]. \end{aligned} \tag{4–8}$$

Since $\hat{E}[\widetilde{\zeta}_{\mathrm{u}_i}^{k-j}]$ have been calculated by the RMPC algorithm, $\hat{E}[\widetilde{\zeta}_{\mathrm{v}}^{k-j}]$ can be very easily calculated by (4–7). However, calculating $\hat{E}[(\widetilde{\zeta}_{\mathrm{v}}^{k-j})^2]$ is not straightforward since $E[(\widetilde{\zeta}_{\mathrm{v}}^{k-j})^2]$ in (4–8) is in fact the second moment of a weighted sum of random variables.

4.2.2 A New Theorem for Calculating the Second Moment of a Weighted Sum of Correlated Random Variables

The Moment Generating Function (MGF) can be used to calculate the second moment for random variables [25]. However, to estimate the second moment of a weighted sum of random variables, the traditional moment generating function usually requires knowing their probability distribution and assuming they are independent. However, most random variables involved in the averaging operations in a video codec are not independent and their probability distributions are unknown. Therefore, some approximations, such as the Cauchy-Schwarz upper bound approximation [32] or the correlation coefficient model approximation [17], are usually adopted to approximate the second moment of a complicated random variable. However, those approximations require very high complexity. For example, for each subpixel, with the N-tap filter interpolation, the Cauchy-Schwarz upper bound approximation requires N integer multiplications for calculating the second moment terms, $N(N-1)/2$ floating-point multiplications and $N(N-1)/2$ square root operations for calculating the cross-correlation

terms, and $N(N-1)/2+N-1$ additions and 1 shift for calculating the estimated distortion. The correlation coefficient model requires an additional $N(N-1)/2$ exponential operations and $N(N-1)/2$ floating-point multiplications when compared to the Cauchy-Schwarz upper bound approximation.

In a wireless video communication system, the computational capability of the real-time encoder is usually very limited, and floating-point processing is undesirable especially for mobile devices. Therefore, the question is how to design a new algorithm to accurately calculate the second moment in (4–8) via only integer multiplication, integer addition, and shifts.

we can design a low-complexity and high-accuracy algorithm to extend the RMPC algorithm through the consideration of the following theorem.

Theorem 4.1. *For any N correlated random variables $\{X_1, X_2, ..., X_N\}$ and $w_i \in \Re$, $i \in \{1, 2, ..., N\}$, the second moment of the weighted sum of these random variables is given by*

$$E[(\sum_{i=1}^{N} w_i \cdot X_i)^2] = \sum_{i=1}^{N} w_i \cdot \sum_{j=1}^{N}[w_j \cdot E(X_j^2)] - \sum_{k=1}^{N-1}\sum_{l=k+1}^{N}[w_k \cdot w_l \cdot E(X_k - X_l)^2] \tag{4–9}$$

Theorem 4.1 is proved in Appendix C

In H.264, most averaging operations, e.g., interpolation, deblocking, and bi-prediction, are the special cases of Theorem 4.1 in that $\sum_{i=1}^{N} w_i = 1$. In (4–9), $\sum_{j=1}^{N}[w_j \cdot E(X_j^2)]$ is the weighted sum of $E(X_j^2)$, which has been estimated by the RMPC algorithm, and the only unknown is $\sum_{k=1}^{N-1}\sum_{l=k+1}^{N}[w_k \cdot w_l \cdot E(X_k - X_l)^2]$. However, we will see that this unknown can be assumed to be negligible for the purposes of mode decision.

4.2.3 The Extended RMPC Algorithm for Mode Decision

Replacing X_k and X_l in (4–9) by $\widetilde{\zeta}_{\mathrm{u}_i}^k$ and $\widetilde{\zeta}_{\mathrm{u}_j}^k$, we obtain

$$\begin{aligned}\widetilde{\zeta}_{\mathrm{u}_i}^k - \widetilde{\zeta}_{\mathrm{u}_j}^k &= \hat{f}_{\mathrm{u}_i}^k - \widetilde{f}_{\mathrm{u}_i}^k - (\hat{f}_{\mathrm{u}_j}^k - \widetilde{f}_{\mathrm{u}_j}^k) \\ &= (\hat{f}_{\mathrm{u}_i}^k - \hat{f}_{\mathrm{u}_j}^k) - (\widetilde{f}_{\mathrm{u}_i}^k - \widetilde{f}_{\mathrm{u}_j}^k)\end{aligned} \tag{4–10}$$

In (4–10), both $\hat{f}^k_{\mathrm{u}_i} - \hat{f}^k_{\mathrm{u}_j}$ and $\widetilde{f}^k_{\mathrm{u}_i} - \widetilde{f}^k_{\mathrm{u}_j}$ depend on the spatial correlation of the reconstructed pixel values in position $\mathbf{u}_i$ and $\mathbf{u}_j$. When $\mathbf{u}_i$ and $\mathbf{u}_j$ are located in the same neighborhood, they are very likely to be transmitted in the same packet. Therefore, the difference between $\hat{f}^k_{\mathrm{u}_i} - \hat{f}^k_{\mathrm{u}_j}$ and $\widetilde{f}^k_{\mathrm{u}_i} - \widetilde{f}^k_{\mathrm{u}_j}$ is very small and hence $E[(\widetilde{\zeta}^k_{\mathrm{u}_i} - \widetilde{\zeta}^k_{\mathrm{u}_j})^2]$ is much smaller than $E[(\widetilde{\zeta}^k_{\mathrm{u}_i})^2]$ and $E[(\widetilde{\zeta}^k_{\mathrm{u}_j})^2]$. On the other hand, distortion is estimated for one MB or one sub-MB as in (3–26) for mode decision. When the cardinality $|\mathcal{V}^k_l|$ is large, $\sum_{\mathrm{v}\in\mathcal{V}^k_l}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}[w_i \cdot w_j \cdot E(\widetilde{\zeta}^k_{\mathrm{u}_i} - \widetilde{\zeta}^k_{\mathrm{u}_i})^2]$ converges to a constant for all modes with high probability due to the summation over all pixels in that MB. For simplicity, we will call it "negligible term" in the following sections. Therefore, in (4–9) only the first term in the right-hand side need to be calculated. Since $\sum_{i=1}^{N} w_i = 1$, we estimate $E[(\widetilde{\zeta}^k_{\mathrm{v}})^2]$ for mode decision by

$$\hat{E}[(\widetilde{\zeta}^k_{\mathrm{v}})^2] = \sum_{i=1}^{N}[w_i \cdot \hat{E}(\widetilde{\zeta}^k_{\mathrm{u}_i})^2]. \tag{4–11}$$

With (4–11), the complexity for estimating the distortion of each subpixel, with the N-tap filter interpolation, is dramatically reduced to only N integer multiplications, $N-1$ additions, and 1 shift. Here, we propose the following algorithm to extend the RMPC algorithm for mode decision.

Algorithm 3. *Rate distortion optimized mode decision for an MB in the k-th frame ($k >= 1$).*

*1) **Input**: QP, PEP.*

2) Initialization of $\hat{E}[\widetilde{\zeta}^0_{\mathrm{u}}]$ and $\hat{E}[(\widetilde{\zeta}^0_{\mathrm{u}})^2]$ for all pixel $\mathbf{u}$.

3) Loop for all available modes for each MB.

estimate $E[\widetilde{\zeta}^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}}]$ via (4–7) and $E[(\widetilde{\zeta}^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}})^2]$ via (4–11) for all pixels in the MB,

estimate $E[\widetilde{\zeta}^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}} + \widetilde{\Delta}^k_{\mathrm{u}}\{\bar{r}, \bar{m}\}]$ via (4–3) for all pixels in the MB,

estimate $E[(\widetilde{\zeta}^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}} + \widetilde{\Delta}^k_{\mathrm{u}}\{\bar{r}, \bar{m}\})^2]$ via (4–4) for all pixels in the MB,

estimate $E[\widetilde{\zeta}^k_{\mathrm{u}}]$ via (4–1) and $E[(\widetilde{\zeta}^k_{\mathrm{u}})^2]$ via (4–2) for all pixels in the MB,

estimate D^k_{u} via (3–24) for all pixels in the MB,

estimate R-D cost for the MB via (3–26)*,*

End

Via (3–27)*, select the best mode with minimum R-D cost for the MB.*

4) ***Output****: the best mode for the MB.*

In this chapter, Algorithm 3 is referred to as ERPMC. Note that if an MV packet is lost, the ERMPC algorithm conceals the MV with integer accuracy to reduce both the concealment complexity and estimation complexity. Therefore, estimating $E[\widetilde{\zeta}^{k-1}_{\mathrm{u}+\check{\mathbf{mv}}^k_{\mathrm{u}}}]$ and $E[(\widetilde{\zeta}^{k-1}_{\mathrm{u}+\check{\mathbf{mv}}^k_{\mathrm{u}}})^2]$ in (4–1) and (4–2) does not require (4–11) and saves computational cost[3] .

4.2.4 Merits and Limitations of ERMPC Algorithm

4.2.4.1 Merits

Since both the Cauchy-Schwarz upper bound approximation [32] and the correlation coefficient model approximation [17] induce floating-point multiplications, round-off error is unavoidable in those algorithms. The algorithm by Yang et al. [17] needs extra complexity to mitigate the effect of round-off error in their distortion estimation algorithm. In contrast, one of the merits of Theorem 4.1 is that it only needs integer multiplications and integer additions. Assuming w_i (and $w_i \cdot w_j$) can be scaled up to be an integer without any round-off error, we may compare all R-D costs by scaling them for all modes. Therefore, round-off error is avoided in the ERMPC algorithm.

In Ref. [8], the authors prove that a low-pass interpolation filter will decrease the frame-level propagated error under some assumptions. In fact, it is easy to prove that when $\sum_{i=1}^{N} w_i = 1$ and $|\mathcal{V}_l^k|$ is large, the negligible term is larger than or equal to zero. Even in the MB-level, the negligible term is larger than or equal to zero with very high

[3] Note that $\check{\mathbf{mv}}^k_{\mathrm{u}}$ denotes the concealed motion vector for pixel $\mathbf{u}^k$, under the case that $\mathbf{mv}^k_{\mathrm{u}}$ is received with error.

probability. From (4–9), we see that the block-level distortion decreases, with very high probability, after the interpolation filter.

One additional benefit of (4–9) is to guide the design of the interpolation filter. Traditional interpolation filter design aims to minimize the prediction error. With (4–9), we may design an interpolation filter by maximizing $\sum_{k=1}^{N}\sum_{l=k+1}^{N}[w_k \cdot w_l \cdot E(X_k - X_l)^2]$ under the constraint of $\sum_{j=1}^{N}[w_j \cdot E(X_j^2)]$.

4.2.4.2 Limitations

In Algorithm 3, the second moment of propagated error $E[(\widetilde{\zeta}^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}})^2]$ is estimated by neglecting the negligible term to reduce the complexity. A more accurate alternative method is to estimate $E(\widetilde{\zeta}^k_{\mathrm{u}_i} - \widetilde{\zeta}^k_{\mathrm{u}_i})^2]$ recursively by storing the value in memory. This will be considered in our future work.

4.3 Experimental Results

In this section, we compare the R-D performance and subjective performance of the ERMPC algorithm with that of the LLN algorithm for mode decision in H.264. We also compare ERMPC with RMPC and ROPE by using the nearest neighbor to approximate the reference pixel pointed by a fractional MV. To compare all algorithms under multi-reference picture motion compensated prediction, we also enhance the original ROPE algorithm [4] with multi-reference capability.

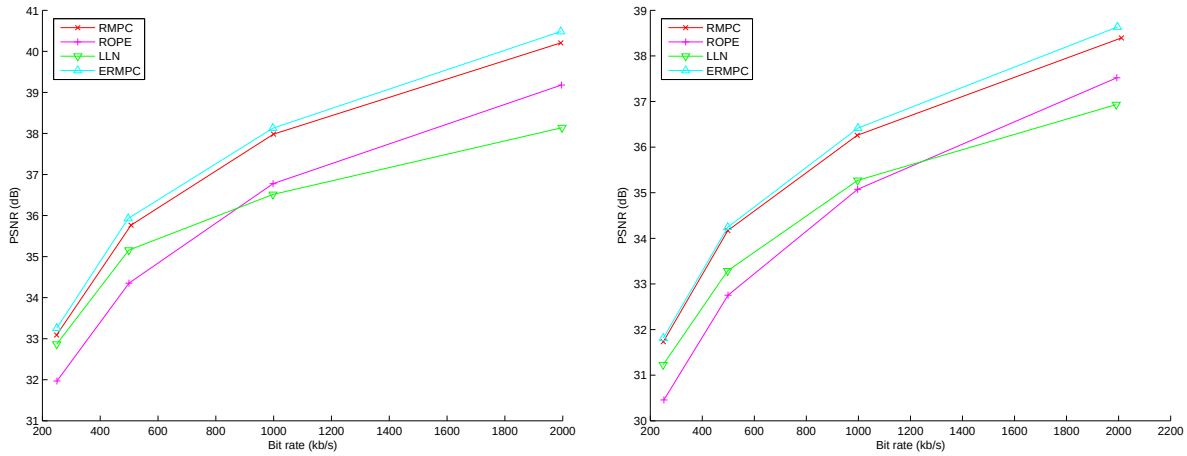
4.3.1 Experiment Setup

The JM16.0 encoder and decoder is used in the experiments. The high profile as defined in the H.264 specification Annex A [22] is used. All the tested video sequences are in CIF resolution at 30fps. Each coded video sequence is tested under different PEP settings from 0.5% to 5%. Each video sequence is coded for its first 100 frames with 3 slices per frame. The error concealment method used for all algorithms is to copy the pixel value in the same position of the previous frame. The first frame is assumed to be correctly received.

The encoder setting is given as below. Constrained intra prediction is enabled; the number of reference frames is 3; B slices are not included; only 4x4 transform is used; CABAC is enabled for entropy coding; in the LLN algorithm, the number of simulated decoders is 30.

4.3.2 R-D Performance

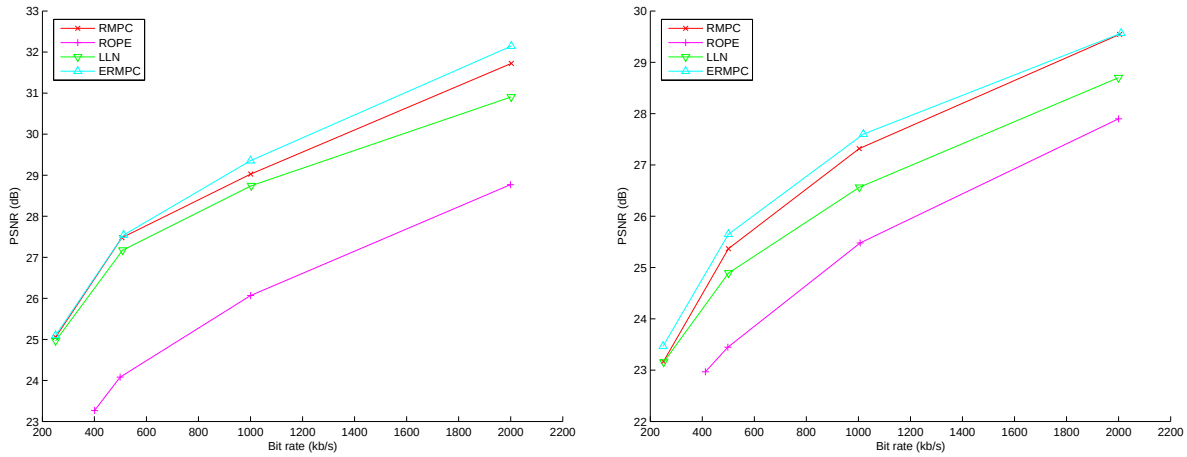
Due to space limitations, we only show the plots of PSNR vs. bit rate for video sequences 'foreman' and 'mobile' under $PEP = 2\%$ and $PEP = 5\%$. Figs. 4-1 and 4-2 show PSNR vs. bit rate for 'foreman' and 'mobile', respectively. The experimental results show that ERMPC achieves the best R-D performance; RMPC achieves the second best R-D performance; ROPE achieves better performance than LLN in some cases such as at high rate in Fig. 4-1, but worse performance than LLN in other cases such as in Fig. 4-2 and at the low rate in Fig. 4-1.



(a) (b)

Figure 4-1. PSNR vs. bit rate for 'foreman': (a) PEP=0.5%, (b) PEP=2%.

It is interesting to see that for some sequences and channel conditions, ERMPC achieves a notable PSNR gain over RMPC. This is, for example, evident with 'mobile' and 'foreman'. For some other cases, however, ERMPC only achieves a marginal PSNR gain over RMPC (e.g., 'coastguard' and 'football'). From the analysis in Section 4.2.1, we know that the only difference between RMPC and ERMPC is the estimate of the error from the reference pixel, i.e., propagated error, under the condition that there



(a) (b)

Figure 4-2. PSNR vs. bit rate for ‘mobile’: (a) PEP=0.5%, (b) PEP=2%.

is no newly induced error in the current pixel. Therefore, the performance gain of ERMPC over RMPC only comes from inter modes, since they both use exactly the same estimates for intra modes. Thus, a higher percentage of intra modes in ‘coastguard’ and ‘football’ may result in a marginal PSNR gain of ERMPC over RMPC.

For most sequences and channel conditions, we observe that the higher the bit rate for encoding, the more the PSNR gain of ERMPC over RMPC, such as in Fig. 4-1 and Fig. 4-2(a). In (3–24), the end-to-end distortion consists of both quantization distortion and transmission distortion. The ERMPC algorithm gives a more accurate estimation of propagated error in transmission distortion than the RMPC algorithm. When the bit rate for source encoding is very low, with rate control the controlled QP is large, and hence the quantization distortion becomes the dominant factor in the end-to-end distortion. Therefore, the PSNR gain of ERMPC over RMPC is marginal. On the contrary, when the bit rate for source encoding is high, the transmission distortion becomes the dominant part in the end-to-end distortion. Therefore, the PSNR gain of ERMPC over RMPC is notable. However, this is not always true as observed in Fig. 4-2(b). In JM16.0, the Lagrange multiplier in (3–26) is a function of QP. A higher bit rate or smaller QP also causes a smaller Lagrange multiplier. Therefore, the rate cost in (3–26) becomes smaller, which may produce a higher percentage of intra modes. In such a case, the

PSNR gain of ERMPC over RMPC decreases when the bit rate becomes higher. As a result, different sequences give different results depending on whether more intra modes are selected when bit rate increases.

LLN has poorer R-D performance than ERMPC. This may be because 30 simulated decoders are still not enough to achieve a reliable distortion estimate although LLN with 30 simulated decoders already incurs much higher complexity than ERMPC. Since the original ROPE does not support the interpolation filtering operation and its extensions [17, 32] induce many floating-point operations and round-off errors, we only use the same nearest neighbor approximation to show how its R-D performance differs from ERMPC, RMPC, and LLN. We see that such an extension is valid for some sequences, such as 'foreman'. However, this approximation gives poor R-D performance for some other sequences, such as 'mobile'. Therefore, RMPC is easier to be extended than ROPE since the nearest neighbor approximation for RMPC in all sequences achieves good performance.

Table 4-1 shows the average PSNR gain (in dB) of ERMPC over RPMC, LLN, and ROPE for different video sequences and different PEP. The average PSNR gain is obtained by the method in Ref. [30], which measures average distance (in PSNR) between two R-D curves. From Table 4-1, we see that ERMPC achieves an average PSNR gain of 0.25dB over RMPC for the sequence 'mobile' under $PEP = 2\%$; it achieves an average PSNR gain of 1.34dB over LLN for the 'foreman' sequence under $PEP = 1\%$; and it achieves an average PSNR gain of 3.18dB over ROPE for the 'mobile' sequence under $PEP = 0.5\%$.

4.3.3 subjective Performance

Since PSNR could be less meaningful for error concealment, a much more important performance criterion is the subjective performance, which directly relates to the degree of user's satisfaction. Fig. 4-3 shows the subjective quality of the 84-th frame and the 99-th frame of 'foreman' sequence under a PER of 1% and a bit rate of

Table 4-1. Average PSNR gain (in dB) of ERMPC over RMPC, LLN and ROPE

Sequence	PEP	ERMPC vs. RMPC	ERMPC vs. LLN	ERMPC vs. ROPE
coastguard	5%	0.09	0.32	0.58
	2%	0.08	0.36	0.46
	1%	0.08	0.46	0.52
	0.5%	0.06	0.37	0.62
football	5%	0.01	0.28	0.47
	2%	0.01	0.39	0.25
	1%	0.01	0.36	0.27
	0.5%	0.03	0.26	0.33
foreman	5%	0.08	0.64	1.59
	2%	0.13	1.07	1.37
	1%	0.21	**1.34**	1.41
	0.5%	0.17	1.24	1.42
mobile	5%	0.20	0.50	1.11
	2%	**0.25**	0.82	1.89
	1%	0.21	0.56	2.79
	0.5%	0.21	0.54	**3.18**

250kbps. From Fig. 4-3, we see the similar performance result as in Section 4.3.2. That is, ERMPC achieves the best performance.

4.3.4 Discussion

4.3.4.1 Effect of clipping noise on the mode decision

The experiments show that since it does not consider clipping noise, ROPE over-estimates the end-to-end distortion for inter modes. Hence, ROPE tends to select intra modes more often than ERMPC, RMPC, and LLN, which leads to higher encoding bit rates. To verify this conjecture, we tested all sequences under the same Quantization Parameter (QP) settings from 20 to 32 without rate control. We observed that the ROPE algorithm always produced a higher bit rate than other schemes as shown in Fig. 4-4 and Fig. 4-5.

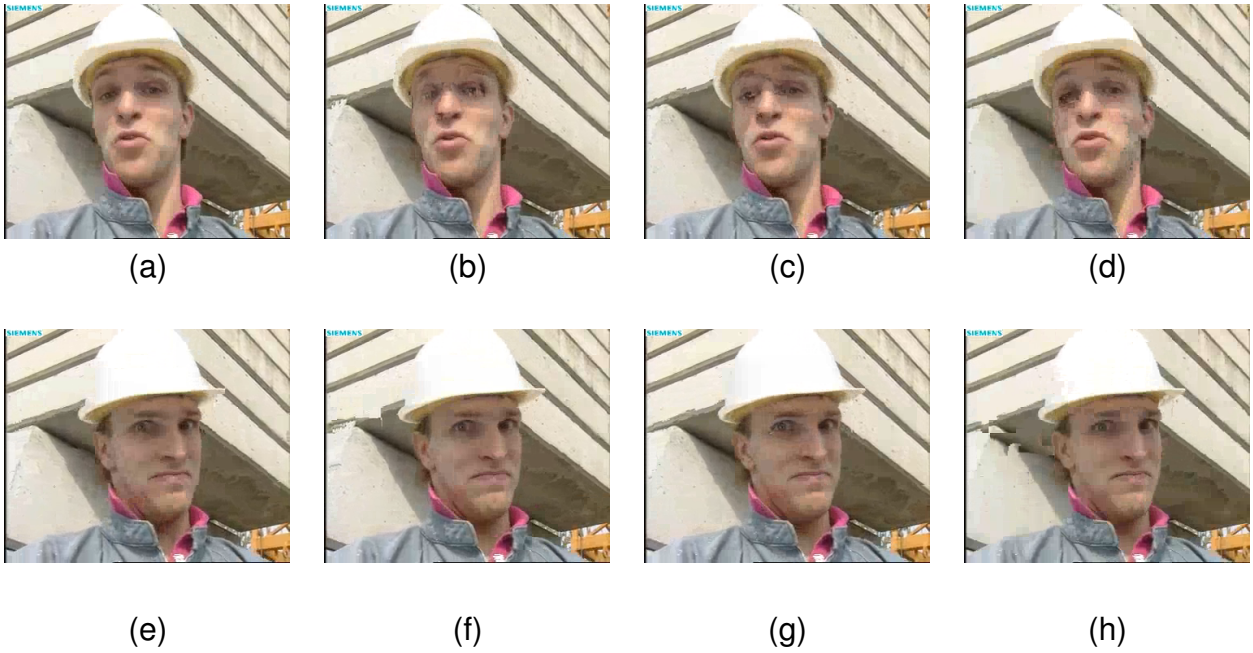


Figure 4-3. (a) ERMPC at the 84-th frame, (b) RMPC at the 84-th frame, (c) LLN at the 84-th frame, (d) ROPE at the 84-th frame, (e) ERMPC at the 99-th frame, (f) RMPC at the 99-th, (g) LLN at the 99-th frame, (h) ROPE at the 99-th frame.

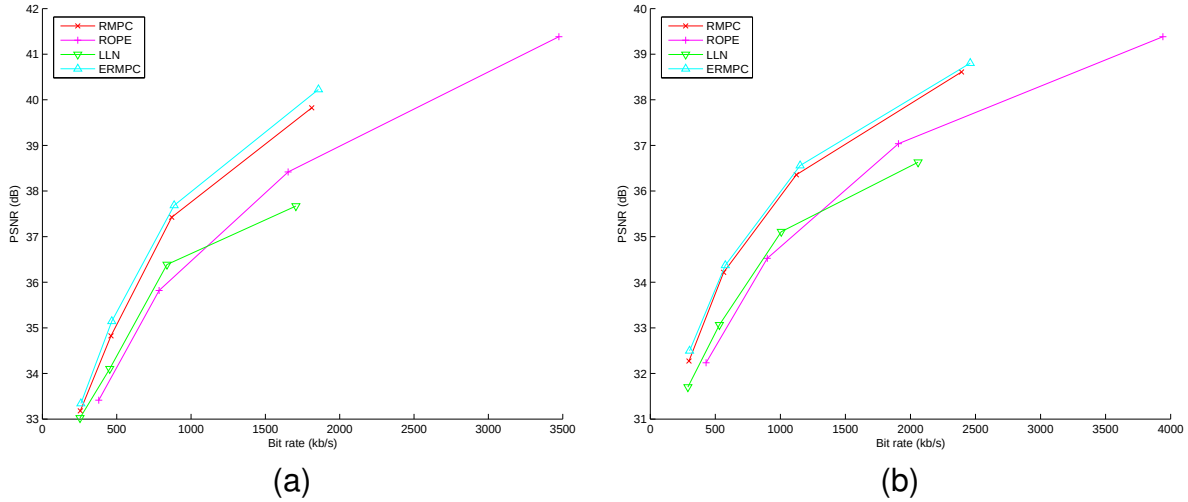
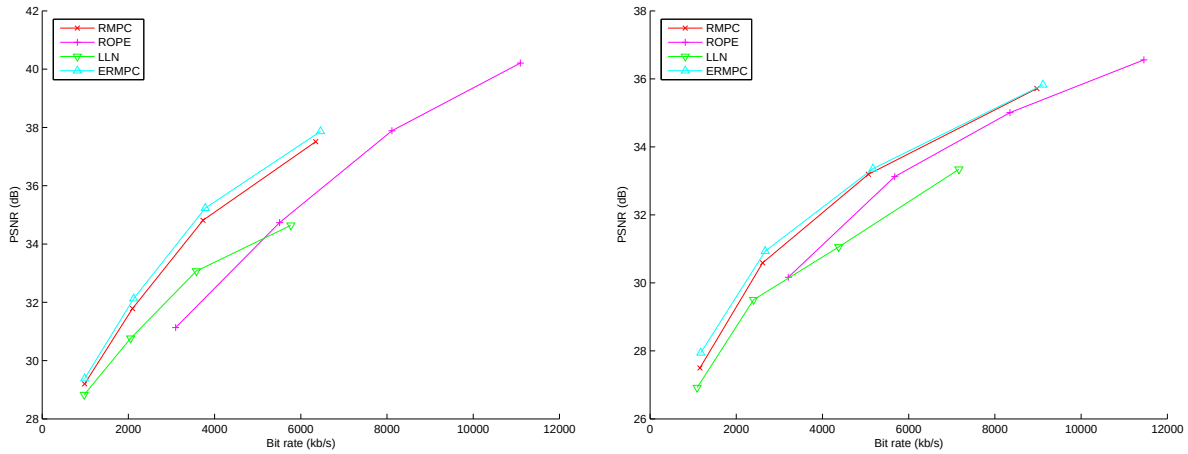


Figure 4-4. PSNR vs. bit rate for 'foreman': (a) PEP=0.5%, (b) PEP=2%.

4.3.4.2 Effect of transmission errors on mode decision

Compared to the regular RDO process in JM16.0 without considering the transmission error, ERMPC/RMPC/LLN/ROPE algorithms show three distinctions from it. 1) The number of intra MBs increases since transmission error is accounted for in the



(a) (b)

Figure 4-5. PSNR vs. bit rate for 'mobile': (a) PEP=0.5%, (b) PEP=2%.

mode decision; 2) The number of MBs with skip mode increases since the transmission error will increase the transmission distortion in all other modes except the skip mode; 3) if we allow the first frame to be erroneous, the second frame will have many intra MBs since the propagated error from the first frame is much higher than for other frames. This is because only the value of 128 can be used to conceal the reconstructed pixel value if the first frame is lost, while if other frames are lost, the collocated pixel in the previous frame can be used to conceal the reconstructed pixel value. Therefore, the transmission error in the first frame gives much higher propagated error than in the other frames.

CHAPTER 5
RATE-DISTORTION OPTIMIZED CROSS-LAYER RATE CONTROL IN WIRELESS VIDEO COMMUNICATION

In this chapter, we derive more accurate source bit rate model and quantization distortion model than existing parametric models. We also improve the performance bound of channel coding with convolutional codes and a Viterbi decoder, and derive its performance under Rayleigh block fading channel. Given the instantaneous channel condition, i.e. SNR and bandwidth, we design a rate-distortion optimized cross-layer rate control (CLRC) algorithm by jointly choosing quantization step size and channel coding rate.

5.1 An Literature Review on Rate Distortion Models in Wireless Video Communication Systems

Under the prevalence of 3G/4G network and smart phones nowadays, real-time mobile video applications, e.g., videophone calls, are becoming more and more popular. However, transmitting video over mobile phone with good quality is particularly challenging since the mobile channels subject to multipath fading, and therefore the channel condition changes from frame to frame. Given the instantaneous channel condition, e.g., signal noise ratio (SNR) and bandwidth, the minimum end-to-end distortion can be achieved by optimally allocating the transmission bit rate between source bit rate and redundant bit rate. In a practical wireless video communication system, this can be achieved by jointly control the source encoding parameters, e.g, quantization step size, in the video encoder, and channel encoding parameters, e.g., channel coding rate, in the channel encoder. Since both the video statistics and channel condition vary with time, we need to dynamically control those parameters for each frame in real-time video encoding and packet transmission. Therefore, we need to estimate the bit rate and distortion for each possible combination of parameters before encoding each frame. As a result, accurate bit rate model and distortion model will be very helpful to achieve the minimum end-to-end distortion with low complexity.

There are many works trying to address this problem during these years. While most of them derive the end-to-end distortion as functions of bit rate and packet error rate [8, 10], others use operational rate-distortion (R-D) functions [34]. The analytical models are more desirable since it is very difficult for the video encoder to get all operational functions for different video statistics and channel conditions before real encoding. However, the existing analytical models are still not accurate enough to accommodate the time-varying channel condition. On the other hand, to get tractable formulae in those analytical models [8, 10], authors all assume that block codes, i.e., Reed-Solomon codes, are adopted as the forward error correction (FEC) scheme. Based on that FEC scheme, the distortion is derived as a function of channel coding rate and bit error rate. However, this assumption has two limitations: 1) most up-to-date video communication systems use convolutional codes or more advanced codes, e.g., turbo codes, for physical layer channel coding due to their flexible choice of channel coding rate without the change the channel coding structure; 2) in the cross-layer optimization problem, the selection of source bit rate and redundant bit rate based on a given bit error rate is suboptimal, while the optimal solution can be achieved by jointly choosing them based on the given instantaneous channel condition, e.g., SNR and channel bandwidth. In this chapter, we aim to solve the cross-layer optimization problem by deriving more accurate bit rate model and end-to-end distortion model, which consists of two parts, that is, quantization distortion model and transmission distortion model.

Plenty of bit rate models have been developed in existing literature. Most of the existing works derive bit rate as a function of video statistics and quantization step size [35–38], while others model bit rate as a function of video statistics and other parameters [39]. In general, these models come from either experimental observation [37, 39–41] or parametric modeling [38, 42, 43]. However, both of them have some limitations. The experimental modeling usually induces some model

parameters which can only be estimated from previous frames. Therefore, the model accuracy depends not only on the statistics and coding parameters but also on the estimation accuracy of those model parameters. However, in theory, the instantaneous frame bit rate should be independent of previous frames given instantaneous video frame statistics and coding parameters. In addition, the estimation error of those model parameters may have a significant impact on the model accuracy, which can be observed in the H.264/AVC JM reference software [33] and will be explained in detail in the experimental section of this chapter. On the other hand, the parametric modeling has the following two limitations: 1) the assumed residual probability distribution, e.g., Laplacian distribution, may deviate significantly from the true histogram; 2) the implicit assumption of all transform coefficients being identically distributed is not valid if run-length coding is conducted before the entropy coding as in most practical encoders. Since the model-selection problem may often be more important than having an optimized algorithm [44], simply applying these parametric models to a real encoder may result in poor R-D performance. In this chapter, we improve the bit rate model by modeling the component of **run-level mapping plus entropy coding** as the process of choosing different codebooks for different quantized transform coefficients. We also compensate the mismatch between the true histogram and the assumed Laplacian distribution in the parametric model by utilizing the estimation deviation of previous frames. Experimental results show that our method achieves a more accurate estimate of bit rate compared to existing models.

Quantization distortion is caused by quantization error under lossy source coding and it has been extensively explored since the seminal work of Shannon's rate distortion theory first proposed in Ref. [1] and later proved in Ref. [2]. The quantization distortion are studied either as a function of bit rate and the source probability distribution, e.g., the classical R-D function for Gaussian source [28, 45], or as a function of level number and the source probability distribution given a certain quantizer, e.g., uniform scaler

quantizer for Gaussian source [46]. In the case of minimum mean-square digitization of memoryless Gaussian sources, quantizers with uniformly spaced levels have entropies that exceed the rate-distortion function by approximately 0.25 bits/sample at relatively high rates [47]. In Ref. [48], the performance of optimum quantizers for a wide class of memoryless non-Gaussian sources is investigated, and it is shown that uniform threshold quantizers perform as effectively as optimum quantizers. For this reason, the uniform quantizer is ususally adopted in a practical video encoder, e.g. H.264 [22]. For a uniform quantizer, the quantization distortion have been derived as a function of quantization step size (or corresponding operation point) and video frame statistics either from experimental observation [37, 39] or by parametric modeling [38, 42]. Although the parametric modeling has achieved quite accurate result, it can be further improved due to the source distribution model inaccuracy. In this chapter, we improve the estimation accuracy of quantization distortion by utilizing the similar method in bit rate model. Experimental results show that our quantization distortion model is more accurate than existing models.

Transmission distortion is caused by transmission error under error-prone channels. Predicting transmission distortion at the transmitter poses a great challenge due to the spatio-temporal correlation inside the input video sequence, the nonlinearity of video codec, and varying PEP in time-varying channels. The existing transmission distortion models can be categorized into the following three classes: 1) pixel-level or block-level models (applied to prediction mode selection) [4–6]; 2) frame-level or packet-level or slice-level models (applied to cross-layer encoding rate control) [7–11]; 3) GOP-level or sequence-level models (applied to packet scheduling) [12–16]. Although the existing transmission distortion models work at different levels, they share some common properties, which come from the inherent characteristics of wireless video communication system, that is, spatio-temporal correlation, nonlinear codec and time-varying channel. However, none of those works analyzed the effect of

non-linear clipping noise on the transmission distortion, and therefore cannot provide accurate transmission distortion estimation. In Chapter 2, we analytically derive, for the first time, the transmission distortion formula as a closed-form function of packet error probability (PEP), video frame statistics, and system parameters; and then in Chapter 3, we design the RMPC algorithm to predict the transmission distortion with low complexity and high accuracy. In this chapter, we will further derive PEP and transmission distortion as functions of SNR, transmission rate, and channel coding rate for cross-layer optimization.

Channel coding can be considered as the embedding of signal constellation points in a higher dimensional signaling space than is needed for communications. By mapping to a higher dimensional space, the distance between points increases, which provides better error correction and detection performance [18]. In general the performance of soft-decision decoding is about 2-3dB better than hard-decision decoding [18]. Since convolutional decoders have efficient soft-decision decoding algorithms, such as Viterbi algorithm [49], we choose convolutional codes for physical layer channel coding in this chapter [1] . In addition, Rate-compatible punctured convolutional (RCPC) codes can adaptively change the coding rate without changing the encoder structure, which makes convolutional codes an appropriate method in real-time video communication over wireless fading channels. In this chapter we improve the performance bound of convolutional codes by adding a threshold for low SNR case, and extend it to support a more flexible SNR threshold for transmitters with channel estimation. For transmitters without channel estimation, we also derive the expected PEP as a simple function of convolutional encoder structure and channel condition under Rayleigh block fading channel.

[1] Our algorithm can also be used for other channel codes, e.g. block codes, Turbo codes, and LDPC codes, given their performance for different coding rates.

Given the bit rate function, quantization distortion function and transmission distortion function, minimizing end-to-end distortion becomes an optimization problem under the transmission bit rate constraint. In this chapter, we also apply our bit rate model, quantization distortion model and transmission distortion model to cross-layer rate control with rate-distortion optimization (RDO). Due to the discrete characteristics and the possibility of non-convexity of distortion function [50], the traditional Lagrange multiplier solution for continuous convex function optimization is infeasible in a video communication system. The discrete version of Lagrangian optimization is first introduced in Ref. [51], and then first used in a source coding application in Ref. [50]. Due to its simplicity and effectiveness, this optimization method is de facto adopted by the practical video codec, e.g., H.264 reference code JM [33]. In this chapter, we will use the same method to solve our optimization problem.

The rest of this chapter is organized as follows. In Section 5.2, we formulate the cross-layer optimization problem. In Section 5.3, we derive our bit rate model, quantization distortion model, and transmission distortion model. In Section 5.4, we propose a practical cross-layer rate control algorithm to achieve minimum end-to-end distortion under the given SNR and channel bandwidth. Section 5.5 shows the experimental results, which demonstrates both the higher accuracy of our models and the better performance of our algorithm over existing algorithms.

5.2 Problem Formulation

Fig. 2-1 shows the structure of a typical wireless video communication system. It consists of an encoder, two channels and a decoder where residual packets and MV packets are transmitted over their respective channels. Note that in this system, both residual channel and MV channel are application-layer channels. Fig. 5-1 shows the channel details for these two channels.

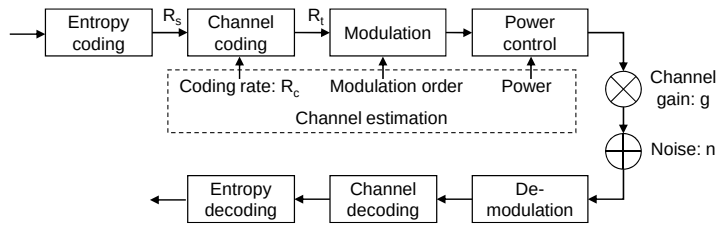


Figure 5-1. Channel model.

The general RDO problem in a wireless video communication system can be formulated as

$$\begin{aligned} min \quad & D_{ETE}^k \\ s.t. \quad & R_t^k \leq R_{con}^k, \end{aligned} \tag{5–1}$$

where D_{ETE}^k is the end-to-end distortion of the k-th frame, R_t^k is the transmitted bit rate of the k-th frame, R_{con}^k (depend on the channel condition) is the bit rate constraint of the k-th frame.

From the definition, we have

$$D_{ETE}^k \triangleq E[\frac{1}{|\mathcal{V}^k|} \sum_{\mathrm{u} \in \mathcal{V}^k} (f_{\mathrm{u}}^k - \widetilde{f}_{\mathrm{u}}^k)^2], \tag{5–2}$$

where $\mathcal{V}^k$ is the set of pixels in the k-th frame; f_{u}^k is the original pixel value for pixel $\mathbf{u}$ in the k-th frame; $\widetilde{f}_{\mathrm{u}}^k$ is the reconstructed pixel value for the corresponding pixel at the decoder;

Define quantization error as $f_{\mathrm{u}}^k - \hat{f}_{\mathrm{u}}^k$ and transmission error as $\hat{f}_{\mathrm{u}}^k - \widetilde{f}_{\mathrm{u}}^k$, where $\hat{f}_{\mathrm{u}}^k$ is the reconstructed pixel value for pixel $\mathbf{u}$ in the k-th frame at the encoder. While $f_{\mathrm{u}}^k - \hat{f}_{\mathrm{u}}^k$ depends only on the quantization parameter (QP) [2] , $\hat{f}_{\mathrm{u}}^k - \widetilde{f}_{\mathrm{u}}^k$ mainly depends on the PEP and the error concealment scheme. In addition, experimental results show that

[2] In the rate control algorithm design, quantization offset is often fixed.

$f_u^k - \hat{f}_u^k$ is zero-mean, which is also obvious in theory for encoders designed under MMSE criterion. Therefore, we make the following assumption.

Assumption 7. *$f_u^k - \hat{f}_u^k$ and $\hat{f}_u^k - \widetilde{f}_u^k$ are uncorrelated, and $E[f_u^k - \hat{f}_u^k] = 0$.*

Under Assumption 7, from (5–2), we obtain

$$\begin{aligned} D_{ETE}^k &= E[\frac{1}{|\mathcal{V}^k|}\sum_{u\in\mathcal{V}^k}(f_u^k - \hat{f}_u^k)^2] + E[\frac{1}{|\mathcal{V}^k|}\sum_{u\in\mathcal{V}^k}(\hat{f}_u^k - \widetilde{f}_u^k)^2] + \frac{2}{|\mathcal{V}^k|}\sum_{u\in\mathcal{V}^k} E[(f_u^k - \hat{f}_u^k)]E[(\hat{f}_u^k - \widetilde{f}_u^k)] \\ &= E[\frac{1}{|\mathcal{V}^k|}\sum_{u\in\mathcal{V}^k}(f_u^k - \hat{f}_u^k)^2] + E[\frac{1}{|\mathcal{V}^k|}\sum_{u\in\mathcal{V}^k}(\hat{f}_u^k - \widetilde{f}_u^k)^2] \qquad (5\text{–}3) \\ &= D_Q^k + D_T^k, \end{aligned}$$

where, the first term in the right-hand side is called frame-level quantization distortion (FQD), i.e., $D_Q^k \triangleq E[\frac{1}{|\mathcal{V}^k|}\sum_{u\in\mathcal{V}^k}(f_u^k - \hat{f}_u^k)^2]$ and the second term in the right-hand side is called frame-level transmission distortion (FTD), i.e., $D_T^k \triangleq E[\frac{1}{|\mathcal{V}^k|}\sum_{u\in\mathcal{V}^k}(\hat{f}_u^k - \widetilde{f}_u^k)^2]$.

In a typical video codec, the spatial correlation and temporal correlation is first removed by intra prediction and inter prediction. Then the residual is transformed and quantized. Given the uniform quantizer, D_Q^k only depends on the quantization step size Q^k and the video frame statistics ϕ_f^k. Therefore, we can express D_Q^k as a function of Q^k and ϕ_f^k, i.e., $D_Q(Q^k, \phi_f^k)$, where $D_Q(\cdot)$ is independent from the frame index k. In Chapter 2, we have derived D_T^k as a function of PEP, video frame statistics ϕ_f^k and system parameters ϕ_s^k, i.e., $D_T(PEP^k, \phi_f^k, \phi_s^k)$. Since PEP^k depends on SNR $\gamma(t)$, transmission bit rate R_t^k, and channel coding rate R_c^k, D_T^k also depends on the R_c^k. The higher channel coding rate, the higher PEP^k and thus the larger D_T^k. However, under the same bandwidth limit, the higher channel coding rate also means the fewer redundant bits or the higher source bit rate, and thus the smaller D_Q^k. In order to design the optimum Q^k and R_c^k to achieve the minimum D_{ETE}^k, we need to have PEP^k as a function of SNR $\gamma(t)$, transmission rate R_t^k, and R_c^k, i.e., $P(\gamma(t), R_t^k, R_c^k)$. Denote ϕ_c^k the channel statistics, i.e. $\phi_c^k = \{\gamma(t), R_t^k\}$, we can express D_T^k as a function of R_c^k, ϕ_c^k, ϕ_f^k, and ϕ_s^k, i.e., $D_T(R_c^k, \phi_c^k, \phi_f^k, \phi_s^k)$. On the other hand, $R_t^k = \frac{R_s^k}{R_c^k}$ where R_s^k denote the source

bit rate and it is a function of the quantization step size Q^k and video frame statistics ϕ_f^k, i.e., $R_s(Q^k, \phi_f^k)$.

Therefore, if we can derive the closed-form functions for $D_Q(Q^k, \phi_f^k)$, $D_T(PEP^k, \phi_f^k, \phi_s^k)$ and $R_s(Q^k, \phi_f^k)$, (5–1) can be solved by finding the best parameter pair $\{Q^k, R_c^k\}$. In other words, the problem in (5–1) is equivalent to

$$\begin{aligned} min \quad & D_Q(Q^k, \phi_f^k) + D_T(R_c^k, \phi_c^k, \phi_f^k, \phi_s^k) \\ s.t. \quad & \sum_{i=1}^{N^k} \frac{R_s(Q^k, \phi_{f,i}^k)}{R_{c,i}^k} \leq R_t^k, \end{aligned} \tag{5–4}$$

where N^k is the total number of packets in the k-th frame, and i is the packet index. In summary, our problem in (5–4) is "given the system structure ϕ_s^k, time-varying video frame statistics ϕ_f^k and time-varying channel statistics ϕ_c^k, how to minimize D_{ETE}^k by jointly controlling the parameters pair $\{Q^k, R_{c,i}^k\}$."

5.3 Derivation of Bit Rate Function, Quantization Distortion Function and Transmission Distortion Function

In this section, we derive the source rate function $R_s(Q^k, \phi_f^k)$, quantization distortion function $D_Q(Q^k, \phi_f^k)$, and transmission distortion function $D_T(PEP^k, \phi_f^k, \phi_s^k)$.

5.3.1 Derivation of Source Coding Bit Rate Function

5.3.1.1 The entropy of quantized transform coefficients for I.I.D. zero-mean Laplacian source under uniform quantizer

Following the similar deriving process as in Ref. [38, 42, 43], it is easy to prove that for independent and identically distributed (i.i.d.) zero-mean Laplacian source under uniform quantizer with quantization step size Q and quantization offset θ_2, the entropy of quantized transform coefficients is

$$H = -P_0 \cdot \log_2 P_0 + (1 - P_0) \cdot (\frac{\theta_1 \cdot \log_2 e}{1 - e^{-\theta_1}} - \log_2(1 - e^{-\theta_1}) - \theta_1 \cdot \theta_2 \cdot \log_2 e + 1), \tag{5–5}$$

where

$$\theta_1 = \frac{\sqrt{2} \cdot Q}{\sigma}; \tag{5–6}$$

Q is the quantization step size; σ is the standard deviation of the Laplacian distribution; θ_2 is the quantization offset; $P_0 = 1 - e^{-\theta_1 \cdot (1-\theta_2)}$ is the probability of quantized transform coefficient being zero. (5–5) is proved in Appendix D.1.

5.3.1.2 Improve with run length model

In a video encoder, the quantized transform coefficients are actually not i.i.d. Although we may assume the DCT transform or integer transform [22] highly de-correlates the correlation among neighboring pixels, different transform coefficients have very different variances in statistics. For example, in a 4x4 integer transform, the 16 coefficients show a decreasing variance in the well-known zigzag scan order as used in H.264. As a result, the coefficients with higher frequency have higher probability of being zeroes after quantization. On the other hand, the coefficients with lower frequency show more randomness in different levels even after quantization. Such characteristics are exploited by the run-level mapping after zigzag scan to further increase the compressibility for entropy coding. We may regard the component of **run-level mapping plus entropy coding** as choosing different codebooks for different quantized transform coefficients. From information theory, we know the concavity of the entropy as a function of the distribution (Theorem 2.7.3 in Ref [28]). Therefore, not considering the mixture of 16 coefficients with different variances will overestimate the entropy of mixed transform coefficients.

To derive the joint entropy for 16 coefficients with different variances, we need to model the variance relationship among those 16 coefficients. Having done extensive experiments, we find an interesting phenomenon[3] , that is, the variance is approximately

[3] This phenomenon is found from samples in one frame or one GOP for CIF sequences, i.e., the number of sample is larger than 101376.

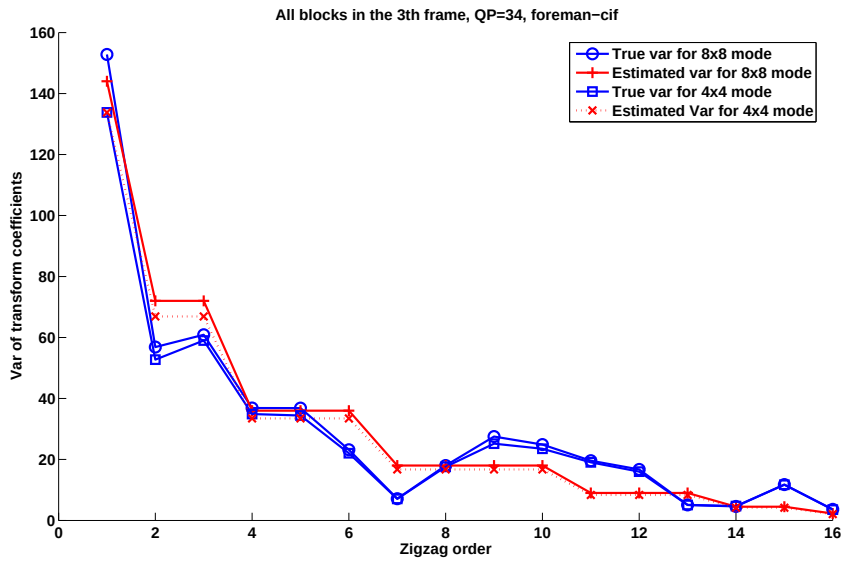


Figure 5-2. Variance model.

a function of position in the two-dimensional transform domain as follows

$$\sigma^2_{(x,y)} = 2^{-(x+y)} \cdot \sigma^2_0, \tag{5–7}$$

where x and y is the position in the two-dimensional transform domain, and σ^2_0 is the variance of the coefficient at position $(0, 0)$.

With (5–7), we can derive the variance $\sigma^2_{(x,y)}$ for all positions given the average variance σ^2 as in Appendix D.2. Fig. 5-2 shows the true variances and estimated variances by (D–5) for all transform coefficients before quantization in the third frame of 'foreman' sequence with $QP = 34$. We only show inter prediction modes 8x8 and 4x4 in Fig. 5-2. The results of other inter prediction modes [20] are similar. However, we also notice that due to the high correlation among all coefficients in intra prediction modes, the true variance of DC component is much larger than estimated variance by (D–5). The more accurate variance model for DC component in intra modes will be investigated in our future work.

Then, the estimated joint entropy of 16 non-identical transform coefficients by compensating the run length coding model is

$$H_{rlc} = \frac{1}{16}\sum_{x=0}^{3}\sum_{y=0}^{3} H_{(x,y)}, \tag{5–8}$$

where $H_{(x,y)}$ is the entropy for coefficient position (x, y), and can be calculated by (D–5), (5–6) and (5–5) with their own $\sigma^2_{(x,y)}$ and $\theta_{1(x,y)}$..

5.3.1.3 Practical consideration of Laplacian assumption

Statistically speaking, (5–8) is only valid for sufficiently large samples. When there are not enough samples or the sample variance is very small, e.g., $Q > 3\sigma$, the Laplacian assumption for individual coefficients is not accurate. In such cases, we may use the mixed distribution in (5–5) as the estimate instead of (5–8). That is,

$$H^k = \begin{cases} estimatedby\ (5\text{–}8), & Q \leq 3\sigma \\ estimatedby\ (5\text{–}5), & otherwise. \end{cases} \tag{5–9}$$

5.3.1.4 Improvement by considering the model inaccuracy

The assumed residual probability distribution, e.g., Laplacian distribution, may deviate significantly from the true histogram especially when the number of samples are not sufficient. Therefore, we need to compensate the mismatch between the true residual histogram and assumed Laplacian distribution to obtain a better estimate. Denote H_l as the entropy for the case with a Laplacian distribution, H_t as the entropy for the case with the true histogram and $\nu = \frac{H_l}{H_t}$. In a video sequence, the changes of residual statistics and quantization step size between adjacent frames have almost the same effect on H_l and H_t. Therefore, we may use the previous frame statistics to compensate the estimated result from (5–8). Assume the ratio between H_l^k and H_t^k approximate ν^{k-1}, we have $\frac{H_l^k}{H_t^k} = \frac{H_l^{k-1}}{H_t^{k-1}}$. As a result, (5–8) can be further compensated as

$$\hat{H}^k = \frac{H_t^{k-1} \cdot H^k}{H_l^{k-1}}. \tag{5–10}$$

Although very simple, (5–8) and (5–10) significantly improve the estimation accuracy of residual entropy as shown in Fig. 5-3.

5.3.1.5 Source coding bit rate estimation for the H.264 encoder

For a hybrid video coder with block-based coding scheme, e.g., H.264 encoder, the encoded bit rate R_s consists of residual bits R_{resi}, motion information bits R_{mv}, prediction mode bits R_{mode}, and syntax bits R_{syntax}. That is,

$$R_s^k = \hat{H}^k \cdot N_{resolution} \cdot N_{fps} + R_{mv}^k + R_{mode}^k + R_{syntax}^k, \tag{5–11}$$

where $N_{resolution}$ is the normalized video resolution considering color components, and N_{fps} means the number of frames per second. Compared to R_{resi}^k, R_{mv}^k, R_{mode}^k, and R_{syntax}^k are less affected by Q. Therefore, R_{mv}^k, R_{mode}^k, R_{syntax}^k can be estimated from the statistics in the previous frames.

5.3.2 Derivation of Quantization Distortion Function

In this subsection, we improve the estimation accuracy of quantization distortion by utilizing the same techniques in Section 5.3.1. In Ref. [38, 42], authors derive the distortion for zero-mean Laplacian residual distribution under uniform quantizer as

$$D_Q = \frac{Q^2 \cdot (\theta_1 \cdot e^{\theta_2 \cdot \theta_1} \cdot (2 + \theta_1 - 2 \cdot \theta_2 \cdot \theta_1) + 2 - 2 \cdot e^{\theta_1})}{\theta_1^2 \cdot (1 - e^{\theta_1})}, \tag{5–12}$$

Since the coefficients after transform is not identical in distribution, we need to derive the overall quantization distortion function by considering each coefficient individually. Using the variance relationship among coefficients in (5–7), we have

$$D_{overall} = \frac{1}{16} \sum_{x=0}^{3} \sum_{y=0}^{3} D_{(x,y)}, \tag{5–13}$$

where $D_{(x,y)}$ is the distortion for coefficient position (x, y), and can be calculated by (D–5), (5–6) and (5–12) with their own $\sigma_{(x,y)}^2$ and $\theta_{1(x,y)}$.

When there are not enough samples or the sample variance is very small, e.g., $Q > 3\sigma$, the Laplacian assumption for individual coefficients is not accurate. In such

cases, we may use the mixed distribution in (5–12) as the estimate instead of (5–13). That is,

$$D_Q^k = \begin{cases} estimatedby\ (5\text{–}13), & Q \leq 3\sigma \\ estimatedby\ (5\text{–}12), & otherwise. \end{cases} \tag{5–14}$$

Similarly, we need to compensate the mismatch between the true residual histogram and assumed Laplacian distribution for quantization distortion estimation. Denote $D_{Q,l}$ as quantization distortion for the case with a Laplacian distribution, $D_{Q,t}$ as quantization distortion for the case with the true histogram and $\mu = \frac{D_{Q,l}}{D_{Q,t}}$. (5–14) can be compensated as

$$\hat{D}_Q^k = \frac{D_{Q,t}^{k-1} \cdot D_{Q,l}^k}{D_{Q,l}^{k-1}}, \tag{5–15}$$

where $D_{Q,l}^k$ is calculated from (5–14). (5–13) and (5–15) significantly improve the estimation accuracy of quantization distortion as shown in Fig. 5-4.

5.3.3 Derivation of Transmission Distortion Function

In this subsection, we derive the FTD as a function of SNR, transmission rate, and channel coding rate.

5.3.3.1 Transmission distortion as a function of PEP

In Chapter 2, we derived the FTD formula under single-reference motion compensation and no slice data partitioning as

$$D_T^k = \bar{P}^k \cdot (E[(\varepsilon^k)^2] + \lambda^k \cdot E[(\xi^k)^2] + D^{k-1}) + (1 - \bar{P}^k) \cdot \alpha^k \cdot D^{k-1} \cdot (1 - \beta^k). \tag{5–16}$$

$\bar{P}^k$ is the weighted average PEP of all packets in the k-th frame; ε^k is the residual concealment error; ξ^k is the MV concealment error; β^k is the percentage of encoded I-MBs in the k-th frame; both the propagation factor α^k and the correlation ratio λ^k depend on video statistics, channel condition and codec structure, and are therefore called **system parameters**; D^{k-1} is the transmission distortion of the $k-1$ frame, which can be iteratively calculated by (5–16).

$\bar{P}^k$ is defined as $\bar{P}^k \triangleq \frac{1}{|\mathcal{V}^k|}\sum_{i=1}^{N^k}(P_i^k \cdot N_i^k)$, where N_i^k is the number of pixels contained in the i-th packet of the k-th frame; P_i^k is PEP of the i-th packet of the k-th frame; N^k is the total number of packets of the k-th frame. The other video frame statistics and system parameters can be easily estimated as described in Chapter 3. We will describe how to estimate PEP in the following subsections.

5.3.3.2 PEP as a function of SNR, transmission rate, and channel coding rate in a fading channel

Below, we analyze the conditional PEP for convolution coding scheme under wireless fading channel, given SNR. Since convolutional codes are linear codes, the probability of error can be obtained by assuming that the all-zero sequence is transmitted, and determining the probability that the decoder decides in favor of a different sequence [18]. The probability of mistaking transmitted sequence with a sequence, Hamming distance d away, is called pairwise error probability, and denoted as $P_2(d)$. With soft decision, if the coded symbols output from the convolutional encoder are sent over an AWGN channel using coherent BPSK modulation with energy $E_c = R_c * E_b$, then it can be show that

$$P_2(d) = Q(\sqrt{\frac{2E_c \cdot d}{N_0}}) = Q(\sqrt{2\gamma \cdot d}). \tag{5–17}$$

Before calculating the PEP, we need to analyze the first error probability, which is defined as the probability that another path that merges with the all-zero path at a given node has a metric that exceeds the metric of the all-zero path for the first time [52]. According to the definition, the first error probability can be approximated by its upper bound, i.e., the probability of mistaking the all-zero path for another path through the trellis, as

$$P_{fe} \leq \sum_{d=d_{free}}^{d_{max}} W_d * P_2(d), \tag{5–18}$$

where W_d is the weight spectrum of the specific convolutional code; d_{free} is the free distance of the specific convolutional code; d_{max} is the maximum distance between the

transmitted sequence and decoded sequence[4] . As a result, the PEP for a block of L decoded bits and for a given SNR can be upper-bounded as [53, 54]

$$PEP(\gamma) \leq 1 - (1 - P_{fe}(\gamma))^L \approx L \cdot P_{fe}(\gamma). \tag{5–19}$$

However, both upper bounds in (5–18) and (5–19) are only tight when γ is large. When γ is small such as in a fading channel, the resulted bound may be much larger than 1, i.e., $L \cdot P_{fe}(\gamma) \gg 1$. From our experimental results, we find that the $PEP(\gamma)$ follows waterfall shape when γ increase, that is, there exist a threshold γ_{th} such that, when $\gamma > \gamma_{th}$, the bound is quite tight, and when $\gamma < \gamma_{th}$, the bound becomes very loose and exceed 1 quickly. Therefore, we improve the performance bound by using the following formula.

$$PEP(\gamma) \approx \begin{cases} \frac{R_t \cdot R_c}{N^k \cdot N_{fps}} \cdot \sum_{d=d_{free}}^{d_{max}} W_d * P_2(d, \gamma), & \gamma \geq \gamma_{th} \\ 1, & \gamma < \gamma_{th}, \end{cases} \tag{5–20}$$

where γ_{th} can be numerically calculated from (5–21) given the convolutional encoder structure (W_d, d_{free}, and d_{max}), coding rate (R_c) and modulation scheme ($P_2(d)$). Note that W_d, d_{free}, and d_{max} in (5–20) are functions of R_c in RCPC. (5–20) is quite accurate as shown in Fig 5-5, where $PEP_{th} = 1$.

$$\sum_{d=d_{free}}^{d_{max}} \frac{R_t \cdot R_c}{N^k \cdot N_{fps}} \cdot W_d \cdot P_2(d, \gamma_{th}) = PEP_{th}. \tag{5–21}$$

Note that a change in the modulation and demodulation technique used to transmit the coded information sequence affects only the computation of $P_2(d)$ [52]. Therefore, (5–20) is general for any modulation scheme.

[4] d_{max} different from the formula in Ref. [52] due to the code is truncated by the packet length, which improve the upper bound but the effect on performance is negligible when packet length is large [52].

In a real-time video communication system, if the estimated $PEP(\gamma)$ is larger than a threshold value, i.e. $PEP(\gamma) > PEP_{th}$, transmitter may discard this packet instead of transmitting it.[5] The benefit of doing this is threefold: 1) if $PEP(\gamma)$ is large, it is a waste of energy and time to transmit the packet; therefore, using (5–20) saves transmission energy; 2) in cross-layer rate control, since video encoder has the knowledge of channel condition, video encoder will skip encoding current frame when the channel gain is very low, which saves the encoding energy; 3) If current frame is skipped, video encoder will use previous encoded frames as references for encoding the following frames, which reduce the reference error propagation.

(5–20) is derived under the condition that γ is known at the transmitter with channel estimation. In some wireless system, γ is unknown for transmitter, e.g., without feedback channel. In such case, the expected PEP, i.e. $E_\gamma[PEP]$, instead of PEP is used for estimating transmission distortion given the probability distribution of channel gain. Proposition 5 gives the formula of expected PEP under a Rayleigh block fading channel.

Proposition 5. *Under a Rayleigh block fading channel, the expected PEP is given by*

$$E_\gamma[PEP] = \frac{\gamma_{th}}{\overline{\gamma}} e^{-\frac{\gamma_{th}}{\overline{\gamma}}} (1 + \frac{1}{d_{free}\gamma_{th}}), \tag{5–22}$$

where γ_{th} is defined by (5–21)*.*

Proposition 5 is proved in Appendix D.3. We see from (D–7) that if $\gamma_{th} \geq \overline{\gamma}$, $E_\gamma[PEP] \geq 1 - e^{-\frac{\gamma_{th}}{\overline{\gamma}}} \geq 1 - e^{-1} \approx 0.63$. So, to control the PEP under a reasonable level, the transmitter should set its transmission power such that $\overline{\gamma} >> \gamma_{th}$ before transmitting the packet.

[5] In some delay-insensitive applications, e.g. streaming video, the buffer is used to hold packets when channel condition is poor. In such cases, the packet will be dropped at the transmitter only when the queue buffer is full or delay bound is violated, which will decrease the PEP.

5.3.3.3 Transmission distortion as a function of SNR, transmission rate, and channel coding rate in a fading channel

In case of adaptive modulation, adaptive transmission power and adaptive bandwidth (subcarrier) allocation, $P_2(d)$ is a function of modulation order M, transmission power P_t and passband bandwidth B. In this chapter, we study the case that modulation, power and bandwidth are all given during the cross-layer rate control. Under such conditions, both transmission bit rate R_t and SNR γ are known values. For example, with modulation order M and Nyquist pulse-shaping, $R_t = B \cdot log_2(M)$ and $\gamma = \frac{P_t T_t g}{N_0}$. As a result, both PEP and D_t depend only on the tuning parameter R_c.

5.4 Rate-Distortion Optimized Cross-layer Rate Control and Algorithm Design

In this section, we apply our models derived in Section 5.3 to cross-layer rate control application. We adopt the discrete version of Lagrange multiplier as used in JM [33] to achieve the R-D optimized parameter pair $\{Q^k, R^k_{c,i}\}$. We also design a practical cross-layer rate control algorithm.

5.4.1 Optimization of Cross-layer Rate Control Problem

To solve (5–4), we may either use Lagrangian approaches or dynamic programming approaches [44]. In terms of complexity, the Lagrangian approach is preferable, since it can be run independently in each coding unit, whereas dynamic programming requires a tree to be grown. Note that the complexity of the dynamic programming approaches can grow exponentially with the number of coding units considered, while the Lagrangian approach's complexity only grow linearly [44]. By using the theorem in Ref. [50, 51], we may use the Lagrangian approach for the i-th packet in the k-th frame independently as

$$(Q^k_i, R^k_{c,i})^* = \arg\min\{D_Q(Q^k_i) + D_T(R^k_{c,i}) + \lambda \cdot \frac{R_s(Q^k_i)}{R^k_{c,i}}\}, \tag{5–23}$$

where λ is the preset Lagrange multiplier, which can be determined either by bi-section search [50, 55] or by modeling [33, 56].

For some special cases, e.g. video conference, the frame size is usually small. In such a case, each frame is transmitted in one packet, and therefore, the bit allocation problem can be simplified. To be more specific, since all bits are allocated into one packet, given γ and R_t, every R_c have a corresponding R_s; every R_s has a corresponding Q and therefore D_Q (by (5–11), (5–10), (5–5) and (5–14)). As mentioned in Section 5.3.3.3, D_T is also a function of R_c. In other words, the end-to-end distortion D_{ETE}^k only depends on R_c. Therefore, there exists an optimum R_c^k, such that D_{ETE} is minimized. As a result, Lagrange multiplier can be omitted. That is, the optimum R_c^k can be achieved by comparing D_{ETE} for all possible R_c, and the optimum Q^k can be calculated by the corresponding R_c^k.

5.4.2 Algorithm Design

In this subsection, we propose a practical algorithm for cross-layer rate-distortion optimization as following.

Algorithm 4. *Cross-layer optimized quantization step size Q and channel coding rate R_c decision for the k-th frame.*

1) ***Input****: R_t, γ, PEP_{th}.*

2) Initialization of Q^0 and R_c^0 for the first frame, i.e. $k = 1$. If $k > 1$, go to 3).

3a) If $N^k > 1$, i.e., each frame is contained in more than one packet.

Initialize $\Lambda_j = \Lambda_0$ by the method proposed in Ref. [50]

loop for $\Lambda_j = \Lambda_0, \Lambda_1, ..., \Lambda^*$,

for packet index i from 1 to $N^k > 1$,

For each packet, loop for all combinations of $\{Q, R_c\}$ under the given Λ_j

calculate γ_{th} by (5–21),

estimate P_i^k for all packets by (5–20),

estimate D_T by (5–16),

estimate θ_1 by (5–6),

estimate D_Q by (5–15),

calculate $D_{ETE}(Q, R_c)$ *by* (5–3),

estimate $R_{s,i}^k$ *by* (5–5), (5–8), (5–10) *and* (5–11),

End

obtain the best $\{Q_i^k(\Lambda_j), R_{c,i}^k(\Lambda_j)\}$, *i.e.,* $\{Q_i^k(\Lambda_j), R_{c,i}^k(\Lambda_j)\}^*$ *via* (5–23),

End

estimate R_t^k *by* $R_{s,i}^k$ *and* $R_{c,i}^k$,

calculate Λ_{j+1},

End

obtain the best $\{Q_i^k, R_{c,i}^k\}$, *i.e.,* $\{Q_i^k(\Lambda^*), R_{c,i}^k(\Lambda^*)\}^*$, *for each packet.*

3b) If $N^k = 1$, *i.e., each frame is contained in one packet.*

loop for all channel coding rates.

calculate γ_{th} *by* (5–21),

estimate PEP *for the* k*-th frame by* (5–20),

estimate D_T^k *by* (5–16),

estimate $\hat{H}^k$ *by* (5–11),

estimate Q^k *by* (5–10), (5–8), (5–5) *and* (5–6),

estimate $\hat{D}_Q^k$ *by* (5–15),

calculate $D_{ETE}(R_c)$ *by* (5–3),

End

select the best R_c^k *and corresponding* Q^k *with minimum end-to-end distortion.*

4) ***Output****: the best* $\{Q_i^k, R_{c,i}^k\}$.

Algorithm 4 is referred to as CLRC. Note that in Algorithm 4, the iterations to acquire the best Lagrange multiplier Λ^* use bi-section search [50, 55]. Since loop for all combinations of $\{Q, R_c\}$ is executed for each candidate Lagrange multiplier, the complexity is very high. We may also use the modeling method [33, 56] instead of bi-section search to design CLRC. In such a case, R-D optimized $\{Q, R_c\}$ decision is

similar to the R-D optimized mode decision in Ref. [33] except three differences: 1) the mode decision is replaced by channel coding rate decision given the Lagrange multiplier, 2) the quantization distortion is replaced by the End-to-end distortion, 3) the source coding bit rate is replaced by the transmission bit rate. Note that the modeling method reduces the complexity to estimate the best $\{Q_i^k, R_{c,i}^k\}$ at the cost of accuracy.

5.5 Experimental Results

In Section 5.5.1, we verify the accuracy of our proposed models. Then in Section 5.5.2, we compare the performance between our CLRC algorithm and existing rate control algorithms.

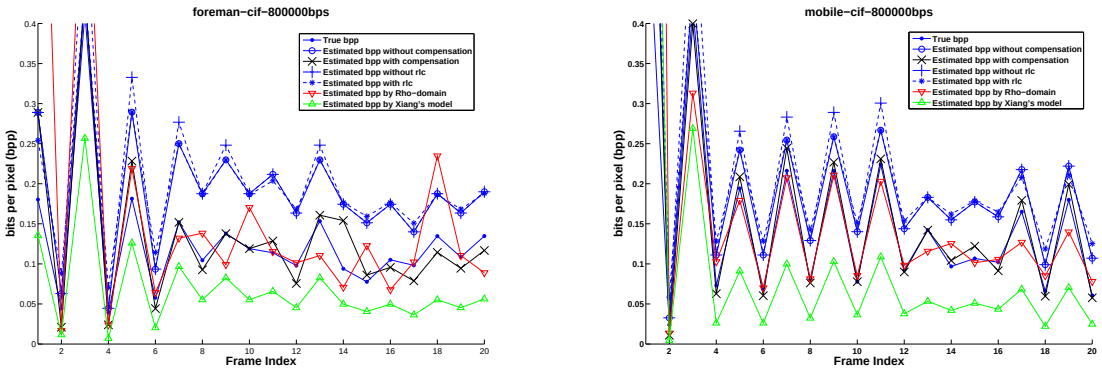
5.5.1 Model Accuracy

In this subsection, we test the bit rate model proposed in (5–10), distortion model proposed in (5–15), and PEP model proposed in (5–20).

5.5.1.1 Bit rate model

The JM16.0 encoder is used to collect the true distortion and required statistics. Fig. 5-3 shows the true residual bit rate and estimated residual bit rate for 'foreman' and 'mobile' for the first 20 frames in order to make different curves distinguishable. In Fig. 5-3, 'True bpp' means the true bit per pixel (bpp) produced by the JM16.0 encoder; 'without rlc' means bpp estimated by (5–5); 'with rlc' means bpp estimated by (5–8); 'without compensation' means bpp estimated by (5–9); 'with compensation' means bpp estimated by (5–9) and (5–10); 'Rho-domain' means bpp estimated by Refs. [10, 57]; 'Xiang's model' means bpp estimated by Refs. [38, 58].

We can see that the estimation accuracy is improved by (5–8) when true bpp is relatively large. However, when true bpp is small, 'without rlc' gives higher estimation accuracy. By utilizing the statistics of the previous frame from (5–10), the estimation accuracy is further improved. We also find that 'Rho-domain' is accurate at low bpp; however, it is not accurate at high bpp. For 'Xiang's model', the estimated bpp is smaller than the true bpp in most cases. Note that we also want to compare the bit rate model



(a) (b)

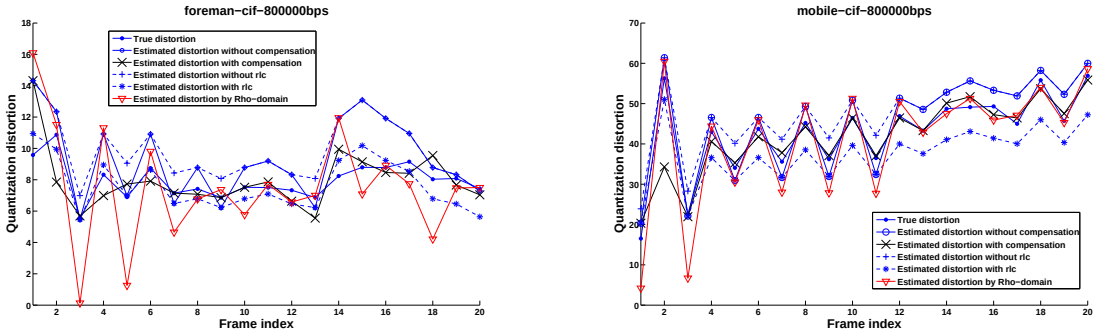
Figure 5-3. bpp vs. Frame index: (a) foreman, (b) mobile.

used in JM16.0. However, due to the estimation error of its model parameters, the first few frames may abnormally underestimate the quantization step size Q. Therefore, the rate control algorithm in JM16.0 use three parameters, i.e., RCMinQPPSlice, RCMaxQPPSlice and RCMaxQPChange, to reduce the effect of the estimation error. Their default values are 8, 42, 4, respectively. However, we believe a good rate control algorithm should depend mainly on the model accuracy rather than those manually chosen thresholds. When those parameters are set as 0, 51, 51, the estimated QP could even be 0 in the first few frames. That is, the first few frames consume most of the allocated bits, and there are only few bits available for the remaining frames in JM. Therefore, we do not test its model accuracy in this subsection. Instead, we will plot the R-D performance for it in Section 5.5.2.

5.5.1.2 Quantization distortion model

Fig. 5-4 shows the corresponding quantization distortion of each bit rate curve in Fig.5-3. Note that since Refs. [38, 58] directly use (5–12) to estimate the quantization distortion, 'without rlc' means the quantization distortion estimated by both (5–12) and 'Xiang's model'. Similar to Fig.5-3, we can see that the estimation accuracy is improved by (5–13) when θ_1 is small, i.e., when quantization distortion is relatively small. However, when quantization step size is large, (5–12) is more accurate than (5–13). Note that, the relativity is for the same video sequence. For different video sequences, since the

residual variances are different, in order to achieve the same bit rate, sequences with larger variance, e.g., 'mobile', will use higher quantization step size than sequences with lower variance, e.g., 'foreman'. Different from the bit rate model, which depends only on θ_1, the quantization distortion model depends on both Q and θ_1. Therefore, we cannot use the absolute value of quantization distortion between two sequences for comparing estimation accuracy of (5–12) and (5–13). After normalized by the factor Q^2 in (5–12) and (5–13), their relative accuracy is valid in most cases. However, in some rare cases, (5–13) is more accurate than (5–12) even when $Q > 3\sigma$. This can be observed for frame index from 14 to 17 in foreman sequence. We still need to investigate the reason behind it to further improve our model accuracy. For all cases, the estimation accuracy is improved by utilizing the statistics of the previous frame from (5–15). Similar to Fig.5-3, 'rho-domain' is more accurate at large θ_1, i.e., low bit rate or relatively large quantization distortion, than at small θ_1.



(a) (b)

Figure 5-4. Quantization vs. Frame index: (a) foreman, (b) mobile.

5.5.1.3 PEP model

Here we verify the accuracy of PEP model derived in (5–20). We use the RCPC codes from Table I-VI in Ref [59]. To be more specific, we choose a typical convolutional encoder structure with constraint length 7, i.e. 6 memory registers, $G1 = 133$ and $G2 = 171$. The channel coding rates are 2/3, 3/4, 4/5, 5/6, 6/7 and 7/8. For completeness, we put all encoder parameters in Table 5-1. Viterbi algorithm is used to decode the received

bits with noise. BPSK modulation is used. Each packet contains 2000 information bits. For each SNR and channel coding rate, there are 1000 packets simulated to collect the true packet error rate (PER).

Table 5-1. RCPC encoder parameters

coding rate	puncturing matrix	dfree	weight spectrum
2/3	[1 1, 1 0]	6	[1, 16, 48, 158, 642, 2435, 9174, 34701, 131533, 499312]
3/4	[1 1, 1 0, 0 1]	5	[8, 31, 160, 892, 4512, 23297, 120976, 624304, 3229885, 16721329]
4/5	[1 1, 1 0, 1 0, 1 0]	4	[3, 24, 172, 1158, 7408, 48706, 319563, 2094852, 13737566, 90083445]
5/6	[1 1, 1 0, 0 1, 1 0, 0 1]	4	[14, 69, 654, 4996, 39677, 314973, 2503576, 19875546, 157824160, 1253169928]
6/7	[1 1, 1 0, 1 0, 0 1, 1 0, 0 1]	3	[1, 20, 223, 1961, 18084, 168982, 1573256, 14620204, 135966265, 1264590899]
7/8	[1 1, 1 0, 1 0, 1 0, 0 1, 1 0, 0 1]	3	[2, 46, 499, 5291, 56137, 598557, 6371293, 67889502, 723039772, 7701832191]

Fig. 5-5 shows the true PER and estimated PEP by the upper bound in (5–20). We can see that the estimated PEP curve is only about 1dB higher than the corresponding true PER curve. [6]

5.5.2 Performance Comparison

In this subsection, we show both objective performance and subjective performance of CLRC algorithm. In order to see the gain achieved by channel estimation, we also compare the performance achieved by (5–20) and (5–22). This result may serve as a guideline for system design to balance the performance and cost. Using (5–22), we can also compare the performance gain achieved by our models from existing models.

[6] From the experimental results, we observe that the estimated PEP curve shows an constant offset from the true PER curve given the RCPC encoder structure, and different RCPC encoder structure shows a different offset. We may utilize this observation to further improve the PEP model in our future work.

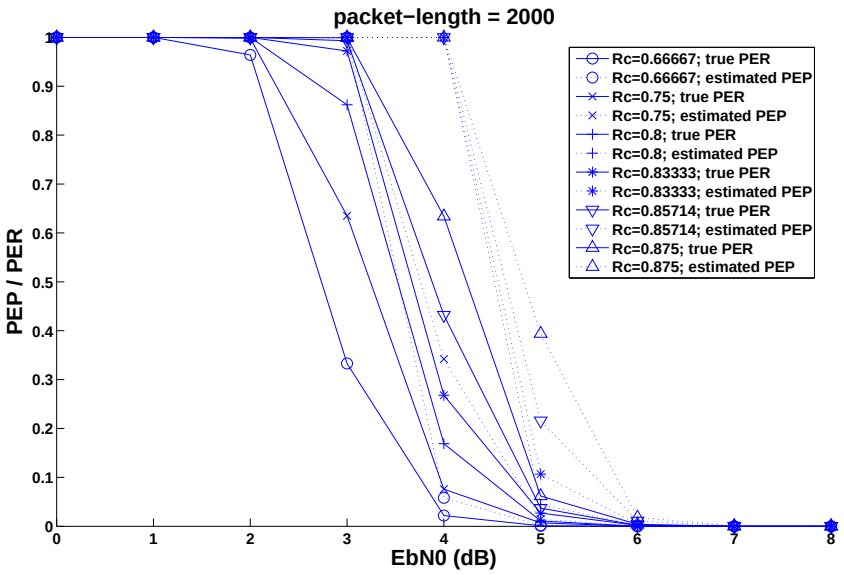


Figure 5-5. PEP under different RCPC coding rates.

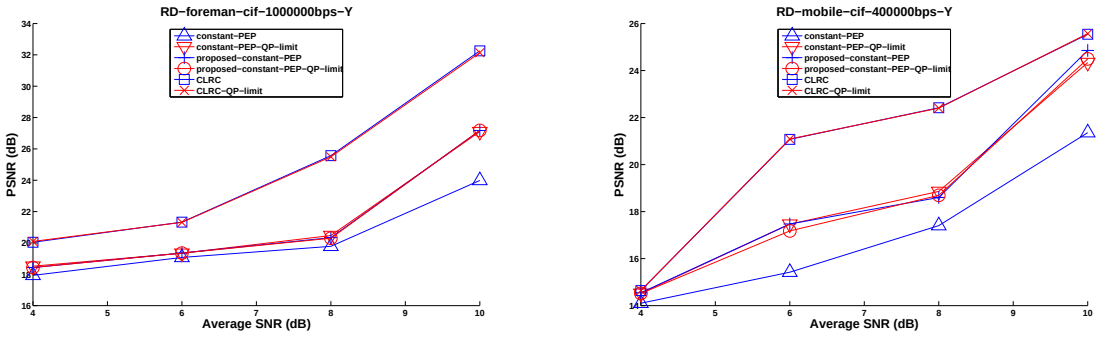
5.5.2.1 Experiment setup

The JM16.0 encoder and decoder [33] are used in the experiments. All the tested video sequences are in CIF resolution at 30fps. Each video sequence is encoded for its first 30 frames where the first frame is an I-frame and the following frames are P-frames. The error concealment method is to copy the pixel value in the same position of the previous frame. The first frame is assumed to be correctly received with enough channel protection or timely acknowledgement feedback. The encoder setting is given as below: Constrained intra prediction is enabled; the number of reference frames is 5; B slices are not included; only 4x4 transform is used; CABAC is enabled for entropy coding; For all rate control algorithms, the first frame use a fix QP, i.e., QP=28.

Each coded video sequence is tested under different rayleigh fading channels, i.e., different combinations of bandwidth from 100Kbps to 1Mbps and average SNR from 4dB to 10dB. For each specific channel condition, we simulate 300 random packet error processes to mitigate the effect of error randomness on each frame. RCPC codes and modulation are the same as those in Section 5.5.1.3.

5.5.2.2 PSNR performance

Figs. 5-6 shows Y-component PSNR vs. average SNR for ‘foreman’ and ‘mobile’. In Fig. 5-6, ‘proposed-constant-PEP’ represent the performance achieved by our models without channel estimation, i.e., using (5–22). ‘constant-PEP’ represent the performance achieved by the default rate control algorithm in JM16.0, i.e., JVT-H017r3 [33, 60, 61], without channel estimation. For each algorithm, we test two parameter settings of (RCMinQPPSlice, RCMaxQPPSlice, RCMaxQPChange), i.e., (8, 42, 4) and (0, 51, 51) to see how accurate those models are under different manually set thresholds. The experimental results show that under the same QP-limitation range, CLRC achieves up to 5dB PSNR gain in ‘foreman’ and up to 4dB PSNR gain in ‘mobile’ over no channel estimation. We observe that both ‘CLRC’ and ‘proposed-constant-PEP’ show very stable result when the QP-limitation range varies, while ‘constant-PEP’ show very different results under different QP-limitation ranges. To be more specific, for ‘constant-PEP’ the smaller QP-limitation range gives up to 3dB PSNR gain over larger QP-limitation range. This phenomenon further proves the higher accuracy of our models.



(a) (b)

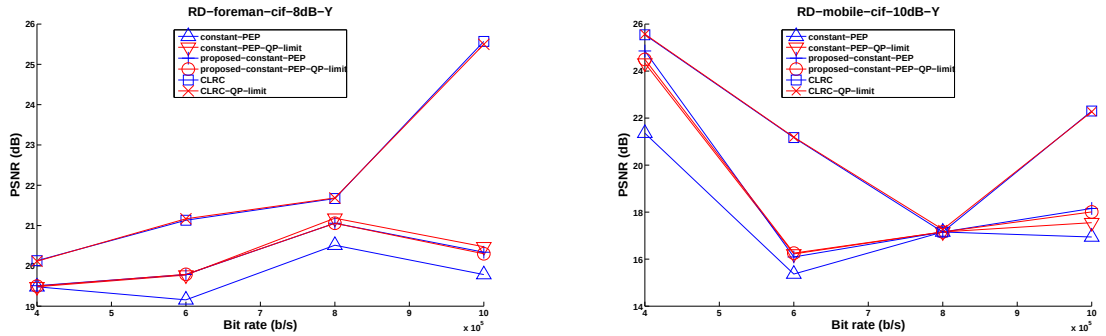
Figure 5-6. PSNR vs. average SNR: (a) foreman, (b) mobile.

Note that in Ref. [10], authors also propose bit rate model, quantization distortion model and transmission distortion model for solving joint source channel rate control problem. However, in both that bit rate model and quantization distortion model, only the model parameter, i.e., ‘rho’, can be estimated from a given bit rate or quantization

distortion. In order to estimated the quantization step size or QP before real encoding, those models requires the prior knowledge of residual histogram [62, 63]. Since H.263 encoders usually use mean square error (MSE) as a criterion for motion estimation, this kind of prior knowledge is accessible in H.263 after motion estimation and before quantization. However, it is not available in H.264 encoders since R-D cost instead of MSE is adopted as a criterion for motion estimation and mode decision. The R-D cost function induces a Lagrange multiplier, which can only be determined after QP is known. Therefore, their bit rate model encounters a chicken-and-egg problem if one tries to apply it for estimating quantization step size in H.264 encoders. Due to this reason, we do not implement those models in Ref. [10] for cross-layer rate control in the H.264 encoder [33]. Note that since the model parameters in Ref. [10] is attainable after real encoding, we still compare their model accuracy in Section 5.5.1. For the accuracy comparison between our transmission distortion model and the transmission distortion model in Ref. [10], please refer to Chapter 3.

Fig. 5-7 shows Y-component PSNR vs. bandwidth for 'foreman' and 'mobile'. We see the similar results as in Fig. 5-6. That is, 1) 'CLRC' achieves the best performance; 2) both 'CLRC' and 'proposed-constant-PEP' show very stable result when the QP-limitation range varies, while 'constant-PEP' show very different results under different QP-limitation ranges. However, we also observe in our experiments that PSNR shows more randomness for a given SNR in Fig. 5-7 than in Fig. 5-6. For example, for 'mobile' the PSNR at $400kbps$ is even higher than PSNR at all other bit rates. After investigation, we find this is due to the randomness of produced SNR sample sequence for a given average SNR in a fading channel. In other words, the randomness of SNR sample sequence has more impact on distortion than bit rate in a fading channel. To mitigate the effect of the randomness, we should simulate sufficiently large number of SNR sample sequences for a given average SNR. Unfortunately, this is prohibitively time consuming and therefore impractical to simulate. For example, if we simulate 1000

SNR sample sequences with 30 frames per sequences for average SNR 8dB; for each SNR sample sequence, 300 random packet error processes are simulated to mitigate the effect of error randomness on each frame; In order to plot PSNR vs. bandwidth for 6 algorithms and settings with 4 bit rates, we need $1000 \times 30 \times 6 \times 4$ frames encoding operations and $1000 \times 30 \times 300 \times 6 \times 4$ frames decoding operations, which at least needs 16,000 hours by using JM16.0 [33] in a PC with a 2.29GHz CPU.



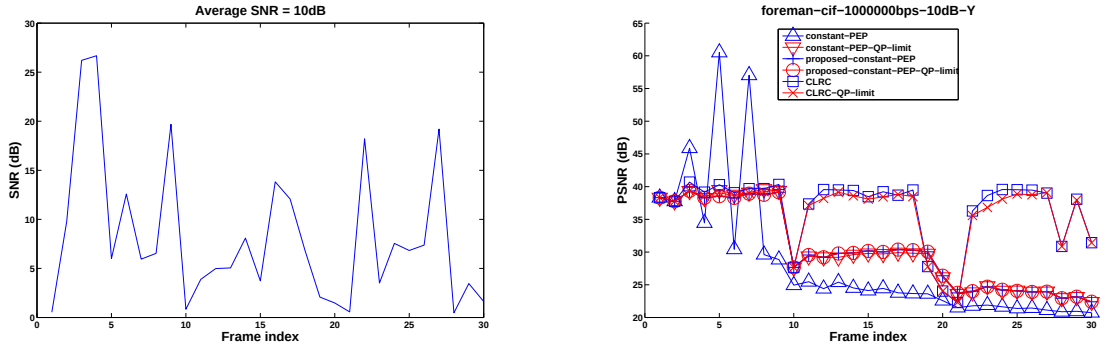
(a) (b)

Figure 5-7. PSNR vs. bandwidth: (a) foreman, (b) mobile.

5.5.2.3 Subjective performance

Since PSNR could be less meaningful for error concealment, a much more important performance criterion is the subjective performance, which directly relates to the degree of user's satisfaction. By utilizing the channel information, i.e., SNR and bandwidth, our CLRC algorithm intelligently chooses the reference frames which are transmitted under the best channel conditions and neglects those references frames which experience poor channel conditions. As a result, the well-known error propagation problem is prohibited even during the encoding process.

To illustrate the subjective performance, we plot four frames from the foreman sequence. Fig. 5-8(a) shows a random channel sample under average SNR=10dB and bit rate=1000kbps; Fig. 5-8(b) shows Distortion vs. Frame index for foreman_cif under this channel; Fig. 5-9 shows the corresponding subjective quality of reconstructed frames. We see that due to a low channel SNR during the timeslots of the 10-th



(a) (b)

Figure 5-8. A random channel sample under average SNR=10dB and bit rate=1000kbps: (a) A random SNR sample, (b) Distortion vs. Frame index for foreman_cif under this channel.

frame, the encoder with CLRC skip encoding these three frames to save encoding and transmission energy. Since there are no packets transmitted, the reconstructed picture of both those three frames at the decoder are the same as at the encoder. Then, when the channel condition goes well in the 11-th frame, encoder with CLRC use the 9-th frame as reference to reconstruct the 11-th fame. Since the channel condition is good in the timeslot of the 11-th frame, there are no transmission distortion at the decoder. Therefore, the error propagation is prohibited in the following frames.

For the encoder without channel estimation, the 10-th frame is encoded and transmitted. Due to the low channel SNR during the timeslots of the 10-th frame, the packets are received with error at the receiver and therefore, the resulted PSNR is almost the same as that of encoder with CLRC. However, without channel information, the encoder still use the 10-th frame as one of the references for encoding the 11-th frame. Therefore, although the 11-th frame is correctly received at the receiver due to good channel condition, the reconstructed error in the 10-th frame are propagated into the 11-th frame at the decoder, which causes both lower subjective quality and PSNR comparing to the encoder with CLRC. In Fig. 5-9, due to the space limit, we only show the subjective quality for encoder with ‘constant-PEP’ under default QP limitation

range. As we may foresee, the subjective quality for encoder with ‘constant-PEP’ under maximum QP limitation range is the worst among all cases.

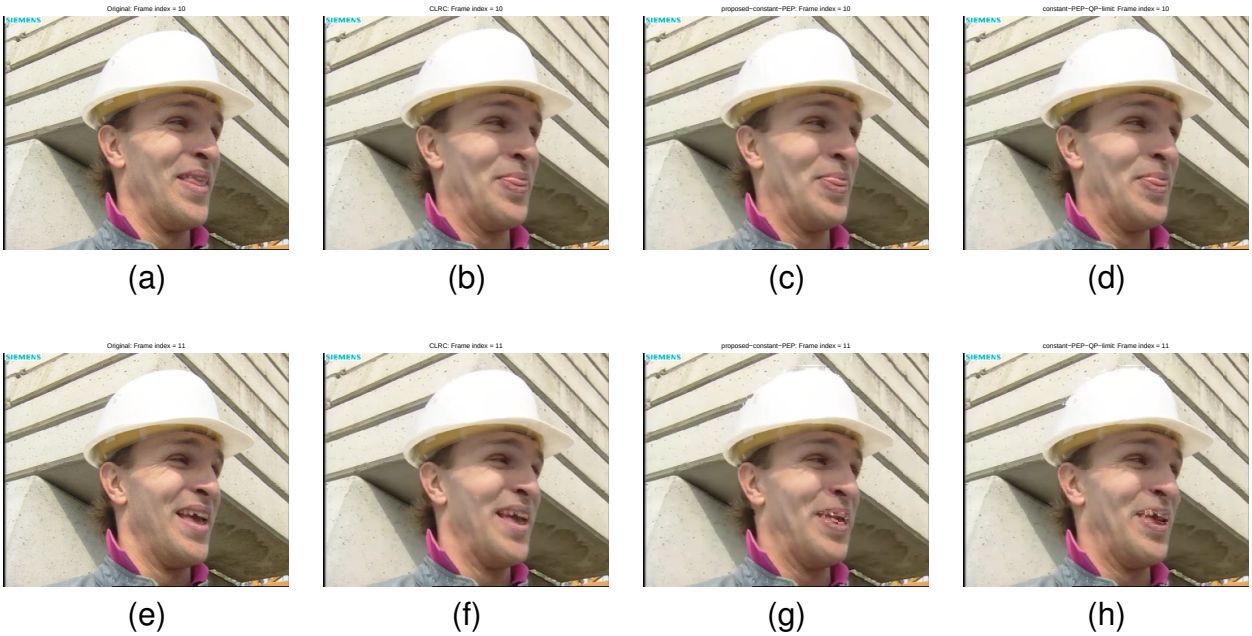


Figure 5-9. For the 10-th frame: (a) original, (b) CLRC, (c) proposed-constant-PEP, (d) constant-PEP-QP-limit; for the 11-th frame: (e) original, (f) CLRC, (g) proposed-constant-PEP, (h) constant-PEP-QP-limit.

CHAPTER 6
CONCLUSION

6.1 Summary of the Dissertation

In this work, we addressed the problem of minimizing end-to-end distortion in wireless video communication system. In Chapter 1, we explained the theoretical background and practical challenges for solving this problem. We also summarized our contributions in Chapter 1.

In Chapter 2, we analytically derived the transmission distortion as a function of video statistics, packet error probability and system parameters. With consideration of spatio-temporal correlation, nonlinear codec and time-varying channel, our formulae provide, for the first time, the following capabilities: 1) support of distortion prediction at different levels (e.g., pixel/frame/GOP level), 2) support of multi-reference picture motion compensated prediction, 3) support of slice data partitioning, 4) support of arbitrary slice-level packetization with FMO mechanism, 5) being applicable to time-varying channels, 6) one unified formula for both I-MB and P-MB, and 7) support of both low motion and high motion video sequences. Besides deriving the transmission distortion formulae, in Chapter 2, we also identified two important properties of transmission distortion for the first time: 1) clipping noise, produced by non-linear clipping, causes decay of propagated error; 2) the correlation between motion vector concealment error and propagated error is negative, and has dominant impact on transmission distortion, among all the correlations between any two of the four components in transmission error. We also discussed the relationship between our formula and existing models; we specify the conditions, under which those existing models are accurate.

In Chapter 3, we designed RMPC-FTD, RMPC-PTD, RMPC-PEED algorithms based on the analysis in Chapter 2. By virtue of considering the non-linear clipping noise and the negative correlation between the MV concealment error and the propagated error, RMPC-FTD algorithm provides more accurate FTD estimation

than existing models as verified by experimental results. In addition, experimental results also show that RMPC-FTD algorithm is more robust against inaccurate estimation of PEP than existing models. We also designed RMPC-MS algorithm for mode decision in H.264. Experimental results show that our RMPC-MS algorithm achieves an remarkable performance gain than existing algorithms.

In Chapter 4, we proved a new theorem for calculating the second moment of a weighted sum of correlated random variables without requiring knowledge of the probability distributions of the random variables. Then, we applied the theorem to design a very low-complexity algorithm to extend the RMPC algorithm to perform mode decision. Experimental results show that, the new algorithm, ERMPC, achieves further performance gain over the existing RMPC algorithm.

In Chapter 5, we derived more accurate source bit rate model and quantization distortion model than existing parametric models. We also improved the performance bound of channel coding with convolutional codes and a Viterbi decoder, and derived its performance under Rayleigh block fading channels. Given the instantaneous channel condition, i.e. SNR and bandwidth, we designed a rate-distortion optimized CLRC algorithm by jointly choosing quantization step size and channel coding rate. Experimental results showed that our proposed R-D models are much more accurate than existing R-D models. Experimental results also showed that the rate control algorithm with our models achieves superior PSNR gain than the existing rate control algorithm in JM. The other more important result is that the subjective quality of our CLRC algorithm is much better than existing algorithms due to its intelligent reference frame selection.

6.2 Future Work

First, we will work on modeling the Lagrange multiplier. Current RMPC and ERMPC algorithms still use the same Lagrange multiplier λ as that for source coding RDO. However, in an error-prone channel, λ is a function of video content, MV, mode, QP, PEP,

error concealment scheme, and constrained bit rate. One of our future research topics is to analytically derive the optimal λ for wireless video transmission, and then design an ERRDO scheme for joint MV, mode, QP selection.

Second, we will also investigate the effect of delay constraint on the end-to-end distortion. Current CLRC algorithm does not address the delay constraint. However, in some delay-insensitive applications, e.g. streaming video, the buffer is used to hold packets when channel condition is poor. In such cases, the packet will be dropped at the transmitter only when the queue buffer is full or delay bound is violated, which will decrease the PEP. On the other hand, the stringent encoding delay constraint is also relaxed, which improves the video quality. In other words, the time resource can be utilized to reduce the end-to-end distortion. In our future work, we will investigate how to minimize the end-to-end distortion with more time diversity in the wireless video communication systems.

APPENDIX A
PROOFS IN CHAPTER 2

A.1 Proof of Lemma 1

Proof. From (2–11) and (2–13), we obtain $\widetilde{f}^{k-1}_{\mathrm{u}+\widetilde{\mathrm{mv}}^k_{\mathrm{u}}} + \widetilde{e}^k_{\mathrm{u}} = \hat{f}^k_{\mathrm{u}} - \widetilde{\xi}^k_{\mathrm{u}} - \widetilde{\varepsilon}^k_{\mathrm{u}} - \widetilde{\zeta}^{k-1}_{\mathrm{u}+\widetilde{\mathrm{mv}}^k_{\mathrm{u}}}$,

Together with (2–9), we obtain

$$\widetilde{\Delta}^k_{\mathrm{u}} = (\hat{f}^k_{\mathrm{u}} - \widetilde{\xi}^k_{\mathrm{u}} - \widetilde{\varepsilon}^k_{\mathrm{u}} - \widetilde{\zeta}^{k-1}_{\mathrm{u}+\widetilde{\mathrm{mv}}^k_{\mathrm{u}}}) - \Gamma(\hat{f}^k_{\mathrm{u}} - \widetilde{\xi}^k_{\mathrm{u}} - \widetilde{\varepsilon}^k_{\mathrm{u}} - \widetilde{\zeta}^{k-1}_{\mathrm{u}+\widetilde{\mathrm{mv}}^k_{\mathrm{u}}}). \tag{A–1}$$

So, $\widetilde{\zeta}^{k-1}_{\mathrm{u}+\widetilde{\mathrm{mv}}^k_{\mathrm{u}}} + \widetilde{\Delta}^k_{\mathrm{u}} = (\hat{f}^k_{\mathrm{u}} - \widetilde{\xi}^k_{\mathrm{u}} - \widetilde{\varepsilon}^k_{\mathrm{u}}) - \Gamma(\hat{f}^k_{\mathrm{u}} - \widetilde{\xi}^k_{\mathrm{u}} - \widetilde{\varepsilon}^k_{\mathrm{u}} - \widetilde{\zeta}^{k-1}_{\mathrm{u}+\widetilde{\mathrm{mv}}^k_{\mathrm{u}}})$, and

$$D^k_{\mathrm{u}}(P) = E[(\widetilde{\zeta}^{k-1}_{\mathrm{u}+\widetilde{\mathrm{mv}}^k_{\mathrm{u}}} + \widetilde{\Delta}^k_{\mathrm{u}})^2] = E[\Phi^2(\widetilde{\zeta}^{k-1}_{\mathrm{u}+\widetilde{\mathrm{mv}}^k_{\mathrm{u}}},\ \hat{f}^k_{\mathrm{u}} - \widetilde{\xi}^k_{\mathrm{u}} - \widetilde{\varepsilon}^k_{\mathrm{u}})]. \tag{A–2}$$

We know from the definition that $D^k_{\mathrm{u}}(p)$ is a special case of $D^k_{\mathrm{u}}(P)$ under the condition $\{\bar{r}, \bar{m}\}$, which means $\widetilde{e}^k_{\mathrm{u}} = \hat{e}^k_{\mathrm{u}}$, i.e. $\widetilde{\varepsilon}^k_{\mathrm{u}} = 0$, and $\widetilde{\mathbf{mv}}^k_{\mathrm{u}} = \mathbf{mv}^k_{\mathrm{u}}$, i.e. $\widetilde{\xi}^k_{\mathrm{u}} = 0$. Therefore, we obtain

$$D^k_{\mathrm{u}}(p) = E[\Phi^2(\widetilde{\zeta}^{k-1}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}},\ \hat{f}^k_{\mathrm{u}})]. \tag{A–3}$$

□

A.2 Proof of Proposition 1

Proof. The probability density function of the random variable having a Laplacian distribution is $f(x|\mu, b) = \frac{1}{2b}\exp\left(-\frac{|x-\mu|}{b}\right)$. Since $\mu = 0$, we have $E[x^2] = 2b^2$, and from (3–10), we obtain

$$\begin{aligned} E[x^2] - E[\Phi^2(x,\ y)] &= \int_{y-\gamma_L}^{+\infty} (x^2 - (y-\gamma_L)^2)\frac{1}{2b}e^{-\frac{x}{b}}dx + \int_{-\infty}^{y-\gamma_H} [x^2 - (y-\gamma_H)^2]\frac{1}{2b}e^{\frac{x}{b}}dx \\ &= e^{-\frac{y-\gamma_L}{b}}((y-\gamma_L)\cdot b + b^2) + e^{-\frac{\gamma_H - y}{b}}((\gamma_H - y)\cdot b + b^2). \end{aligned} \tag{A–4}$$

From the definition of propagation factor, we obtain $\alpha = \frac{E[\Phi^2(x,y)]}{E[x^2]} = 1 - \frac{1}{2}e^{-\frac{y-\gamma_L}{b}}(\frac{y-\gamma_L}{b}+1) - \frac{1}{2}e^{-\frac{\gamma_H - y}{b}}(\frac{\gamma_H - y}{b}+1)$. □

A.3 Proof of Lemma 2

Proof. For P-MBs with slice data partioning, from (2–19) and (2–35) we obtain

$$D^k(P) = \frac{1}{|\mathcal{V}|}\sum_{\mathrm{u}\in\mathcal{V}^k}(P^k_{\mathrm{u}}\{r,m\}\cdot D^{k-1}_{\mathrm{u}+\check{\mathrm{m}}\mathrm{v}^k_{\mathrm{u}}}) + \frac{1}{|\mathcal{V}|}\sum_{\mathrm{u}\in\mathcal{V}^k}(P^k_{\mathrm{u}}\{r,\bar{m}\}\cdot D^{k-1}_{\mathrm{u}+\mathrm{m}\mathrm{v}^k_{\mathrm{u}}}) + \frac{1}{|\mathcal{V}|}\sum_{\mathrm{u}\in\mathcal{V}^k}(P^k_{\mathrm{u}}\{\bar{r}\}\cdot D^k_{\mathrm{u}}(p)). \tag{A–5}$$

Denote $V^k_i\{r,m\}$ the set of pixels in the k-th frame with the same XEP $P^k_i\{r,m\}$; denote $N^k_i\{r,m\}$ the number of pixels in $V^k_i\{r,m\}$; denote $N^k\{r,m\}$ the number of sets with different XEP $P^k_i\{r,m\}$ in the k-th frame.

We have

$$\frac{1}{|\mathcal{V}|}\sum_{\mathrm{u}\in\mathcal{V}^k}(P^k_{\mathrm{u}}\{r,m\}\cdot D^{k-1}_{\mathrm{u}+\check{\mathrm{m}}\mathrm{v}^k_{\mathrm{u}}}) = \frac{1}{|\mathcal{V}|}\sum_{i=1}^{N^k\{r,m\}}(P^k_i\{r,m\}\sum_{\mathrm{u}\in\mathcal{V}^k_i\{r,m\}}D^{k-1}_{\mathrm{u}+\check{\mathrm{m}}\mathrm{v}^k_{\mathrm{u}}}). \tag{A–6}$$

For large $N^k_i\{r,m\}$, we have $\frac{1}{N^k_i\{r,m\}}\sum_{\mathrm{u}\in\mathcal{V}^k_i\{r,m\}}D^{k-1}_{\mathrm{u}+\check{\mathrm{m}}\mathrm{v}^k_{\mathrm{u}}}$ converges to D^{k-1}, so the first term in the right-hand side in (A–5) is $D^{k-1}\cdot\bar{P}^k\{r,m\}$, where $\bar{P}^k\{r,m\} = \frac{1}{|\mathcal{V}|}\sum_{i=1}^{N^k\{r,m\}}(P^k_i\{r,m\}\cdot N^k_i\{r,m\})$.

Following the same process, we obtain the second term in the right-hand side in (A–5) as $D^{k-1}\cdot\bar{P}^k\{r,\bar{m}\}$, where $\bar{P}^k\{r,\bar{m}\} = \frac{1}{|\mathcal{V}|}\sum_{i=1}^{N^k\{r,\bar{m}\}}(P^k_i\{r,\bar{m}\}\cdot N^k_i\{r,\bar{m}\})$; and

$$\frac{1}{|\mathcal{V}|}\sum_{\mathrm{u}\in\mathcal{V}^k}(P^k_{\mathrm{u}}\{\bar{r}\}\cdot D^k_{\mathrm{u}}(p)) = \frac{1}{|\mathcal{V}|}\sum_{i=1}^{N^k\{\bar{r}\}}(P^k_i\{\bar{r}\}\sum_{\mathrm{u}\in\mathcal{V}^k_i\{\bar{r}\}}D^k_{\mathrm{u}}(p)). \tag{A–7}$$

For large $N^k_i\{\bar{r}\}$, we have $\frac{1}{N^k_i\{\bar{r}\}}\sum_{\mathrm{u}\in\mathcal{V}^k_i\{\bar{r}\}}D^k_{\mathrm{u}}(p)$ converges to $D^k(p)$, so the third term in the right-hand side in (A–5) is $D^k(p)\cdot(1-\bar{P}^k(r))$.

Note that $P^k_i\{r,m\}+P^k_i\{r,\bar{m}\} = P^k_i\{r\}$ and $N^k_i\{r,m\} = N^k_i\{r,\bar{m}\}$. So, we obtain

$$D^k(P) = D^{k-1}\cdot\bar{P}^k(r) + D^k(p)\cdot(1-\bar{P}^k(r)). \tag{A–8}$$

For P-MBs without slice data partitioning, it is straightforward to acquire (A–8) from (3–18). For I-MBs, from (2–37), it is also easy to obtain $D^k(P) = D^{k-1}\cdot\bar{P}^k(r)$. So, together with (A–8), we obtain (2–40). □

A.4 Proof of Lemma 3

Proof. Since $E[(\hat{f}^{k-1}_{\mathrm{u}+\check{\mathrm{m}}\mathrm{v}^k_{\mathrm{u}}})^2] = E[(\hat{f}^{k-1}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}})^2]$, we have $E[(\hat{f}^{k-1}_{\mathrm{u}+\check{\mathrm{m}}\mathrm{v}^k_{\mathrm{u}}})^2] = E[(\xi^k_{\mathrm{u}} + \hat{f}^{k-1}_{\mathrm{u}+\check{\mathrm{m}}\mathrm{v}^k_{\mathrm{u}}})^2]$ and therefore $E[\xi^k_{\mathrm{u}} \cdot \hat{f}^{k-1}_{\mathrm{u}+\check{\mathrm{m}}\mathrm{v}^k_{\mathrm{u}}}] = -\frac{E[(\xi^k_{\mathrm{u}})^2]}{2}$.

Following the same deriving process, we can prove $E[\xi^k_{\mathrm{u}} \cdot \hat{f}^{k-1}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}}] = \frac{E[(\xi^k_{\mathrm{u}})^2]}{2}$. □

A.5 Proof of Lemma 4

Proof. For P-MBs with slice data partioning, from (2–19) and (2–62) we obtain

$$D^k(P) = \frac{1}{|\mathcal{V}|}\sum_{\mathrm{u}\in\mathcal{V}^k}(P^k_{\mathrm{u}}\{r,m\}\cdot D^{k-1}_{\mathrm{u}+\check{\mathrm{m}}\mathrm{v}^k_{\mathrm{u}}}) + \frac{1}{|\mathcal{V}|}\sum_{\mathrm{u}\in\mathcal{V}^k}(P^k_{\mathrm{u}}\{r,\bar{m}\}\cdot D^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}}) + \frac{1}{|\mathcal{V}|}\sum_{\mathrm{u}\in\mathcal{V}^k}(P^k_{\mathrm{u}}\{\bar{r}\}\cdot D^k_{\mathrm{u}}(p)). \tag{A–9}$$

The first term in the right-hand side in (A–9) is exactly the same as the first term in the right-hand side in (A–5), that is, it equal to $D^{k-1}\cdot\bar{P}^k\{r,m\}$, where $\bar{P}^k\{r,m\} = \frac{1}{|\mathcal{V}|}\sum_{i=1}^{N^k\{r,m\}}(P^k_i\{r,m\}\cdot N^k_i\{r,m\})$.

Denote $V^k_i(j)\{r,\bar{m}\}$ the set of pixels using the same reference frame $k-j$ in the k-th frame with the same XEP $P^k_i(j)\{r,\bar{m}\}$; denote $N^k_i(j)\{r,\bar{m}\}$ the number of pixels in $V^k_i(j)\{r,\bar{m}\}$; denote $N^k(j)\{r,\bar{m}\}$ the number of sets with different XEP $P^k_i(j)\{r,\bar{m}\}$ but the same reference frame $k-j$ in the k-th frame.

We have

$$\frac{1}{|\mathcal{V}|}\sum_{\mathrm{u}\in\mathcal{V}^k}(P^k_{\mathrm{u}}\{r,\bar{m}\}\cdot D^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}}) = \frac{1}{|\mathcal{V}|}\sum_{i=1}^{N^k(j)\{r,\bar{m}\}}(P^k_i(j)\{r,\bar{m}\}\sum_{j=1}^{J}\sum_{\mathrm{u}\in\mathcal{V}^k_i(j)\{r,\bar{m}\}}D^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}}). \tag{A–10}$$

For large $N^k_i(j)\{r,\bar{m}\}$, we have $\frac{1}{N^k_i(j)\{r,\bar{m}\}}\sum_{\mathrm{u}\in\mathcal{V}^k_i(j)\{r,\bar{m}\}}D^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}}$ converges to D^{k-j}, so (A–10) becomes

$$\frac{1}{|\mathcal{V}|}\sum_{\mathrm{u}\in\mathcal{V}^k}(P^k_{\mathrm{u}}\{r,\bar{m}\}\cdot D^{k-j}_{\mathrm{u}+\mathrm{mv}^k_{\mathrm{u}}}) = \frac{1}{|\mathcal{V}|}\sum_{i=1}^{N^k(j)\{r,\bar{m}\}}(P^k_i(j)\{r,\bar{m}\}\sum_{j=1}^{J}N^k_i(j)\{r,\bar{m}\}\cdot D^{k-j}). \tag{A–11}$$

Similar to the definition in (2–29), we define the weighted average over joint PEPs, of event that residual is received with error and MV is received without error, for the set of

pixels using the same reference frame $k-j$ in the k-th frame as

$$\bar{P}^k(j)\{r,\bar{m}\} \triangleq \frac{1}{|\mathcal{V}^k(j)|}\sum_{i=1}^{N^k(j)\{r,\bar{m}\}}(P_i^k(j)\{r,\bar{m}\}\cdot N_i^k(j)\{r,\bar{m}\}). \tag{A–12}$$

We have

$$\begin{aligned}\frac{1}{|\mathcal{V}|}\sum_{\mathrm{u}\in\mathcal{V}^k}(P_{\mathrm{u}}^k\{r,\bar{m}\}\cdot D_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-j}) &= \frac{1}{|\mathcal{V}|}\sum_{j=1}^{J}(\bar{P}^k(j)\{r,\bar{m}\}\cdot|\mathcal{V}^k(j)|\cdot D^{k-j})\\ &= \sum_{j=1}^{J}(\bar{P}^k(j)\{r,\bar{m}\}\cdot w^k(j)\cdot D^{k-j}).\end{aligned} \tag{A–13}$$

Following the same process, we obtain

$$\frac{1}{|\mathcal{V}|}\sum_{\mathrm{u}\in\mathcal{V}^k}(P_{\mathrm{u}}^k\{\bar{r}\}\cdot D_{\mathrm{u}}^k(p)) = \sum_{j=1}^{J}(\bar{P}^k(j)\{\bar{r}\}\cdot w^k(j)\cdot\alpha^k(j)\cdot D^{k-j}), \tag{A–14}$$

where

$$\bar{P}^k(j)\{\bar{r}\} \triangleq \frac{1}{|\mathcal{V}^k(j)|}\sum_{i=1}^{N^k(j)\{\bar{r}\}}(P_i^k(j)\{\bar{r}\}\cdot N_i^k(j)\{\bar{r}\}) \tag{A–15}$$

is the weighted average over joint PEPs, of event that residual is received without error, for the set of pixels using the same reference frame $k-j$ in the k-th frame.

Therefore, we obtain

$$D^k(P) = D^{k-1}\cdot\bar{P}^k\{r,m\} + \sum_{j=1}^{J}(\bar{P}^k(j)\{r,\bar{m}\}\cdot w^k(j)\cdot D^{k-j}) + \sum_{j=1}^{J}(\bar{P}^k(j)\{\bar{r}\}\cdot w^k(j)\cdot\alpha^k(j)\cdot D^{k-j}). \tag{A–16}$$

For P-MBs without slice data partitioning, $\bar{P}^k\{r,m\} = \bar{P}^k\{r\}$ and $\bar{P}^k(j)\{r,\bar{m}\} = 0$, therefore we have

$$D^k(P) = D^{k-1}\cdot\bar{P}^k\{r\} + \sum_{j=1}^{J}(\bar{P}^k(j)\{\bar{r}\}\cdot w^k(j)\cdot\alpha^k(j)\cdot D^{k-j}). \tag{A–17}$$

For I-MBs, from (2–37), it is also easy to obtain $D^k(P) = D^{k-1}\cdot\bar{P}^k(r)$. So, together with (A–16), we obtain (2–65). □

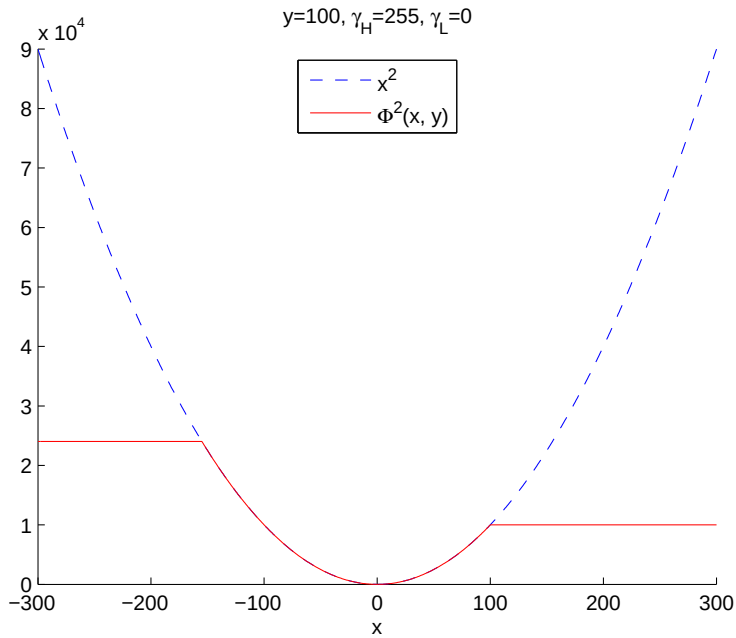


Figure A-1. Comparison of $\Phi^2(x, y)$ and x^2.

A.6 Lemma 5 and Its Proof

To prove Proposition 2, we need to use the following lemma.

Lemma 5. *The error reduction function* $\Phi(x,\ y)$ *satisfies* $\Phi^2(x,\ y) \leq x^2$ *for any* $\gamma_L \leq y \leq \gamma_H$.

Proof. From the definition in (3–10), we obtain

$$\Phi^2(x,\ y) - x^2 = \begin{cases} (y-\gamma_L)^2 - x^2, & x > y - \gamma_L \\ 0, & y - \gamma_H \leq x \leq y - \gamma_L \\ (y-\gamma_H)^2 - x^2, & x < y - \gamma_H. \end{cases} \tag{A–18}$$

Since $y \geq \gamma_L$, we obtain $(y-\gamma_L)^2 < x^2$ when $x > y - \gamma_L$. Similarly, since $y \leq \gamma_H$, we obtain $(y-\gamma_H)^2 < x^2$ when $x < y - \gamma_H$. Therefore $\Phi^2(x,\ y) - x^2 \leq 0$ for $\gamma_L \leq y \leq \gamma_H$, which is also shown in Fig. A-1. □

A.7 Lemma 6 and Its Proof

Before presenting the proof, we first give the definition of Ideal Codec.

Definition 1. *Ideal Codec: both the true MV and concealed MV are within the search range, and the position pointed by the true MV, i.e.,* $\mathbf{u} + \mathbf{mv}_{\mathrm{u}}^k$*, is the best reference pixel, under the MMSE criteria, for* $\hat{f}_{\mathrm{u}}^k$ *within the whole search range* $\mathcal{V}_{SR}^{k-1}$*, that is,* $\mathbf{v} = \underset{\mathrm{v} \in \mathrm{V}_{SR}^{k-1}}{\arg\min} \ \{(\hat{f}_{\mathrm{u}}^k - \hat{f}_{\mathrm{v}}^{k-1})^2\}$.

To prove Corollary 1, we need to use the following lemma.

Lemma 6. *In an ideal codec,* $\widetilde{\Delta}_{\mathrm{u}}^k\{\bar{p}\} = 0$*, In other words, if there is no propagated error, the clipping noise for the pixel* $\mathbf{u}^k$ *at the decoder is always zero no matter what kind of error event occurs in the* k*-th frame.*

Proof. In an ideal codec, we have $(\hat{e}_{\mathrm{u}}^k)^2 = (\hat{f}_{\mathrm{u}}^k - \hat{f}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-1})^2 \leq (\hat{f}_{\mathrm{u}}^k - \hat{f}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1})^2$. Due to the spatial and temporal continuity of the natural video, we can prove by contradiction that in an ideal codec $\hat{f}_{\mathrm{u}}^k - \hat{f}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-1}$ and $\hat{f}_{\mathrm{u}}^k - \hat{f}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1}$ have the same sign, that is either

$$\hat{f}_{\mathrm{u}}^k - \hat{f}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1} \geq \hat{e}_{\mathrm{u}}^k \geq 0, \quad \text{or} \quad \hat{f}_{\mathrm{u}}^k - \hat{f}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1} \leq \hat{e}_{\mathrm{u}}^k \leq 0. \tag{A–19}$$

If the sign of $\hat{f}_{\mathrm{u}}^k - \hat{f}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-1}$ and $\hat{f}_{\mathrm{u}}^k - \hat{f}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1}$ is not the same, then due to the spatial and temporal continuity of the input video, there exists a better position $\mathbf{v} \in \mathcal{V}^{k-1}$ between $\mathbf{mv}_{\mathrm{u}}^k$ and $\check{\mathbf{mv}}_{\mathrm{u}}^k$, and therefore within the search range, so that $(\hat{e}_{\mathrm{u}}^k)^2 \geq (\hat{f}_{\mathrm{u}}^k - \hat{f}_{\mathrm{v}}^{k-1})^2$. In this case, encoder will choose $\mathbf{v}$ as the best reference pixel within the search range. This contradicts the assumption that the best reference pixel is $\mathbf{u} + \mathbf{mv}_{\mathrm{u}}^k$ within the search range.

Therefore, from (A–19), we obtain

$$\hat{f}_{\mathrm{u}}^k \geq \hat{f}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1} + \hat{e}_{\mathrm{u}}^k \geq \hat{f}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1}, \quad \text{or} \quad \hat{f}_{\mathrm{u}}^k \leq \hat{f}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1} + \hat{e}_{\mathrm{u}}^k \leq \hat{f}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1}. \tag{A–20}$$

Since both $\hat{f}_{\mathrm{u}}^k$ and $\hat{f}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1}$ are reconstructed pixel value, they are within the range $\gamma_H \geq \hat{f}_{\mathrm{u}}^k, \hat{f}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1} \geq \gamma_L$. From (A–20), we have $\gamma_H \geq \hat{f}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1} + \hat{e}_{\mathrm{u}}^k \geq \gamma_L$, and thus $\Gamma(\hat{f}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1} + \hat{e}_{\mathrm{u}}^k) = \hat{f}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1} + \hat{e}_{\mathrm{u}}^k$. As a result, we obtain $\widetilde{\Delta}_{\mathrm{u}}^k\{\bar{r}, m, \bar{p}\} = (\hat{f}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1} + \hat{e}_{\mathrm{u}}^k) - \Gamma(\hat{f}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1} + \hat{e}_{\mathrm{u}}^k) = 0$.

Since $\widetilde{\Delta}_{\mathrm{u}}^k\{\bar{r}, \bar{m}, \bar{p}\} = \hat{\Delta}_{\mathrm{u}}^k = 0$, and from Section 2.3.4.1, we know that $\widetilde{\Delta}_{\mathrm{u}}^k\{r, \bar{p}\} = 0$, hence we obtain $\widetilde{\Delta}_{\mathrm{u}}^k\{\bar{p}\} = 0$. □

Remark 1. *Note that Lemma 6 is proved under the assumption of pixel-level motion estimation. In a practical encoder, block-level motion estimation is adopted with the criterion of minimizing the MSE of the whole block, e.g., in H.263, or minimizing the cost of residual bits and MV bits, e.g., in H.264. Therefore, some reference pixels in the block may not be the best reference pixel within the search range. On the other hand, Rate Distortion Optimization (RDO) as used in H.264 may also cause some reference pixels not to be the best reference pixels. However, the experiment results for all the test video sequences show that the probability of* $\widetilde{\Delta}_{\mathrm{u}}^{k}\{\bar{r}, m, \bar{p}\} \neq 0$ *is negligible.*

A.8 Proof of Corollary 1

Proof. From (A–1), we obtain $\widetilde{\Delta}_{\mathrm{u}}^{k}\{\bar{p}\} = (\hat{f}_{\mathrm{u}}^{k} - \widetilde{\xi}_{\mathrm{u}}^{k} - \widetilde{\varepsilon}_{\mathrm{u}}^{k}) - \Gamma(\hat{f}_{\mathrm{u}}^{k} - \widetilde{\xi}_{\mathrm{u}}^{k} - \widetilde{\varepsilon}_{\mathrm{u}}^{k})$. Together with Lemma 6, which is presented and proved in Appendix A.7, we have $\gamma_L \leq \hat{f}_{\mathrm{u}}^{k} - \widetilde{\xi}_{\mathrm{u}}^{k} - \widetilde{\varepsilon}_{\mathrm{u}}^{k} \leq \gamma_H$. From Lemma 5 in Appendix A.6, we have $\Phi^2(x, y) \leq x^2$ for any $\gamma_L \leq y \leq \gamma_H$; together with (A–2), it is straightforward to prove that $E[(\widetilde{\zeta}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1} + \widetilde{\Delta}_{\mathrm{u}}^{k})^2] \leq E[(\widetilde{\zeta}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1})^2]$. By expanding $E[(\widetilde{\zeta}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1} + \widetilde{\Delta}_{\mathrm{u}}^{k})^2]$, we obtain

$$E[\widetilde{\zeta}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1} \cdot \widetilde{\Delta}_{\mathrm{u}}^{k}] \leq -\frac{1}{2}E[(\widetilde{\Delta}_{\mathrm{u}}^{k})^2] \leq 0. \tag{A–21}$$

The physical meaning of (A–21) is that $\widetilde{\zeta}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1}$ and $\widetilde{\Delta}_{\mathrm{u}}^{k}$ are negatively correlated if $\widetilde{\Delta}_{\mathrm{u}}^{k} \neq 0$. Since $\widetilde{\Delta}_{\mathrm{u}}^{k}\{r\} = 0$ as noted in Section 2.3.4.1 and $\widetilde{\Delta}_{\mathrm{u}}^{k}\{\bar{p}\} = 0$ as proved in Lemma 6, we know that $\widetilde{\Delta}_{\mathrm{u}}^{k} \neq 0$ is valid only for the error events $\{\bar{r}, m, p\}$ and $\{\bar{r}, \bar{m}, p\}$, and $\widetilde{\Delta}_{\mathrm{u}}^{k} = 0$ for any other error event. In other words, $\widetilde{\zeta}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1}$ and $\widetilde{\Delta}_{\mathrm{u}}^{k}$ are negatively correlated under the condition $\{\bar{r}, p\}$, and they are uncorrelated under other conditions. □

APPENDIX B
PROOFS IN CHAPTER 3

B.1 Proof of Proposition 3

Proof. From Chapter 2, we know that $\widetilde{\Delta}_{\mathrm{u}}^{k} \triangleq (\widetilde{f}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-j'} + \widetilde{e}_{\mathrm{u}}^{k}) - \Gamma(\widetilde{f}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-j'} + \widetilde{e}_{\mathrm{u}}^{k})$. If there is no newly induced error, that is, $\widetilde{e}_{\mathrm{u}}^{k} = \hat{e}_{\mathrm{u}}^{k}$ and $\widetilde{\mathbf{mv}}_{\mathrm{u}}^{k} = \mathbf{mv}_{\mathrm{u}}^{k}$, we have $\widetilde{f}_{\mathrm{u}+\widetilde{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-j'} + \widetilde{e}_{\mathrm{u}}^{k} = \widetilde{f}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-j} + \hat{e}_{\mathrm{u}}^{k} = \hat{f}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-j} - \widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-j} + \hat{e}_{\mathrm{u}}^{k} = \hat{f}_{\mathrm{u}}^{k} - \widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-j}$. Therefore, we have

$$\widetilde{\Delta}_{\mathrm{u}}^{k}\{\bar{r}, \bar{m}\} = \begin{cases} \hat{f}_{\mathrm{u}}^{k} - \widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-j} - 255, & \hat{f}_{\mathrm{u}}^{k} - \widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-j} > 255 \\ \hat{f}_{\mathrm{u}}^{k} - \widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-j}, & \hat{f}_{\mathrm{u}}^{k} - \widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-j} < 0 \\ 0, & \text{otherwise.} \end{cases} \tag{B–1}$$

Adding $\widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^{k}}^{k-j}$ to the left-hand side and right-hand side in (B–1), we obtain (3–23). □

B.2 Proof of Theorem 3.1

Proof. Since there is no slice data partitioning, $D_{\mathrm{u},ETE}^{k} = (1 - P_{\mathrm{u}}^{k}) \cdot D_{\mathrm{u},ETE}^{k}\{\bar{r}, \bar{m}\} + P_{\mathrm{u}}^{k} \cdot D_{\mathrm{u},ETE}^{k}\{r, m\}$.

First, if the packet is lost, from (4–2) we obtain

$$\begin{aligned} D_{\mathrm{u}}^{k}\{r, m\} &= (\varepsilon_{\mathrm{u}}^{k} + \xi_{\mathrm{u}}^{k})^2 + 2(\varepsilon_{\mathrm{u}}^{k} + \xi_{\mathrm{u}}^{k}) \cdot E[\widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1}] + D_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1} \\ &= (\varepsilon_{\mathrm{u}}^{k} + \xi_{\mathrm{u}}^{k} + E[\widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1}])^2 + \sigma_{\widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1}}^{2}, \end{aligned} \tag{B–2}$$

and from (4–1) we obtain

$$E[\widetilde{\zeta}_{\mathrm{u}}^{k}]\{r, m\} = \varepsilon_{\mathrm{u}}^{k} + \xi_{\mathrm{u}}^{k} + E[\widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1}]. \tag{B–3}$$

Together with (B–2), (B–3) and (3–24), we obtain the end-to-end distortion for the case where the packet is lost as below

$$D_{\mathrm{u},ETE}^{k}\{r, m\} = (f_{\mathrm{u}}^{k} - \hat{f}_{\mathrm{u}}^{k} + \varepsilon_{\mathrm{u}}^{k} + \xi_{\mathrm{u}}^{k} + E[\widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1}])^2 + \sigma_{\widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^{k}}^{k-1}}^{2}. \tag{B–4}$$

By definition, we have $\varepsilon_{\mathrm{u}}^k = \hat{e}_{\mathrm{u}}^k - \check{e}_{\mathrm{u}}^k$ and $\xi_{\mathrm{u}}^k = \hat{f}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-j} - \hat{f}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1}$. So, we obtain $\varepsilon_{\mathrm{u}}^k + \xi_{\mathrm{u}}^k = \hat{f}_{\mathrm{u}}^k - \hat{f}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1} - \check{e}_{\mathrm{u}}^k$, and

$$D_{\mathrm{u},ETE}^k\{r,m\} = (f_{\mathrm{u}}^k - \hat{f}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1} - \check{e}_{\mathrm{u}}^k + E[\widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1}])^2 + \sigma_{\widetilde{\zeta}_{\mathrm{u}+\check{\mathrm{mv}}_{\mathrm{u}}^k}^{k-1}}^2 . \tag{B–5}$$

Note that if the error concealment scheme is to copy the reconstructed pixel value from the previous frame, we have $\varepsilon_{\mathrm{u}}^k + \xi_{\mathrm{u}}^k = \hat{f}_{\mathrm{u}}^k - \hat{f}_{\mathrm{u}}^{k-1}$.

Note that the error concealment method is the same for intra mode and inter mode since there is no mode information for decoder if the packet is received in error; hence $\check{\mathbf{mv}}_{\mathrm{u}}^k$ and $\check{e}_{\mathrm{u}}^k$ in (B–5) are the same for both intra mode and inter mode. On the other hand, the value of f_{u}^k is known before the mode decision and all other variables in (B–5) come from the previous frame. Therefore, the resulting end-to-end distortion in this case, i.e., $D_{\mathrm{u},ETE}^k\{r,m\}$ will also be the same for both intra mode and inter mode.

Second, if the packet is correctly received, from (4–2) we obtain $D_{\mathrm{u}}^k\{\bar{r},\bar{m}\} = D_{\mathrm{u}}^k(p)$ and from (4–1) we obtain $E[\widetilde{\zeta}_{\mathrm{u}}^k]\{r,m\} = E[\widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-j} + \widetilde{\Delta}_{\mathrm{u}}^k\{\bar{r},\bar{m}\}]$. From (3–24), we obtain the end-to-end distortion as

$$D_{\mathrm{u},ETE}^k\{\bar{r},\bar{m}\} = (f_{\mathrm{u}}^k - \hat{f}_{\mathrm{u}}^k)^2 + D_{\mathrm{u}}^k(p) + 2(f_{\mathrm{u}}^k - \hat{f}_{\mathrm{u}}^k)\cdot E[\widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-j} + \widetilde{\Delta}_{\mathrm{u}}^k\{\bar{r},\bar{m}\}]. \tag{B–6}$$

Since both P_{u}^k and $D_{\mathrm{u},ETE}^k\{r,m\}$ are the same for all modes, we can denote $P_{\mathrm{u}}^k \cdot D_{\mathrm{u},ETE}^k\{r,m\}$ by C_{u}^k, which is independent of all modes. Let $D_{\mathrm{u},\overline{ETE}}^k = (1-P_{\mathrm{u}}^k)\cdot D_{\mathrm{u},ETE}^k\{\bar{r},\bar{m}\}$; then we have

$$D_{\mathrm{u},ETE}^k(\omega_m) = D_{\mathrm{u},\overline{ETE}}^k(\omega_m) + C_{\mathrm{u}}^k, \tag{B–7}$$

where

$$\begin{aligned} D_{\mathrm{u},\overline{ETE}}^k(\omega_m) &= (1-P_{\mathrm{u}}^k)\cdot D_{\mathrm{u},ETE}^k\{\bar{r},\bar{m}\} && \text{(B–8)} \\ &= (1-P_{\mathrm{u}}^k)\cdot\{(f_{\mathrm{u}}^k - \hat{f}_{\mathrm{u}}^k)^2 + D_{\mathrm{u}}^k(p) + 2(f_{\mathrm{u}}^k - \hat{f}_{\mathrm{u}}^k)\cdot E[\widetilde{\zeta}_{\mathrm{u}+\mathrm{mv}_{\mathrm{u}}^k}^{k-j} + \widetilde{\Delta}_{\mathrm{u}}^k\{\bar{r},\bar{m}\}]\}. && \text{(B–9)} \end{aligned}$$

□

B.3 Proof of Proposition 4

Proof. From (3–28), we have

$$\begin{aligned}
\arg\min_{\omega_m}\{\hat{D}^k_{ETE}(\omega_m) + \lambda \cdot R(\omega_m)\} &= \arg\min_{\omega_m}\{\sum_{\mathrm{u}\in\mathcal{V}_i^k}[D^k_{\mathrm{u},\overline{ETE}}(\omega_m) + C^k_{\mathrm{u}}] + \lambda \cdot R(\omega_m)\} \\
&= \arg\min_{\omega_m}\{\sum_{\mathrm{u}\in\mathcal{V}_i^k} D^k_{\mathrm{u},\overline{ETE}}(\omega_m) + \sum_{\mathrm{u}\in\mathcal{V}_i^k} C^k_{\mathrm{u}} + \lambda \cdot R(\omega_m)\} \\
&= \arg\min_{\omega_m}\{\hat{D}^k_{\overline{ETE}}(\omega_m) + \lambda \cdot R(\omega_m)\} \\
&= \hat{\omega}_m.
\end{aligned} \tag{B–10}$$

This is, Algorithm A and Algorithm 2 produce the same solution. □

APPENDIX C
PROOFS IN CHAPTER 4

Proof. If $\sum_{i=1}^{N} w_i \neq 0$, let us define $S \triangleq \sum_{i=1}^{N} w_i$ and $p_i \triangleq \frac{w_i}{S}$. Therefore, we have $w_i = p_i \cdot S$, $\sum_{i=1}^{N} p_i = 1$ and $E[(\sum_{i=1}^{N} w_i \cdot X_i)^2] = E[(\sum_{i=1}^{N} p_i \cdot S \cdot X_i)^2] = S^2 \cdot E[(\sum_{i=1}^{N} p_i \cdot X_i)^2]$.

We further define $D \triangleq E[(\sum_{i=1}^{N} p_i \cdot X_i)^2]$. Therefore we have

$$E[(\sum_{i=1}^{N} w_i \cdot X_i)^2] = S^2 \cdot D, \tag{C–1}$$

where D can be calculated by

$$\begin{aligned}
D &= \sum_{j=1}^{N}[p_j^2 \cdot E(X_j^2)] + \sum_{k=1}^{N}\sum_{\substack{l=1\\(l\neq k)}}^{N}[p_k \cdot p_l \cdot E(X_k \cdot X_l)] \\
&= \sum_{j=1}^{N}[p_j \cdot E(X_j^2)] - \sum_{j=1}^{N}[p_j \cdot (1-p_j) \cdot E(X_j^2)] + \sum_{k=1}^{N}\sum_{\substack{l=1\\(l\neq k)}}^{N}[p_k \cdot p_l \cdot E(X_k \cdot X_l)] \\
&= \sum_{j=1}^{N}[p_j \cdot E(X_j^2)] - \sum_{j=1}^{N}[p_j \cdot \sum_{\substack{j'=1\\(j'\neq j)}}^{N} p_{j'} \cdot E(X_j^2)] + \sum_{k=1}^{N}\sum_{\substack{l=1\\(l\neq k)}}^{N}[p_k \cdot p_l \cdot E(X_k \cdot X_l)] \\
&= \sum_{j=1}^{N}[p_j \cdot E(X_j^2)] - \sum_{j=1}^{N}\sum_{\substack{j'=1\\(j'\neq j)}}^{N}[p_j \cdot p_{j'} \cdot E(X_j^2)] + \sum_{k=1}^{N}\sum_{\substack{l=1\\(l\neq k)}}^{N}[p_k \cdot p_l \cdot E(X_k \cdot X_l)] \\
&= \sum_{j=1}^{N}[p_j \cdot E(X_j^2)] - \sum_{k=1}^{N-1}\sum_{l=k+1}^{N}\{p_k \cdot p_l \cdot [E(X_k^2) + E(X_l^2)]\} + \sum_{k=1}^{N-1}\sum_{l=k+1}^{N}[2 \cdot p_k \cdot p_l \cdot E(X_k \cdot X_l)] \\
&= \sum_{j=1}^{N}[p_j \cdot E(X_j^2)] - \sum_{k=1}^{N-1}\sum_{l=k+1}^{N}[p_k \cdot p_l \cdot E(X_k - X_l)^2].
\end{aligned} \tag{C–2}$$

From (C–1) and (C–2), we have

$$\begin{aligned}
E[(\sum_{i=1}^{N} w_i \cdot X_i)^2] &= S \cdot \sum_{j=1}^{N}[w_j \cdot E(X_j^2)] - \sum_{k=1}^{N-1}\sum_{l=k+1}^{N}[w_k \cdot w_l \cdot E(X_k - X_l)^2] \\
&= \sum_{i=1}^{N} w_i \cdot \sum_{j=1}^{N}[w_j \cdot E(X_j^2)] - \sum_{k=1}^{N-1}\sum_{l=k+1}^{N}[w_k \cdot w_l \cdot E(X_k - X_l)^2].
\end{aligned} \tag{C–3}$$

If $\sum_{i=1}^{N} w_i = 0$, we have

$$w_j = -\sum_{\substack{j'=1 \\ (j' \neq j)}}^{N} w_{j'}, \tag{C–4}$$

and

$$\begin{aligned} E[(\sum_{i=1}^{N} w_i \cdot X_i)^2] &= \sum_{j=1}^{N} [w_j^2 \cdot E(X_j^2)] + \sum_{k=1}^{N} \sum_{\substack{l=1 \\ (l \neq k)}}^{N} [w_k \cdot w_l \cdot E(X_k \cdot X_l)] \\ &= -\sum_{j=1}^{N} [w_j \cdot \sum_{\substack{j'=1 \\ (j' \neq j)}}^{N} w_{j'} \cdot E(X_j^2)] + \sum_{k=1}^{N} \sum_{\substack{l=1 \\ (l \neq k)}}^{N} [w_k \cdot w_l \cdot E(X_k \cdot X_l)] \\ &= -\sum_{k=1}^{N-1} \sum_{l=k+1}^{N} [w_k \cdot w_l \cdot E(X_k - X_l)^2]. \end{aligned} \tag{C–5}$$

We see that (C–5) is just a special case of (C–3) under $\sum_{i=1}^{N} w_i = 0$. Therefore, for any $w_i \in \Re$, we have the general form (C–3). □

APPENDIX D
PROOFS IN CHAPTER 5

D.1 Proof of Equation (5–5)

Proof. For transform coefficients with i.i.d. zero-mean Laplacian distribution, the probability density function (pdf) is $p(x) = \frac{1}{\sqrt{2}\cdot\sigma} \cdot e^{\frac{\sqrt{2}\cdot|x|}{\sigma}}$, where σ is the standard deviation. For the uniform quantizer with quantization step size Q and quantization offset θ_2, the probability of zero after quantization is

$$P_0 = 2\int_0^{Q\cdot(1-\theta_2)} p(x)dx = 1 - e^{-\theta_1\cdot(1-\theta_2)}, \tag{D–1}$$

and the probability of level n after quantization is

$$P_n = \int_{Q\cdot(n-\theta_2)}^{Q\cdot(n+1-\theta_2)} p(x)dx = \frac{1}{2}(1 - e^{-\theta_1}) \cdot e^{\theta_1\cdot\theta_2} \cdot e^{-\theta_1\cdot n}, \tag{D–2}$$

where $\theta_1 = \frac{\sqrt{2}\cdot Q}{\sigma}$.

As a result,

$$\begin{aligned} H &= -P_0 \cdot \log_2 P_0 - 2\sum_{n=1}^{\infty} P_n \cdot \log_2 P_n \\ &= -P_0 \cdot \log_2 P_0 + (1 - P_0) \cdot (\frac{\theta_1 \cdot \log_2 e}{1 - e^{-\theta_1}} - \log_2(1 - e^{-\theta_1}) - \theta_1 \cdot \theta_2 \cdot \log_2 e + 1). \end{aligned} \tag{D–3}$$

□

D.2 Calculation of Entropy for Different Quantized Transform Coefficients

Proof. For a 4x4 integer transform with average variance σ^2, the variance for each transform coefficient can be calculate by (5–7) as

$$\sigma^2 = \frac{1}{16}\sum_{x=0}^{4}\sum_{y=0}^{4}\sigma^2_{(x,y)} = \frac{225}{1024} \cdot \sigma_0^2. \tag{D–4}$$

Therefore, we have

$$\sigma^2_{(x,y)} = 2^{-(x+y)} \cdot \frac{1024}{225} \cdot \sigma^2. \tag{D–5}$$

□

D.3 Proof of Proposition 5

Proof. In a Rayleigh fading channel, the received signal amplitude has the Rayleigh distribution, and received signal power has the exponential distribution. Therefore, SNR in receiver has the exponential distribution [18], that is,

$$P_\gamma(\gamma) = \frac{1}{\overline{\gamma}} e^{-\frac{\gamma}{\overline{\gamma}}}, \tag{D–6}$$

where $\overline{\gamma} = \frac{P_t \overline{g}}{N_0 B}$; P_t is transmission power; $\overline{g}$ is the mean of channel gain; $\frac{N_0}{2}$ is noise power spectral density; and B is passband bandwidth.

By using the well-known upper bound as approximation for Q function, i.e., $Q(x) \approx \frac{1}{2} \cdot e^{-\frac{x^2}{2}}$ [64], from (5–20) we have

$$\begin{aligned} E_\gamma[PEP] &= \int_0^\infty PEP(\gamma) P_\gamma(\gamma) d\gamma \\ &\approx \int_0^{\gamma_{th}} \frac{1}{\overline{\gamma}} e^{-\frac{\gamma}{\overline{\gamma}}} d\gamma + \int_{\gamma_{th}}^\infty \Big(\sum_{d=d_{free}}^{d_{max}} \frac{1}{2} L W_d e^{-\gamma d} \Big) \frac{1}{\overline{\gamma}} e^{-\frac{\gamma}{\overline{\gamma}}} d\gamma \\ &= 1 - e^{-\frac{\gamma_{th}}{\overline{\gamma}}} + \sum_{d=d_{free}}^{d_{max}} \frac{1}{2} L W_d \int_{\gamma_{th}}^\infty \frac{1}{\overline{\gamma}} e^{-\gamma d - \frac{\gamma}{\overline{\gamma}}} d\gamma \\ &= 1 - e^{-\frac{\gamma_{th}}{\overline{\gamma}}} + \sum_{d=d_{free}}^{d_{max}} \frac{1}{2} L W_d e^{-d\gamma_{th}} \frac{1}{1 + d\overline{\gamma}} e^{-\frac{\gamma_{th}}{\overline{\gamma}}}, \end{aligned} \tag{D–7}$$

where γ_{th} is defined by (5–21).

let $f(d) = \frac{1}{2} L W_d e^{-d\gamma_{th}}$ and $PEP_{th} = 1$, from (5–21) we have $\sum_{d=d_{free}}^{d_{max}} f(d) = 1$. If we regard $f(d)$ as a pmf for d and further let $g(d) = \frac{1}{1+d\overline{\gamma}} e^{-\frac{\gamma_{th}}{\overline{\gamma}}}$, the third term in (D–7) can be regarded as a expected value of $g(d)$ with pmf $f(d)$. Since $f(d)$ decays exponentially with the increase of d, $g(d)$ can be approximated by a close upper bound $\frac{1}{1+d_{free}\overline{\gamma}} e^{-\frac{\gamma_{th}}{\overline{\gamma}}}$. Therefore, (D–7) becomes

$$\begin{aligned} E_\gamma[PEP] &\approx 1 - e^{-\frac{\gamma_{th}}{\overline{\gamma}}} + \frac{e^{-\frac{\gamma_{th}}{\overline{\gamma}}}}{1 + d_{free}\overline{\gamma}} \sum_{d=d_{free}}^{d_{max}} \frac{1}{2} L W_d \cdot e^{-d\gamma_{th}} \\ &= 1 - e^{-\frac{\gamma_{th}}{\overline{\gamma}}} + \frac{e^{-\frac{\gamma_{th}}{\overline{\gamma}}}}{1 + d_{free}\overline{\gamma}}, \end{aligned} \tag{D–8}$$

In a practical communication system, $d_{free}\overline{\gamma} >> 1$. On the other hand, since $\overline{\gamma} >> \gamma_{th}$ as mentioned in Section 5.3.3.2 and $e^{-x} \approx 1 - xe^{-x}$ at small x, we may approximate $1 - e^{-\frac{\gamma_{th}}{\overline{\gamma}}}$ by $\frac{\gamma_{th}}{\overline{\gamma}} e^{-\frac{\gamma_{th}}{\overline{\gamma}}}$. Therefore, we have

$$\begin{aligned} E_{\gamma}[PEP] &\approx \frac{\gamma_{th}}{\overline{\gamma}} e^{-\frac{\gamma_{th}}{\overline{\gamma}}} + \frac{e^{-\frac{\gamma_{th}}{\overline{\gamma}}}}{d_{free}\overline{\gamma}} \\ &= \frac{\gamma_{th}}{\overline{\gamma}} e^{-\frac{\gamma_{th}}{\overline{\gamma}}} (1 + \frac{1}{d_{free}\gamma_{th}}) \end{aligned} \tag{D–9}$$

Note that xe^{-x} increases while x increases in the interval $0 < x < 1$. Therefore, $E_{\gamma}[PEP]$ decreases while $\overline{\gamma}$ increases. □

REFERENCES

[1] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, 623–656, 1948.

[2] ——, "Coding theorems for a discrete source with a fidelity criterion," *IRE Nat. Conv. Rec. Part*, vol. 4, pp. 142–163, 1959.

[3] T. Berger and J. Gibson, "Lossy source coding," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2693–2723, 1998.

[4] R. Zhang, S. L. Regunathan, and K. Rose, "Video coding with optimal inter/intra-mode switching for packet loss resilience," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 966–976, Jun. 2000.

[5] T. Stockhammer, M. Hannuksela, and T. Wiegand, "H. 264/AVC in wireless environments," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 657–673, 2003.

[6] T. Stockhammer, T. Wiegand, and S. Wenger, "Optimized transmission of h.26l/jvt coded video over packet-lossy networks," in *IEEE ICIP*, 2002.

[7] M. Sabir, R. Heath, and A. Bovik, "Joint source-channel distortion modeling for MPEG-4 video," *IEEE Transactions on Image Processing*, vol. 18, no. 1, pp. 90–105, 2009.

[8] K. Stuhlmuller, N. Farber, M. Link, and B. Girod, "Analysis of video transmission over lossy channels," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 1012–1032, Jun. 2000.

[9] J. U. Dani, Z. He, and H. Xiong, "Transmission distortion modeling for wireless video communication," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM'05)*, 2005.

[10] Z. He, J. Cai, and C. W. Chen, "Joint source channel rate-distortion analysis for adaptive mode selection and rate control in wireless video coding," *IEEE Transactions on Circuits and System for Video Technology, special issue on wireless video*, vol. 12, pp. 511–523, Jun. 2002.

[11] Y. Wang, Z. Wu, and J. M. Boyce, "Modeling of transmission-loss-induced distortion in decoded video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 6, pp. 716–732, Jun. 2006.

[12] J. Chakareski, J. Apostolopoulos, W.-T. Tan, S. Wee, and B. Girod, "Distortion chains for predicting the video distortion for general packet loss patterns," in *Proc. ICASSP*, 2004.

[13] J. Chakareski, J. Apostolopoulos, S. Wee, W.-T. Tan, and B. Girod, “Rate-distortion hint tracks for adaptive video streaming,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 10, pp. 1257–1269, 2005.

[14] C. Zhang, H. Yang, S. Yu, and X. Yang, “GOP-level transmission distortion modeling for mobile streaming video,” *Signal Processing: Image Communication*, 2007.

[15] M. T. Ivrlač, L. U. Choi, E. Steinbach, and J. A. Nossek, “Models and analysis of streaming video transmission over wireless fading channels,” *Signal Processing: Image Communication*, vol. 24, no. 8, pp. 651–665, Sep. 2009.

[16] Y. J. Liang, J. G. Apostolopoulos, and B. Girod, “Analysis of packet loss for compressed video: Effect of burst losses and correlation between error frames,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 7, pp. 861–874, Jul. 2008.

[17] H. Yang and K. Rose, “Advances in recursive per-pixel end-to-end distortion estimation for robust video coding in H. 264/AVC,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 7, p. 845, 2007.

[18] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.

[19] S. Wenger, “H.264/AVC over IP,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 645– 656, Jul. 2003.

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

[21] D. Agrafiotis, D. R. Bull, and C. N. Canagarajah, “Enhanced error concealment with mode selection,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 8, pp. 960–973, Aug. 2006.

[22] *ITU-T Series H: Audiovidual and Multimedia Systems, Advanced video coding for generic audiovisual services*, Nov. 2007.

[23] 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 Transactions on Multimedia*, vol. 9, no. 3, pp. 445–454, Apr. 2007.

[24] B. Girod, “The efficiency of motion-compensating prediction for hybrid coding of video sequences,” *IEEE Journal on Selected Areas in Communications*, vol. 5, no. 7, pp. 1140–1154, Aug. 1987.

[25] G. Casella and R. L. Berger, *Statistical Inference*, 2nd ed. Duxbury Press, 2001.

[26] J. Wu, X. Liu, and K.-Y. Yoo, “A temporal error concealment method for h.264/avc using motion vector recovery,” *IEEE Transactions on Consumer Electronics*, vol. 54, no. 4, pp. 1880–1885, Nov. 2008.

[27] Y. Wang and Q.-F. Zhu, “Error control and concealment for video communication: a review,” in *Proceedings of IEEE*, vol. 86, no. 5, 1998, pp. 974–997.

[28] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley-Interscience, 1991.

[29] T. Stockhammer, D. Kontopodis, and T. Wiegand, “Rate-distortion optimization for JVT/H. 26L video coding in packet loss environment,” in *Proc. International Packet Video Workshop*. Citeseer, 2002.

[30] G. Bjontegaard, “Calculation of average PSNR differences between RD-curves, 13th VCEG-M33 Meeting,” *Austin, USA*, 2001.

[31] JCT-VC, *Test Model under Consideration (TMuC)*, Dresden, Apr. 2010, JCTVC-A205, 1st JCT-VC Meeting.

[32] A. Leontaris and P. Cosman, “Video compression for lossy packet networks with mode switching and a dual-frame buffer,” *IEEE Transactions on Image Processing*, vol. 13, no. 7, pp. 885–897, 2004.

[33] “H.264/AVC reference software JM16.0,” Jul. 2009. [Online]. Available: http://iphome.hhi.de/suehring/tml/download

[34] L. Kondi, F. Ishtiaq, and A. Katsaggelos, “Joint source-channel coding for motion-compensated DCT-based SNR scalable video,” *IEEE Transactions on Image Processing*, vol. 11, no. 9, pp. 1043–1052, 2002.

[35] J. Ribas-Corbera and S. Lei, “Rate control in DCT video coding for low-delay communications,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 1, pp. 172–185, 1999.

[36] T. Chiang and Y. Zhang, “A new rate control scheme using quadratic rate distortion model,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 1, pp. 246–250, 1997.

[37] S. Ma, W. Gao, and Y. Lu, “Rate-distortion analysis for H. 264/AVC video coding and its application to rate control,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 12, p. 1533, 2005.

[38] X. Li, N. Oertel, A. Hutter, and A. Kaup, “Laplace distribution based Lagrangian rate distortion optimization for hybrid video coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 2, pp. 193–205, 2009.

[39] Z. He and S. Mitra, “Optimum bit allocation and accurate rate control for video coding via-domain source modeling,” *IEEE transactions on Circuits and Systems for Video Technology*, vol. 12, no. 10, 2002.

[40] H. Lee, T. Chiang, and Y. Zhang, “Scalable rate control for MPEG-4 video,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 6, pp. 878–894, 2000.

[41] K. Yang, A. Jacquin, and N. Jayant, “A normalized rate-distortion model for H. 263-compatible codecs and its application to quantizer selection,” in *Proceedings of IEEE ICIP*, vol. 2, 1997, pp. 41–44.

[42] H. Hang and J. Chen, “Source model for transform video coder and its application. I. Fundamental theory,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 2, pp. 287–298, 1997.

[43] F. Moscheni, F. Dufaux, and H. Nicolas, “Entropy criterion for optimal bit allocation between motion and prediction error information,” in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, vol. 2094, 1993, pp. 235–242.

[44] A. Ortega and K. Ramchandran, “Rate-distortion methods for image and video compression,” *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 23–50, 1998.

[45] T. Berger, *Rate distortion theory: A mathematical basis for data compression*. Prentice-Hall, Englewood Cliffs, NJ, 1971.

[46] J. Proakis and M. Salehi, *Digital communications*, 4th ed. McGraw-hill New York, Aug 2000.

[47] I. QUANTIZERS and R. ROBERTS, “Analog Source Digitization: A Comparison of Theory and Practice,” *IEEE Transactions on Information Theory*, 1967.

[48] N. Farvardin and J. Modestino, “Optimum quantizer performance for a class of non-Gaussian memoryless sources,” *IEEE Transactions on Information Theory*, vol. 30, no. 3, p. 485, 1984.

[49] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*. Prentice-Hall, 1983.

[50] Y. Shoham and A. Gersho, “Efficient bit allocation for an arbitrary set of quantizers.” *IEEE Trans. Acoust. Speech Signal Process.*, vol. 36, no. 9, pp. 1445–1453, 1988.

[51] H. Everett III, “Generalized Lagrange multiplier method for solving problems of optimum allocation of resources,” *Operations Research*, vol. 11, no. 3, pp. 399–417, 1963.

[52] J. G. Proakis, *Digital Communications*, 4th ed. McGraw-Hill, 2001.

[53] E. Malkamaki and H. Leib, “Evaluating the performance of convolutional codes over block fading channels,” *IEEE Transactions on Information Theory*, vol. 45, no. 5, p. 1643, 1999.

[54] S. Kallel and C. Leung, “Efficient ARQ schemes with multiple copy decoding,” *IEEE Transactions on Communications*, vol. 40, no. 3, pp. 642–650, 1992.

[55] K. Ramchandran and M. Vetterli, “Best wavelet packet bases in a rate-distortion sense,” *IEEE Transactions on Image Processing*, vol. 2, no. 2, pp. 160–175, 1993.

[56] G. Sullivan and T. Wiegand, “Rate-distortion optimization for video compression,” *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 74–90, 1998.

[57] Z. He, Y. Kim, and S. Mitra, “Object-level bit allocation and scalable rate control for MPEG-4 video coding,” in *MPEG-4. 2001 Proceedings of Workshop and Exhibition on*. IEEE, 2002, pp. 63–66.

[58] X. Li, N. Oertel, A. Hutter, and A. Kaup, “Rate-distortion optimized frame level rate control for H.264/AVC,” in *European Signal Process. Conf. (EUSIPCO)*, Lausanne, Switzerland, 2008.

[59] G. Begin, D. Haccoun, and C. Paquin, “Further results on high-rate punctured convolutional codes for Viterbi and sequential decoding,” *IEEE Transactions on Communications*, vol. 38, no. 11, pp. 1922–1928, 1990.

[60] S. Ma, Z. Li, and F. Wu, “Proposed draft of adaptive rate control,” in *Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, Doc. JVT-H017r3, 8th Meeting, Geneva*, 2003, pp. 20–26.

[61] Z. Li, W. Gao, F. Pan, S. Ma, K. Lim, G. Feng, X. Lin, S. Rahardja, H. Lu, and Y. Lu, “Adaptive rate control for H.264,” *Journal of Visual Communication and Image Representation*, vol. 17, no. 2, pp. 376–406, 2006.

[62] Y. Kim, Z. He, and S. Mitra, “A novel linear source model and a unified rate control algorithm for H. 263/MPEG-2/MPEG-4,” in *Proceedings of ICASSP*, vol. 3. IEEE, 2001, pp. 1777–1780.

[63] Z. He and S. Mitra, “A linear source model and a unified rate control algorithm for DCT video coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 11, pp. 970–982, 2002.

[64] M. Simon and D. Divsalar, “Some new twists to problems involving the Gaussian probability integral,” *IEEE Transactions on Communications*, vol. 46, no. 2, 1998.

BIOGRAPHICAL SKETCH

Zhifeng Chen received the B.E. degree from the East China University of Science and Technology, Shanghai, China, in 2001, and the M.S. and Ph.D. degrees from the University of Florida, Gainesville, Florida, in 2008 and 2010 respectively. From 2002 to 2003, he was an engineer in EPSON (China), and from 2003 to 2006, he was a senior engineer in Philips (China), both working in mobile phone system solution design. From May 2009 to Aug 2009, he was an intern in Dolby, Burbank, CA, where he had worked in error resilient rate distortion optimization. He joined the strategic engineering department at Interdigital, King of Prussia, PA, in 2010, where he is currently a staff engineer working in the video coding research. His research interests include low-complexity video and image compression, perceptual video coding, Error-resilient video coding, rate-distortion optimization, rate control, cross-layer design, information theory, statistics, signal processing. He is the author of several journal and conference papers and has been awarded four patents.