

การเข้ารับสวัสดิต์สน์ที่ทนทานต่อความผิดพลาดแบบต้นทาง-ช่องสัญญาณ  
ร่วมกันสำหรับการส่งวิดีโอไร้สาย

นายรานคลี โดมิงโก คาโจต

ศูนย์วิทยุทรัพยากร

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2552

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

JOINT SOURCE CHANNEL ERROR-RESILIENT VIDEO CODING FOR  
WIRELESS VIDEO TRANSMISSION



Mr. Rhandley D. Cajote

A Thesis Submitted in Partial Fulfillment of the Requirements  
for the Degree of Doctor of Philosophy Program in Electrical Engineering

Department of Electrical Engineering

Faculty of Engineering

Chulalongkorn University

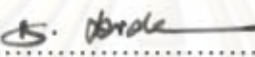
Academic Year 2009

Copyright of Chulalongkorn University

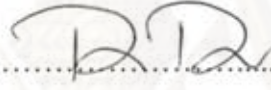
Thesis Title                   JOINT SOURCE CHANNEL ERROR-RESILIENT  
VIDEO CODING FOR WIRELESS VIDEO  
TRANSMISSION  
By                                 Mr Rhandley D. Cajote  
Field of Study                 Electrical Engineering  
Thesis Advisor               Assistant Professor Supavadee Aramvith, Ph.D.

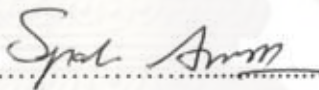
---

Accepted by the Faculty of Engineering, Chulalongkorn University in  
Partial Fulfillment of the Requirements for the Doctoral Degree


..... Dean of the Faculty of Engineering  
(Associate Professor Boonsom Lerdhirunwong, Dr.Ing)


THESIS COMMITTEE

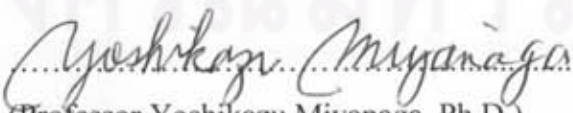
.....Chairman  
(Professor Prasit Prapingmongkolkarn, Ph.D.)

.....Thesis Advisor  
(Assistant Professor Supavadee Aramvith, Ph.D.)

.....Examiner  
(Associate Professor Watit Benjapolakul, Ph.D.)

.....Examiner  
(Assistant Professor Suree Pumrin, Ph.D.)

.....External Examiner  
(Assistant Professor Wuttiping Kumwilaisak, Ph.D.)

.....External Examiner  
(Professor Yoshikazu Miyanaga, Ph.D.)

Rhandley D. Cajote: การเข้ารหัสวิดีโอที่ทนทานต่อความผิดพลาดแบบต้นทาง-ช่องสัญญาณร่วมกันสำหรับการส่งวิดีโอไร้สาย (JOINT SOURCE CHANNEL ERROR-RESILIENT VIDEO CODING FOR WIRELESS VIDEO TRANSMISSION) อ. ที่ปริกษาวิทยานิพนธ์หลัก: ศศ.คร. สุภาวดี อร่วมวิทย์, 101หน้า.

ความสำเร็จของการส่งวิดีโอผ่านช่องสัญญาณแบบไร้สายเป็นปัญหาเชิงเทคนิคที่ทำห้ายมากที่สุดในการสื่อสารแบบสื่อประสม ข้อผิดพลาดที่เกิดขึ้นในการบีบอัดข้อมูลวิดีโอและธรรมชาติของช่องสัญญาณไร้สายที่แปรเปลี่ยนตามเวลาและเฟดดิ้งรวมถึงข้อจำกัดทางโครงสร้างและงานประยุกต์ที่ใช้ ส่งผลให้มีประเด็นที่ต้องการเทคนิคใหม่มาใช้ในการแก้ปัญหา

เพื่อปรับปรุงให้สัญญาณวิดีโอที่ส่งผ่านช่องสัญญาณไร้สายมีความทนทานต่อความผิดพลาด และประสิทธิภาพสูงขึ้น งานวิทยานิพนธ์นี้ได้นำเสนอการพัฒนาเทคนิคในการใช้คุณลักษณะความทนทานต่อความผิดพลาดของ H.264 และวิธีการปรับปรุงการควบคุมอัตราระดับเฟรม ในลำดับแรกได้นำเสนอเทคนิคใหม่ในการใช้การจัดเรียงมาโครบล็อกแบบยืดหยุ่นได้ (FMO) แบบซัดแจ้งอย่างมีประสิทธิภาพมากขึ้นในการส่งวิดีโอไร้สาย โดยรวมตัวชี้วัดความสำคัญเชิงพื้นที่และเชิงเวลาของมาโครบล็อกในการสร้างแผนที่กลุ่มสไลซ์สำหรับวิดีโอ H.264 ในลำดับที่สองได้พัฒนาระเบียบวิธีการจัดเรียงลำดับที่แบ่งประเภทของมาโครบล็อกเป็นหลายกลุ่มสไลซ์สำหรับการจัดเรียงมาโครบล็อกแบบยืดหยุ่นได้ใน H.264 เพื่อปรับปรุงการใช้งานของการจัดเรียงมาโครบล็อกแบบยืดหยุ่นได้แบบซัดแจ้ง ในลำดับที่สามได้เสนอการปรับปรุงการควบคุมอัตราระดับเฟรมของ H.264 เมื่อมีการใช้การจัดเรียงมาโครบล็อกแบบยืดหยุ่นได้ในการส่งวิดีโอ เพื่อการจัดสรรบิตและประสิทธิภาพในการเข้ารหัสที่ดีขึ้น ซึ่งประกอบด้วยการนำเสนอแบบจำลองบิตส่วนหัวใหม่ การวัดความซับซ้อนของเฟรมที่ดีขึ้น การจัดสรรบิต และการปรับพารามิเตอร์การควอนไทซ์ที่ดีขึ้น ในท้ายที่สุดได้นำเสนอกรอบของการเลือกการจัดเรียงมาโครบล็อกแบบยืดหยุ่นได้จากข้อมูลป้อนกลับ

โดยสรุปในวิทยานิพนธ์นี้ได้นำเสนอกรอบการทำงานของการจัดเรียงมาโครบล็อกแบบยืดหยุ่นได้แบบซัดแจ้ง การควบคุมอัตรา และข้อมูลป้อนกลับในการส่งวิดีโอไร้สาย ผลการทดลองแสดงให้เห็นว่าเทคนิคที่นำเสนอมีประสิทธิภาพในการปรับปรุงคุณภาพวิดีโอให้ดีขึ้น โดยเฉพาะลำดับภาพวิดีโอที่มีความซับซ้อนและที่อัตราบิตต่ำ

ภาควิชา.....วิศวกรรมไฟฟ้า.....ลายมือชื่อผู้สมัคร.....*Randley*  
สาขาวิชา.....วิศวกรรมไฟฟ้า.....ลายมือชื่ออาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก.....*M.L.*  
ปีการศึกษา.....2552.....



## 4971873121: MAJOR ELECTRICAL ENGINEERING

KEYWORDS: JOINT SOURCE-CHANNEL / VIDEO CODING / WIRELESS / ERROR-RESILIENT/

RHANDLEY D. CAJOTE: JOINT SOURCE CHANNEL ERROR-RESILIENT VIDEO CODING FOR WIRELESS TRANSMISSION. THESIS ADVISOR: ASST. PROF. SUPAVADEE ARAMVITH, Ph.D., 101 pp.

Successful video communication over wireless channels is one of the most technically challenging problems in multimedia communications. The inherent vulnerability of compressed video to transmission errors and the time varying and fading environments of the wireless channel combined with network and application constraints presents some major technological issues that needs to be addressed.

To improve the error resiliency and efficiency of transmitting video over wireless channels. Several techniques are developed that contribute to better utilize the error resilient features available in H.264 and propose enhancements to the frame layer rate control. Firstly, new techniques on how to use explicit Flexible Macroblock Ordering (FMO) more effectively in wireless video transmission by using combined spatial and temporal indicators of macroblock (MB) importance to generate slice group maps for H.264. Secondly, an improved sorting algorithm that classifies macroblocks into different slice groups for FMO in H.264 that further improves the use of explicit FMO. Thirdly, enhancements to the H.264 frame layer rate control is proposed, that takes into consideration the effects of using FMO for video transmission, to better manage bit allocation and improve coding efficiency. We propose a new header bits model, an enhanced frame complexity measure, a bit allocation and a quantization parameter (QP) adjustment scheme. Lastly, a framework for adaptive FMO selection scheme based on feedback information is presented.

In summary, in this dissertation we propose the framework of using explicit FMO, rate control and feedback information for wireless video transmission. Experimental results show that the proposed techniques are effective in improving the video quality especially for complex video sequences and at low-bit rates.

Department.....Electrical Engineering... Student's Signature.....*R. Cajote*.....  
 Field of study.....Electrical Engineering... Advisor's Signature.....*Supavadee Aramvith*.....  
 Academic year.....2009.....

## ACKNOWLEDGEMENTS

I would like to express my sincerest gratitude to my advisor Asst. Prof. Supavadee Aramvith for her tireless efforts, encouragement, support and guidance throughout the entire course of my study.

My thanks also goes to the members of the committee, Associate Professor Watit Benjapolakul, Ph.D., Assistant Professor Suree Pumrin, Ph.D. Assistant Professor Wuttiping Kumwilaisak, Ph.D, Professor Yoshikazu Miyanaga, Ph.D, and the Chairperson Professor .Prasit Prapingmongkolkarn, Ph.D, for their time, critical review and advice for this work.

Thanks to all members of the Video Processing Research Group for their assistance and companionship. They have been good friends in the lab: Jip, Ton, Bua, Fang, Simon, Seno, M and Nan.

Finally, I would like to express my heartfelt thanks to all my colleagues at the Institute of Electrical and Electronics Engineering at the University of the Philippines and also to the people of the Digital Signal Processing (DSP) Lab. Special mentions to Ma'am Gev, Doc Ramos, Doc Marciano, Franz, Tess, Tata, Mimi and Veron for all their support.

I would like to thank also my parents, everyone in my family, my cousins in Cebu, Cavite and Laguna for their understanding and moral encouragements.

Special thanks also to the people at ISE and AUN/SEED-Net: Khun Kanasom, Khun Tudtoo, Khun Vantanee and Khun Kalayaporn for their support and advice.

This research has been supported in part by the Collaborative Research Project entitled Wireless Video Transmission, JICA Project for AUN/SEEDNet, Japan.

# CONTENT

	<b>Page</b>
THAI ABSTRACT .....	iv
ENGLISH ABSTRACT.....	v
ACKNOWLEDGEMENTS .....	vi
LIST OF TABLES .....	x
LIST OF FIGURES .....	xi
<b>CHAPTER I INTRODUCTION</b>	
1.1 Motivation.....	1
<b>CHAPTER II BACKGROUND</b>	
2.1 Video Compression Concepts.....	5
2.2 The H.264/AVC video coding standard.....	7
2.3 Slice Structured Coding and Flexible Macroblock Ordering in H.264/AVC .....	10
2.4 The Importance of Rate Control .....	15
2.5 Frame Layer Rate Control in H.264/AVC.....	18
2.6 Wireless Channel Characteristics .....	20
2.6.1 Time Delay Spread and Coherence Bandwidth.....	20
2.6.2 Fading, Coherence Time and Doppler Spread.....	22
2.6.3 Wireless Channel Simulator .....	23
2.6.4 Wireless Channel Prediction and Markov Modeling.....	27
<b>CHAPTER III ERROR RESILIENT VIDEO CODING IN H.264 FOR WIRELESS VIDEO TRANSMISSION</b>	
3.1 Spatial-Temporal Indicator for Explicit FMO (STI-FMO) in H.264/AVC .....	29
3.1.1 Effect of errors on FMO slice groups.....	30
3.1.2 Limitations of Bit-Count and Distortion Measure for	

Explicit FMO .....	30
3.1.3 Combining Spatial and Temporal Indicators.....	32
3.1.3.1. Using the Initial Slice map.....	33
3.1.3.2. Loss probability of a macroblock.....	33
3.1.3.3. Distortion-from-propagation measure.....	33
3.1.3.4. Experimental Set-up.....	34
3.1.3.5. Results and Analysis .....	35
3.1.3.6. Summary .....	37
3.2 Improved Sorting Algorithm for Explicit FMO in H.264/AVC.....	37
3.2.1 Independence of Slice groups in Slice Structured Coding of H.264/AVC.....	37
3.2.2 Divide and Conquer Approach to Classification Process.....	38
3.2.3 Improved Sorting Algorithm for FMO.....	40
3.2.4 Results and Analysis.....	41
3.2.5 Summary.....	44
3.3 Improved FMO-based Frame Layer Rate Control for H.264/AVC.....	45
3.3.1 Proposed Header Bits Model.....	45
3.3.2 Proposed Frame Complexity Measure .....	48
3.3.3 Proposed Frame Layer Rate Control Enhancement .....	49
3.3.4 Computation of the Frame Layer Target Bits.....	50
3.3.5 Using the Proposed Header Bits Model .....	51
3.3.6 Using Frame Complexity and QP Adjustment.....	51
3.3.6.1. Negative Target Bit.....	52
3.3.6.2. Positive Target Bit.....	52
3.3.6.3. Lower Bound on Texture Bit .....	53
3.3.7 Frame Skipping.....	53
3.3.8 Results and Discussion .....	53
3.3.9 Trade-off of the Proposed Rate Control Enhancements.....	59
3.3.10 Summary.....	61
3.4 Feedback based FMO selection strategy.....	64



3.4.1 Simulated Video Transmission System.....	65
3.4.2 Proposed FMO-selection strategy .....	65
3.4.3 Experimental Results.....	66
3.4.4 Summary.....	69
CHAPTER IV CONCLUSIONS AND FUTURE WORKS	
4.1 Summary and Conclusion.....	70
4.2 Future Works .....	72
REFERENCES .....	73
APPENDICES .....	76
VITAE.....	101



ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย

## LIST OF TABLES

<b>Table 3.1</b> Comparison of Number of Undecodable Macroblocks .....	35
<b>Table 3.2</b> Comparison of Average PSNR (dB) .....	36
<b>Table 3.3.</b> Summary statistics of bit-count and $D_{CE}$ using simple sorting and the proposed modified sorting algorithm .....	42
<b>Table 3.4</b> Summary of MB loss count for bit count and Average PSNR for Distortion .....	43
<b>Table 3.5</b> Number of non-zero motion vector difference at different QP and different number of FMO slice groups for carphone sequence .....	46
<b>Table 3.6</b> Comparison of $R^2$ values between the model in (Kwon, 2007) and .....	47
<b>Table 3.7.</b> Comparison of PSNR and PSNR standard deviations averaged over several bit rates and different numbers of FMO slice groups.....	54
<b>Table 3.8</b> Comparison of PSNR and PSNR standard deviation averaged over different number of FMO slice groups at 20 kbps.....	57
<b>Table 3.9</b> Comparison of PSNR and PSNR standard deviation averaged over different number of FMO slice groups at 32 kbps.....	58
<b>Table 3.10</b> PSNR for foreman sequence at different bit rates and different number of FMO slice groups.....	58
<b>Table 3.11</b> Average PSNR (dB) at 20 Kbps.....	66

## LIST OF FIGURES

<b>Figure 2.1</b> Block Diagram of a hybrid video encoder.....	6
<b>Figure 2.2</b> Effect of bit errors on compressed video stream (a) without slices headers (b) with slice headers. ....	11
<b>Figure 2.3</b> Different predefined FMO map types in H.264/AVC.....	12
<b>Figure 2.4</b> Example of explicit FMO slice group with eight slice groups.....	12
<b>Figure 2.5</b> The motion vector for macroblock X is predicted from neighboring macroblocks A, B and C.....	14
<b>Figure 2.6</b> Illustration of a Rate Control for H.264/AVC Video Encoder.....	15
<b>Figure 2.7</b> Rate and Distortion relationship of hybrid video encoders.....	17
<b>Figure 2.8</b> Maximum time delay spread of a typical indoor wireless channel.....	21
<b>Figure 2.9</b> Schematic of the wireless channel simulator.....	23
<b>Figure 2.10</b> Effect of inter-path delay and SNR on BER.....	24
<b>Figure 2.11</b> Effect of antenna diversity on BER.....	25
<b>Figure 2.12</b> Average duration of fade (ADF) and Level crossing rate (LCR).....	25
<b>Figure 2.13</b> Effect of Doppler frequency $f_D$ on ADF.....	26
<b>Figure 2.14</b> Effect of Doppler frequency $f_D$ on the frequency of error burst.....	26
<b>Figure 2.15</b> Two-state Markov model of wireless channel.....	27
<b>Figure 3.16</b> Distortion measure, 6 <sup>th</sup> frame of Akiyo sequence.....	31
<b>Figure 3.17</b> Macroblock-to-slice group assignment, 6 <sup>th</sup> frame of Akiyo.....	32
<b>Figure 3.18</b> Comparison of PSNR using carphone sequence under slow fading.....	36
<b>Figure 3.19</b> Comparison of PSNR using carphone sequence under fast fading.....	36
<b>Figure 3.20</b> From the 3 <sup>rd</sup> frame of carphone sequence. (a) The MBA map using bit- count and the improved sorting algorithm, (b) the MBA map using bit-count and interleave sorting algorithm.....	41
<b>Figure 3.21</b> (a)-(b) Carphone sequence encoded at QP=32 and Rate=48 kbps, no FMO and FMO with eight slice groups.....	49
<b>Figure 3.22</b> Comparison of PSNR at 32kbps using FMO with eight slice groups for Carphone and Foreman sequence.....	56
<b>Figure 3.23</b> Comparison of R-D curve with JVT reference rate control for Carphone .....	62
<b>Figure 3.24</b> Comparison of R-D curve with JVT reference rate control for Foreman	63
<b>Figure 3.25</b> Plots of Error ratio, packet errors and FMO mode selection for carphone sequence at 20 kbps. ....	67
<b>Figure 3.26</b> Comparison of PSNR of Caprhone sequence at 20 kbps using different number of FMO slice groups and the adaptive selection method.....	67

**Figure 3.27** Plots of Error ratio, packet errors and FMO mode selection for carphone sequence at 32 kbps. ....68

**Figure 3.28** Comparison of PSNR of Carphone sequence at 32 kbps using different number of FMO slice groups and the adaptive selection method.....69



ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย



# CHAPTER I

## INTRODUCTION

### 1.1 Motivation

In today's multimedia communication systems, the transmission of video over mobile radio channels is the most demanding in terms of bit rate and presents many technically challenging issues that have not been completely addressed. Robustness of video transmission systems through error prone environments is the one of the most difficult things to achieve in video communications technology. Even with state-of-the-art compression technique, compressed video signals still require a significant amount of transmission bandwidth. Video coding at "low-bit rate" involves video compression at bit rates ranging from a few tens to a few hundreds of kilobits per second and is inherently a variable-rate process. Although, transmission of low resolution and limited motion video sequences typical in video phone applications are now made possible due to the developments in higher bandwidth networks such as UMTS (Universal Mobile Telecommunication Systems). The transmission of high quality video signals is still beyond the practical capacities of existing wireless communication networks. The demand for higher bandwidths in the wireless communications segment have spin-off new technologies to deliver very high speed data services for fixed and mobile locations such as WiMAX (Worldwide Interoperability for Microwave Access) and LTE (Long Term Evolution). But because of the hostile propagating environments inherent in wireless communication systems, characterized by high error rates caused by the combined effects of attenuation and multi-path fading, coupled with the requirements of user mobility and multiple access will likely reduce the throughput of any wireless communication system. In general, the available bandwidths over radio links will always be limited and low bit rates video transmissions are likely to be typical, thus achieving very high compression efficiency is necessary for successful video transmission in a wireless mobile environment.

In order to achieve very high compression efficiencies, modern video compression technologies exploit both temporal and spatial redundancies in the video

signal. The previously encoded and reconstructed video frames are used to predict the next frames. This inter frame coding mechanism is combined with very efficient entropy coding techniques to generate highly compressed video streams. Efficient encoding of video signals renders the compressed bit stream highly prone to transmission errors. Loss of information on one frame has the potential to degrade the video quality of succeeding frames due to the temporal and spatial propagation of errors in the video sequence. This makes the compressed video signals extremely vulnerable to transmission errors under fading wireless channels. Although traditional channel coding schemes can detect and correct some errors, this often leads to reduced channel throughput because of the added redundant information.

In any video transmission system there is an inherent trade-off in coding efficiency and error-resiliency. Because of the hostile propagation environments inherent in mobile radio link channels, video compression algorithms must not only be very efficient but error-resilient as well. Aside from the limited available bit rate, the most difficult issue in wireless video transmission is the error-prone characteristic of wireless channels. Transmission errors on mobile wireless radio channels range from random bit errors to frequent burst errors and high bit error rates occur during fading periods. Even with the use of common transmission error protection schemes such as FEC (Forward Error Correction) and ARQ (Automatic Repeat reQuest), transmission errors cannot be avoided. The design of wireless video transmission systems always involves a trade-off between channel coding redundancy to protect the bit stream and source coding redundancy to provide greater error resiliency to the video decoder (Girod, 2000).

Thus successful video communication in the presence of errors requires careful design strategies at the encoder, decoder and other system layers. However, most error control schemes developed so far are pragmatic engineering solution to a problem at hand and are not based on a generalized framework. The trade-off in designing the overall transmission system chains are not yet fully understood and still needs further study. Ultimately, the end goal is to achieve a more generalized theoretical framework for joint optimization of source coding parameters, channel coding, transmission protocols, rate and complexity constraints that makes most efficient use of the limited wireless network resources. In the meantime, we are content to be able to achieve more modest goals. This research contributes to the development of an error-resilient video transmission system by making more effective

use of the tools available at the encoder and considering channel information to make more intelligent encoder decisions.

The need for error-resilient tools at the encoder and coordinated error control strategies are some of the motivations in the development of the latest video coding standard. To date, the H.264/AVC is the best candidate codec for video communications because of its high coding efficiency and error robustness features. The H.264/AVC also known as MPEG-4 Part 10 is the latest international video coding standard developed by the Joint Video Team (JVT). The novel features of the H.264/AVC have significantly improved the coding efficiency and provide support for enhanced error resiliency compared to previous coding standards. Error resilient tools such as Flexible Macroblock Ordering (FMO), Data partitioning, Arbitrary Slice Ordering (ASO) and redundant slice are just some of the enhancements in the new standard. In this work we focus on the use of FMO for wireless video transmission. Using FMO allows partitioning a frame into several slices, and each slice into several slice groups. The standard allows full flexibility in assigning macroblocks to slice groups at the expense of reduced coding efficiency because of the additional overhead bits incurred for the slice group header bits, signaling and macroblock-to-slice group map information.

In this research we develop methods on how best to use FMO for wireless video transmission. One approach to effectively use FMO is to classify important macroblocks in the video sequence and assign them to slice groups in such a way that loss of a particular slice during transmission will not be too detrimental to the quality of the decoded video. Based on temporal and spatial characteristics of the video sequence, we develop a suitable mapping of macroblocks to slice groups that make the video sequence more error resilient to transmission errors in both slow and fast fading wireless channel conditions compared with the predefined FMO mappings in H.264/AVC. In line with this, we also develop an improved sorting algorithm for classifying macroblocks into slice groups. The improved sorting algorithm takes into consideration the macroblock importance parameter that is used, either spatial or temporal information, then classifies the macroblocks such that the statistical properties of the macroblock parameter is more or less equal across all slices.

Using FMO as an error-resilient tool also serves as a macroblock level interleaving tool that has the ability to stop the spatial propagation of error within the frame. FMO also helps the decoder make use of the available macroblock spatial



information to design better error concealment algorithms. However, overhead incurred by using FMO becomes prohibitive at low bit rate, and judicious use of FMO is required to meet the target rate and still provide sufficient error resiliency. To improve the coding efficiency when FMO is enabled at the encoder we propose FMO-based enhancements to the frame layer rate control of the H.264/AVC. By using a new header bits model, an improved frame complexity measure, target bit allocation and a quantization parameter (QP) adjustment scheme that takes into consideration the buffer fullness, available bandwidth and number of slice groups a more efficient frame layer bit allocation scheme is achieved compared to the existing adaptive rate control in H.264/AVC.

The problem of successful video communication over wireless networks have been partially addressed with our proposed explicit FMO mapping strategy and improved sorting algorithm, combined with the proposed enhancements to the frame layer rate control, improvements in video quality and coding efficiency have been achieved. The burst error nature of the wireless channel however presents a difficult problem to address. Burst errors can affect consecutive packets that have the potential to damage several frames. At low bit rates the effects of burst errors is severe degradation in the decoded video quality. To address this problem we develop a feedback-based FMO mode selection scheme to efficiently select the frames that will be encoded with FMO. By making use of feedback information to predict the conditions of the channel a decision is made to enable FMO only to certain frames that are likely to suffer from burst errors. This scheme has been found to be effective in mitigating the effects of burst errors for low bit rate applications with certain wireless transmission constraints.

The organization of the dissertation is as follows: Chapter 2 we discuss some basic video compression concepts, some background about the H.264/AVC video codec, some error resilient tools in H.264/AVC and the issues about rate control. In Chapter 3 we discuss the proposed techniques to make more effective use of explicit FMO in wireless channel environments, our proposed rate control enhancements and FMO selection scheme for low bandwidth systems under slow fading environments. In Chapter 4 we make some conclusions and recommendations for future works.



## CHAPTER II

### BACKGROUND

In this chapter we give an overview of some video compression concepts and the latest H.264/AVC video coding standard. A discussion on the trade-offs of using FMO as an error-resilient tool is presented and also the importance of rate control in any video coding system. The characteristic of the mobile wireless channel is presented to provide some basic understanding of the nature of errors in fast and slow fading channel conditions. Finally, some discussion about wireless channel simulation and modeling.

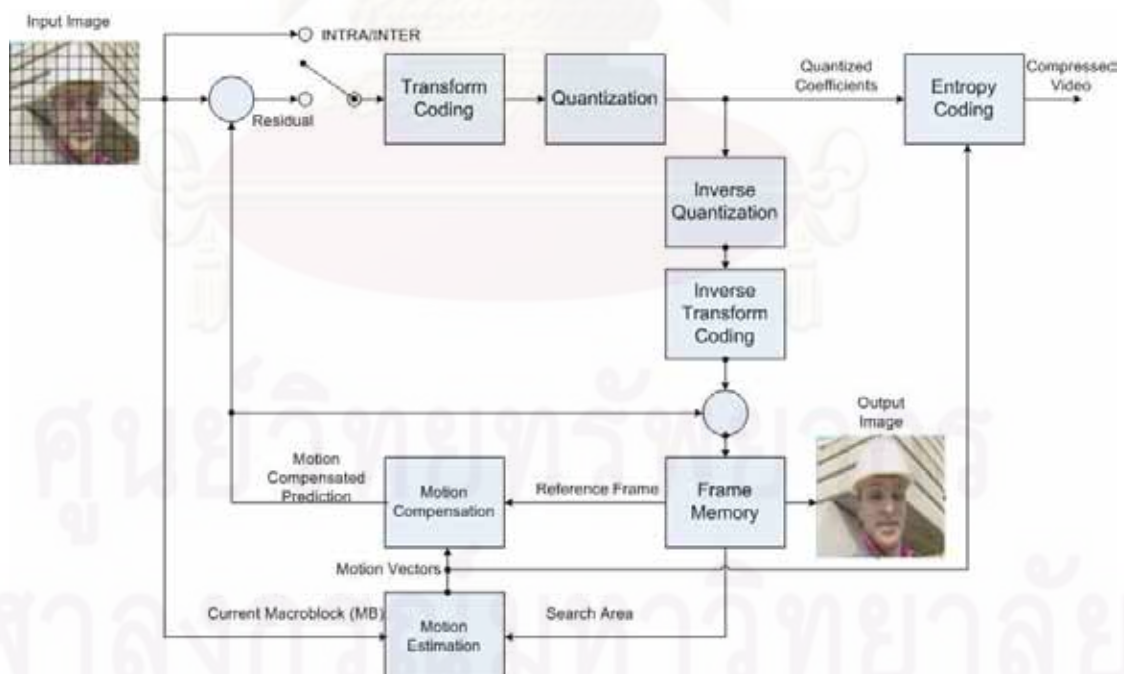
#### 2.1 Video Compression Concepts

One way to achieve video compression is to code each image frame using standard image coding scheme such as JPEG. The most common JPEG compression scheme is to segment the image frame into smaller units, usually into 8x8 blocks of samples, these are also called macroblocks. Each macroblock is transformed using Discrete Cosine Transform (DCT) to reduce spatial redundancy. The transform coefficients are then quantized; this quantization process results in some loss of spatial details depending on the preferred quality of the decoded picture. The amount of spatial detail that is retained in the decoded picture depends on a quantization parameter (QP) used in the quantization process. The QP also determines the desired level of compression, using higher QP values will result in less compression but more spatial detail and lower QP values will result in greater compression but degrades the picture quality. Afterwards, the quantized coefficients are entropy coded to reduce statistical redundancy. This form of coding is called INTRA-coding, since the picture is coded without referring to other pictures in the video sequence.

Improved coding performance can be achieved by exploiting the large amount of temporal redundancy in the video content. Much of the depicted scene is essentially repeated in succeeding pictures with small significant change. By coding only the changes in the scene as compared to coding each picture separately the video content can be represented more efficient. The difference or change in the scene is called *residual* information, and the technique of coding only the *residual* information is called INTER-coding. The ability to use temporal redundancy to improve coding

efficiency is what fundamentally distinguishes video compression from still-image compression.

Further improvement in coding efficiency can be achieved by exploiting the temporal dependence in the video signal. This technique is called *motion compensated prediction* (MCP). The motivation in using MCP is that most changes in the video content are typically due to the motion of objects in the depicted scene relative to the image plane. A small amount of motion can result in a large difference in values of the samples in the picture. By predicting an area of the current picture from a region in the previous picture that is displaced by a few samples in spatial location can reduce the amount of residual information. The use of spatial displacement motion vectors to form a prediction is called *motion compensation* (MC). And the encoder's search for the best *motion vector* (MV) is called *motion estimation* (ME). The coding of the resulting residual information is called MCP residual coding. The use of two redundancy reduction techniques using prediction and transformation leads to the basic design of the hybrid video encoder. The technologies applied to the development and design of efficient hybrid video encoders, as shown in Figure 2.1, form the basic building blocks of most video coding standards.



**Figure 2.1** Block Diagram of a hybrid video encoder

The operations of the hybrid video encoder are described as follows. When the

encoder operates in the INTRA mode, each macroblock in the image is transformed (DCT), quantized and entropy coded. The usual entropy coding scheme uses Variable Length Coding (VLC), but more recent standards use Arithmetic coding. At the decoder, the inverse operations are performed, Variable Length Decoding (VLD), followed by inverse quantization and inverse transform (IDCT).

But when the encoder operates in INTER mode, each macroblock is compared with the previously reconstructed frame stored in the frame memory and motion estimation is performed to find the best motion vector that will be used for motion compensation. The output of motion compensation block is a motion compensated prediction of each macroblock. The difference between the motion compensated prediction and the original macroblock is computed to generate the prediction error, or *residual* information. The residual information is encoded, quantized, and entropy coded together with the motion vectors as side information. At the decoder the motion vectors are used in motion estimation to recover the motion compensated prediction information and the residual information is added to reconstruct the original image.

The subsequent improvements in motion compensated prediction are the main reasons for the improved coding efficiency in modern video coding standards from generation to generation. More sophisticated motion compensated prediction techniques also requires increased computational complexity. The improvements in motion compensated prediction that eventually found their way into the latest video coding standard, H.264/AVC, are fractional-sample-accurate motion compensation, motion vectors over picture boundaries. There are also bi-predictive, variable block size, multi-picture, multi-hypothesis and weighted motion compensated prediction.

## **2.2 The H.264/AVC video coding standard**

The H.264/AVC or MPEG-4 Part 10 was recently developed by the JVT (Joint Video Team) consisting of experts from VCEG (Video Communication Experts Group) and MPEG (Moving Picture Experts Group) to improve the video compression of the previous standards. The H.264/AVC is conceptually divided into two layers, the Video Coding Layer (VCL) and the Network abstraction layer (NAL). The VCL is responsible for the efficient representation of the source signal achieved through significant improvements in the encoding process. The design of the NAL allows improved network adaptation that allows seamless and easy integration of the

coded video signals into all current and possible future protocol and multiplex architecture. Thus H.264/AVC can support various applications like video broadcasting, video conferencing, video streaming over fixed and wireless networks, and over different transport protocols.

Similar to previous standards the H.264/AVC shares the same basic building blocks of a hybrid video encoder as discussed in the previous section. Some of the main features of H.264/AVC that highlights the features of the video coding layer are given here. A more detail discussion of the source coding tools in H.264/AVC can be found in (Weigand, 2003). It should be noted that there is no single component of the H.264/AVC that can be attributed to the dramatic increase in coding performance. Rather it is due to the combination of small performance improvements in each of the functional components of the system.

The improvements in the prediction method are as follows:

- Adaptive block size for motion compensation: the standard supports more flexibility in the selection of motion compensation block sizes and shapes, with a minimum luminance motion compensation block size as small as 4x4.
- Quarter-sample-accurate motion compensation: The H.264/AVC allows quarter-sample motion vector accuracy, and also further reduces the complexity of the interpolation process compared to previous standards.
- Multiple Reference Picture Motion Compensation: The H.264/AVC allows the encoder to select among a large number of pictures that have been decoded and stored in the decoder for motion compensation purposes.
- Directional spatial prediction for Intra coding: a new technique of extrapolating the edges of previously-decoded parts of the current picture is applied in regions of pictures that are Intra coded. This improves the quality of the prediction signal and also allows prediction from neighboring areas that were not coded using intra coding.
- In-loop deblocking filter: Block-based video coding produces artifacts known as blocking artifacts. The artifacts can originate from both the prediction and residual difference coding stages of the decoding process. Application of a deblocking filter is a well-known method of improving the quality of the decoded signal. The H.264/AVC deblocking filter is brought within the motion-compensated prediction loop, so that the improvement in quality can



be used in inter-picture prediction to improve the ability to predict other pictures as well.

Other improvements that contribute to improved coding efficiency are the following:

- Small block-size transforms: the H.264/AVC design is based on 4x4 transform, this allows the encoder to represent signals in a more locally-adaptive fashion.
- Short-word length transform: prior standards designed generally require that encoder and decoder use 32-bit processing, the H.264/AVC only requires 16-bit arithmetic.
- Exact-match inverse transform: In previous video coding standards, the transform used for representing the video was generally specified only within an error tolerance bound, due to the impracticality of obtaining an exact match to the ideal specified by the inverse transform. The H.264/AVC is the first standard to achieve exact equality of decoder video content from all decoders by using integer transforms.

The robustness to data errors and losses of the H.264/AVC is enabled by the following error resilient tools.

- Flexible slice structured coding: Slices provide spatially distinct resynchronization points within the video data for a single frame. Intra prediction and motion vector prediction are not allowed over slice boundaries. The slice sizes in H.264/AVC are highly flexible as compared to the rigid slice structure found in MPEG-2.
- Flexible Macroblock Ordering (FMO): A new ability to partition the picture into regions called slice groups and each slice becomes an independently-decodable subset of the slice group. When used effectively, in conjunction with an appropriate error concealment method, FMO can significantly enhance the robustness of data losses by managing the spatial relationship between regions that are coded in each slice.
- Arbitrary Slice Ordering (ASO): Since each slice of a coded picture can be decoded independent of the other slices of the picture, the H.264/AVC design

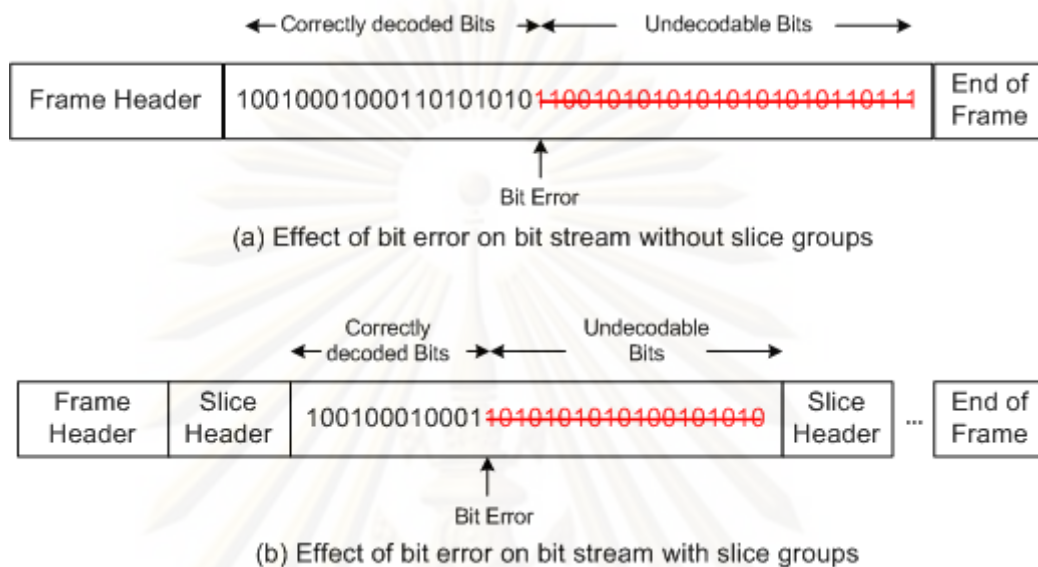
enables sending and receiving the slices of the picture in any order relative to each other.

- Redundant pictures: To enhance the robustness to data loss, the H.264/AVC design has a new ability to allow the encoder to send redundant representation of regions of pictures, enabling a coarse representation of regions of pictures for which the primary representation has been lost during data transmission
- Data partitioning: Since the coded information for representation of each region is more important than other information for purposes of representing the video content, H.264/AVC allows the syntax of each slice to be separated into up to three different partitions for transmission.
- SP/SI synchronization/switching pictures: A new feature in H.264/AVC consisting of picture types that allow exact synchronization of the decoding process of some decoders with an ongoing video stream produced by other decoders without penalizing all decoders with the loss of efficiency resulting from sending an Intra-coded picture. This can enable switching a decoder between representation of the video content that used different data rates, recovery from data losses or errors as well as enabling fast-forward and fast-reverse playback functionality.

### **2.3 Slice Structured Coding and Flexible Macroblock Ordering in H.264/AVC**

The basic motivation in developing FMO as an error resilient tool comes from the slice structured coding nature of H.264/AVC. The vulnerability of compressed video to transmission error comes from the use of variable length codes in entropy coding. The decoder relies on the length of previously decoded symbols to be able to decode the current symbol. If all the compressed data belonging to one frame is transmitted as a single data unit, then a single bit error in the bit stream can cause loss of synchronization in the decoder such that succeeding symbols are rendered undecodable until the next synchronization point. To make the compressed video frame more resilient to errors, the concept of *slices* is introduced. Slices provide spatially distinct resynchronization points within the frame. Slice header contains information about the macroblocks belonging to that slice and also contains syntactic and semantic resynchronization information. Thus dividing the compressed data within the frame into slices can stop the spatial propagation of error to within the slice, at the expense of overhead in slice headers, as illustrated in Figure 2.2. Also,

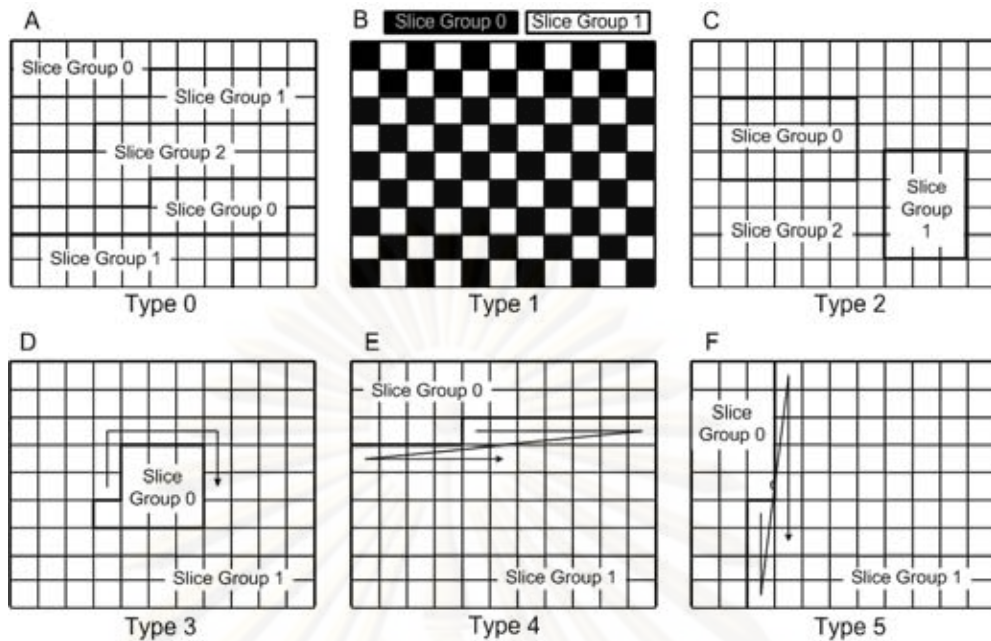
Intra prediction and inter-frame motion vector prediction is restricted to macroblocks belonging to the same slice, this makes each slice independently decodable from other slices. The concept of slices in H.264/AVC takes their most flexible and advanced form with the concept of flexible macroblock ordering.



**Figure 2.2** Effect of bit errors on compressed video stream (a) without slices headers (b) with slice headers.

Using FMO allows mapping of macroblocks to so-called *slice groups*. A slice group itself may contain several slices; hence a *slice group* can be thought of as an entity similar to a picture consisting of slices when FMO is not used. The mapping of the macroblocks to slice group map is specified using the Macroblock Allocation map or MBAMap defined in the *picture parameter set* (PPS). This allows flexibility in encoding and transmitting macroblocks other than the normal raster scan order. The H.264/AVC supports seven different FMO map types; six are predefined as shown in Figure 2.3. For a more detailed discussion of the predefined FMO map types, see (Lambert, 2006).





**Figure 2.3** Different predefined FMO map types in H.264/AVC

The seventh type of FMO map (Type 6), also called the *explicit* FMO map type, allows full flexibility in assigning macroblocks to slice groups as specified in the MBAMap through the PPS. An example of an explicit FMO slice group map is illustrated in Figure 2.4, extracted from the 8<sup>th</sup> frame of Akiyo sequence. The standard supports a maximum of up to eight slice groups. In this research focused mainly on the use of explicit FMO maps, by modifying the macroblock-to-slice group maps we can mitigate the effects of transmission errors when transmitting video over wireless channels. Nevertheless, because of the non-stationary nature of errors in the wireless channel and because of an almost infinite number of ways to assign macroblocks to slice groups, the issue of designing appropriate macroblock-to-slice group map for purposes of being more error resilient still remains an active research area.



**Figure 2.4** Example of explicit FMO slice group with eight slice groups

One approach to designing appropriate macroblock-to-slice group maps is to specify an *importance* parameter to a macroblock. Once an importance parameter is



defined, macroblocks can then be assigned to slice groups by a *classification* process using the chosen macroblock parameter. Many previous works have investigated on how to quantify different macroblock importance parameters. In (J.Lie, 2005), the error sensitivity of a macroblock is determined by computing a macroblock PSNR parameter. In (Dhondt, 2006), a macroblock *impact factor* is computed that depends on some information from the used pixels. In (N.Thomos, 2005), a distortion measure based on the mean square error of the original and reconstructed pixels are used. In (Im, 2005), a macroblock *importance factor* is computed based on two distortion measures, a distortion of the coded macroblock and a distortion if the macroblock is lost and concealed. In (Hantanong, 2005), the macroblock coded *bit-count* has been used as an indicator of macroblock importance derived from the *spatial* characteristics of the video sequence. In (Panyavaraporn, 2007), a distortion-from-concealment measure is used as an importance indicator derived from the *temporal* characteristics of the video. In this work we demonstrate the use of combined *spatial* and *temporal* indicators of macroblock importance to generate explicit FMO slice group maps for H.264/AVC wireless video transmission that works better than the predefined FMO map types. We also introduce the concept of *remapping* an initial macroblock-to-slice group map to obtain better performance from an error-concealment point of view.

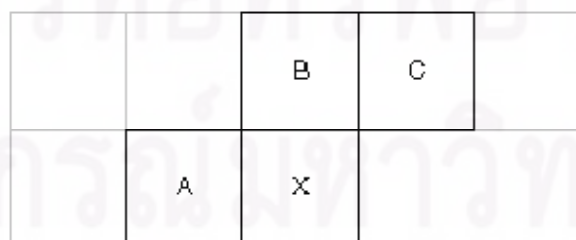
The other aspect of using explicit FMO, aside from the importance parameter, is the macroblock classification process. Methods of classifying the importance of macroblocks have not yet been thoroughly investigated. Different methods have been used with varying degrees of complexity. In (Hantanong, 2005) and (Panyavaraporn, 2007) a simple interleave sorting method is used; the idea is to simply spread the seemingly important macroblocks to different slice groups. In (Thomos, 2005), dynamic programming technique is used while (Lie, 2005), (Dhondt, 2006), and (Im, 2005) uses heuristic classification methods. In this work, without adding too much complexity we propose an improved sorting algorithm that makes a conscious effort to minimize the variations in the statistics of the chosen macroblock parameter. We show that by modifying the classification process modest quality improvements can be achieved.

The advantages of slice structured coding and utilizing a highly flexible macroblock interleaving tool such as FMO to improve error resiliency however comes with certain trade-offs. The main trade-off in using FMO is reduced coding efficiency and increased overhead bits. For purposes of error-resiliency, the macroblocks-to-slice grouping can be arranged such that macroblocks belonging to the same slice group are spatially scattered across the frame. This not only helps in limiting the effect of spatial error propagation but can also help improve error concealment by maximizing the number of macroblocks that are available for interpolation. Having a

scattered macroblock arrangement within the frame reduces coding efficiency for several reasons:

- Using FMO disables Intra- and Inter- prediction across slice boundaries, the motion vectors are restricted in a way that they refer only to decoded macroblocks in the same slice. Therefore, there is loss of coding efficiency due to constrained or dispersed search space.
- If the MBA map is allowed to change every frame, then a PPS header must be constructed and inserted to the bit stream. This result in increased overhead bits due to slice headers and PPS bits, therefore reducing the bits allocated for texture information.
- Efficiency of the entropy coding is reduced because FMO resets the codeword context, in case of Context Adaptive Binary Arithmetic Coding (CABAC) and Context Adaptive Variable Length Coding (CAVLC) at each slice.

Another important trade-off of using FMO that is significant with our research is the effect on motion vector encoding. Motion vectors of neighboring macroblocks are often correlated because object motion may extend over large regions of a frame. Video encoders exploit this correlation by encoding only the difference of the actual motion vector and motion vector prediction from the available neighboring macroblocks. But using FMO changes the encoding order of macroblocks. This reordering has an effect on the number macroblocks that will be available for motion vector prediction. If FMO is not used macroblock encoding follows a normal raster scan ordering starting from the upper leftmost macroblock then proceeding from left to right, then top to bottom. In most cases each macroblock (X) would have three neighboring macroblocks (A, B and C) available for motion vector prediction as shown in Figure 2.5. Because macroblocks A, B and C are already encoded before macroblock X, the neighboring motion vectors can be used to reduce the bits needed to encode the motion vector of macroblock X. The difference between the prediction and the actual motion vector, the *motion vector difference*, is encoded and transmitted.

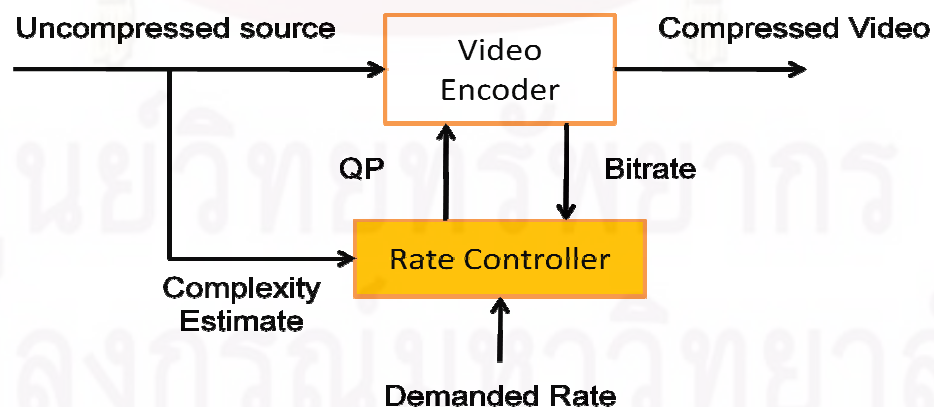


**Figure 2.5** The motion vector for macroblock X is predicted from neighboring macroblocks A, B and C

The trade-off of reduced coding efficiency and increased overhead bits in using FMO are felt most severely at low bit rates. Also, the adaptive rate control mechanism of H.264/AVC is not optimized to work with FMO such that the target bit rate is exceeded whenever FMO is enabled. In order to address this issue we proposed improvements on the frame layer rate control of H.264/AVC to better manage the rate whenever FMO is enabled.

## 2.4 The Importance of Rate Control

Compressed video are inherently variable rate mainly because of the variations in the texture and spatial information between frames and because video encoders can use different coding parameters for each picture. While most of today's wireless network infrastructure are designed towards constant bit rate transmission. One way to transmit compressed video over a constant bit rate channel is to use a FIFO (First In First Out) buffer at the output of the encoder to smooth out the variations in the bit rate and to match the required channel rate for successful transmission. The encoder puts bits in the buffer at a variable rate, while bits are drained from the buffer at a constant rate. To prevent the buffer from overflow or underflow, and to achieve good over-all video quality, a rate-control scheme is applied to dynamically adjust the encoder parameters in order to achieve the target bit rate. One method to control the bit rate is to vary the degree of quantization for each frame. By varying the quantization, a constant bit rate video stream can be achieved with variable PSNR per frame, resulting in a variable video quality per frame.



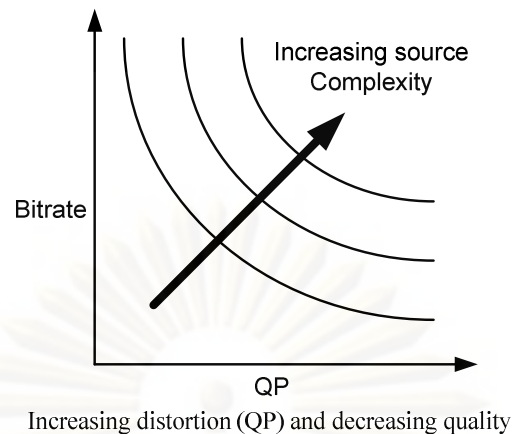
**Figure 2.6** Illustration of a Rate Control for H.264/AVC Video Encoder

An illustration of the rate control for H.264/AVC is shown in Figure 2.6. The rate of the compressed video is controlled by estimating the complexity of the frame and adjusting the QP parameter to achieve the desired rate. The QP must be dynamically adjusted such that the bit rate and the demanded rate are the same.

With the progress of video coding standardization and various video applications, extensive studies on different implementations of rate control have been done. The operations of the rate control are not specified in the standards so flexibility in implementing efficient rate control algorithms are left to the designers. The video standards group only issues non-normative guidance to aid in its implementation because it has become an essential component of any video encoding system. The performance of the rate-control algorithms to determine the optimal encoder parameters and at the same time satisfy the given application and transmission constraints has a large impact on the resulting video quality. A properly designed rate control algorithm can efficiently enhance the rate-distortion performance of the video coding system. Because of the importance of the rate control algorithm in the operation of the video encoder, we are motivated to improve the existing rate control algorithm in H.264/AVC in combination with FMO and wireless channel information.

The challenge in rate control is to determine a combination of suitable encoding parameters, such as quantization step size, macroblock mode, motion search, etc., that achieves the best overall video quality and at the same time satisfying the network and application constraints. Block-based hybrid video encoding schemes such as H.264/AVC achieve compression not only by removing truly redundant information but also by making small quality compromises in ways that are intended to be minimally perceptible. In particular, the *quantization parameter* (QP) regulates how much spatial detail is saved. When QP is very small almost all of the detail is retained. As QP is increased, some of that detail is aggregated so that the bit rate drops, but at the expense of increased *distortion* and some loss of quality. The rate and distortion relationship of a particular input picture is illustrated in Figure 2.7, lowering the bit rate can be achieved by increasing QP at the cost of increased distortion. The figure also illustrates that the rate-distortion relationship varies depending on the source complexity. The rate control algorithm also takes into consideration the buffer fullness, the available channel bandwidth and the video frame complexity in determining QP.





**Figure 2.7** Rate and Distortion relationship of hybrid video encoders

To achieve good video quality, one approach in rate control algorithms is to use mathematical modeling of the source statistics combined with rate-distortion optimization techniques to find the optimal quantizer that satisfies the given constraints (Ortega, 1994). In the H.264/AVC the assumption is that the source statistics follow a Laplacian distribution (Chiang, 1997) and apply Lagrange optimization techniques to find the optimal or nearly-optimal bit allocation for each frame or macroblock. One difficulty in this method is the computational complexity involved in finding the optimal solutions. With information such as current buffer fullness, frame complexity, bandwidth, coding mode, etc. mathematical formulas are used to determine the quantization parameter for rate control.

## 2.5 Rate Control in H.264/AVC

The H.264/AVC has adopted some new coding techniques such as rate-distortion (R-D) optimized motion estimation and mode decision, also called RDO, with various intra- and inter-prediction modes and support for multiple reference frames have significantly increases coding efficiency. This also makes the H.264/AVC rate control more complicated. In the H.264/AVC rate control, the quantization parameter is used in both the rate control algorithm and the RDO process. To perform RDO for each macroblock in the current frame a QP must first be determined for each macroblock by using the mean absolute difference (MAD) of the current frame or macroblock. However, the MAD of the current frame or macroblock is only available after the RDO process. The inter-dependency between RDO and rate control creates a “chicken and egg” dilemma (Li, 2003), and makes the H.264/AVC rate control more

challenging and more complicated than previous standards.

The H.264/AVC rate control uses a fluid flow traffic model and linear tracking theory to compute the target bits. The frame complexity, measured in terms of MAD, of the current frame is computed by using a linear MAD model from the previous frames to solve the “chicken and egg” dilemma. A quadratic rate-distortion model is used to compute the quantization parameter for RDO of each frame or macroblock.

A brief overview of the operations of the rate control as described in (Li, 2003) is presented here. The main objective of the adaptive rate control in the H.264/AVC is to compute the target bits and map the target bits to an appropriate QP. The adaptive rate control consists of three interrelated components.

1. **GOP level rate control.** Calculate the available bits for the remaining frames in the GOP and initialize the QP of the first Intra and Inter frame of the GOP.
2. **Frame level rate control.** The quantization step  $Q_{step}$  is computed using the quadratic rate model (Chiang, 1997) and then used to perform Rate Distortion Optimization (RDO) (Weigand, 2003) for each macroblock in the frame. The quadratic rate model is given by Eq. 2.1.

$$R = \frac{a_1 \times \tilde{S}}{Q_{step}} + \frac{a_2 \times \tilde{S}^2}{Q_{step}^2} - h \quad (2.1)$$

Where  $h$  is the number of bits for header and motion vectors,  $a_1$  and  $a_2$  are model parameters estimated by linear regression and  $S$  is the mean absolute difference (MAD) of the current stored picture which is predicted by a linear model according to the actual MAD of previous stored picture given by Eq. 2.2.

$$\tilde{S}_i = b_1 \times \tilde{S}_{i-1} + b_2 \quad (2.2)$$

Where  $b_1$  and  $b_2$  are the model parameters updated by linear regression. The corresponding QP is then calculated according to  $Q_{step}$ . Afterwards the number of frames that need to be skipped is calculated to meet the target rate.

3. **Basic Unit level rate control.** Operates in the same way as the frame level rate control. The basic unit is defined as a picture, slice, macroblock row or a set of macroblocks. A linear model is used to predict the mean absolute difference (MAD) of the current basic unit in the frame and a quadratic R-D model is used to calculate the QP which is used for RDO in the basic unit. The basic unit rate control is used to obtain a good trade-off in video quality and

bit rate.

The design of the adaptive rate control in the JM reference software still has some room for improvement in terms of better buffer managements and better frame complexity measures. The computation of the target bits relies on estimation of the frame complexity using a linear MAD prediction, from Eq. 2.2, of the previous frames. Since the prediction does not consider the complexity of the current frame, more often the MAD is not an accurate estimate of the frame complexity especially in complex sequences containing a lot of motion. The mapping of the target bits to the frame QP uses a quadratic model; and the number of bits allocated for texture depends on the computed target bits and the average header bits used in the previous frames. For low-bit rate applications and complex sequences the target bits and header bits are not accurately predicted, thus the resulting QP assignment for the current frame may not be optimal. Also the design of the rate control does not consider the overhead of using FMO; so whenever FMO is enabled the adaptive rate control cannot accurately meet the target bits.

Many studies have been done to improve the existing adaptive rate control in H.264/AVC but very few have addressed the issues associated with FMO and rate control. In (Wu, 2006), the optimal frame selection of using FMO is based on a rate-distortion analysis and rate constraints, but implemented with constant QP and variable rate. In (Ha, 2009), bit rate reduction is accomplished by classifying macroblocks into two slice groups with similar transform coefficient distributions, this however limits the usefulness of FMO to only two slice groups. In (Kannue, 2008), FMO is used to classify macroblocks according to a region of interest and assigns different QP to different slice groups. The approach taken in the studies mentioned modifies the FMO map to minimize the over head in bits; the rate control essentially remains the same.

In this work we propose enhancements to the H.264/AVC frame layer rate control regardless of the FMO mapping, using the explicit FMO map type, to better control the rate when FMO is enabled. The approach is similar to other studies that improve the H.264/AVC rate control. Several researches have been done to improve the frame complexity measure and QP adjustments to improve the rate control. In (Lee, 2006), (Jiang, 2005) and (Jiang, 2004) for example, enhancements to the existing MAD-based frame complexity have been proposed combined with QP

adjustment schemes to improve the frame layer H.264/AVC rate control. In (Kwon, 2007), a header bits model is proposed to better estimate the header bits for rate control.

The approach that we have taken combines several concepts from previous studies in improving the rate control and also considers extension to FMO related issues. The proposed improvements to the frame layer rate control of H.264/AVC are improved bit allocation by modifying the target bit using frame complexity measure, enhance the existing MAD complexity measure, a new header bits model and adjustment of QP with FMO considerations.

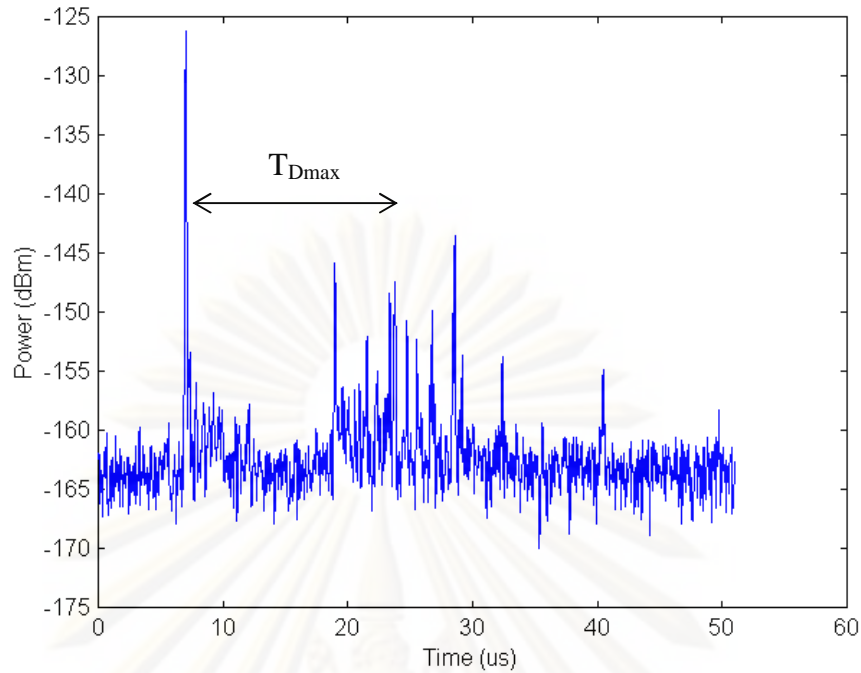
## 2.6 Wireless Channel Characteristics

In this section some important radio channel parameters are presented. The understanding of these parameters is essential to the understanding of the characteristics of the wireless channel.

### 2.6.1 Time Delay Spread and Coherence Bandwidth

The two parameters that are often used to characterize multipath wireless channels are the *time delay spread* and the *coherence bandwidth*. The time delay spread,  $T_{delay}$ , is a measure of the length of the impulse response for the multipath wireless channel. Time delay spread leads to inter-symbol interference (ISI) and degrades the performance of the wireless communication system. One way to define the impulse response of a multipath wireless channel is by its maximum time delay spread,  $T_{D,max}$ , defined as the range of delays over which the delay peaks of the channel power delay profile is not less than 30 dB from the peak of the first received pulse (Ngan, 2001). An illustration of a typical power delay profile of an indoor wireless channel showing the maximum time delay spread is shown in Figure 2.8.





**Figure 2.8** Maximum time delay spread of a typical indoor wireless channel

As a measure of fading correlation between frequencies, the coherence bandwidth is directly related to the time delay spread. For an exponentially distributed delay spread power profile, the coherence bandwidth is given as in Eq. 2.3

$$B_{coherence} = \frac{1}{2\pi T_{delay}} \quad (2.3)$$

Two frequencies lying within the coherence bandwidth are likely to experience correlated fading. Such a channel is frequency non-selective and all frequency components are subject to the same attenuation, phase shift and time delay. Narrowband wireless channels are usually characterized as a frequency non-selective channel, the bandwidth of the channel is sufficiently narrow that its frequency response can be considered “flat” or non-selective. It also implies that bandwidth of the transmitted signal is less than the coherence bandwidth of the channel.

On the other hand if two frequencies are separated by more than the coherence bandwidth, they are affected differently by the channel. When the coherence bandwidth is small compared to the bandwidth of the transmitted signal, the channel is frequency selective. This is usually the case for wideband communications systems where the signal bandwidth significantly exceeds the channels coherence bandwidth.

### 2.6.2 Fading, Coherence Time and Doppler Spread

Fading refers to the deviation in attenuation experienced by a carrier-modulated

signal across a certain propagation media. The fading may vary with time, position and frequency and is usually modeled as a random process. A *fading channel* is a communication channel that experiences fading. In wireless systems, fading can be due to multi-path propagation referred to as *multi-path fading* or due to shadowing of obstacles affecting wave propagation referred to as *shadow fading*. In a typical indoor wireless environment, signal reflections create multiple signal paths from the transmitter to receiver. Superposition of the received signal traversing different paths can lead to destructive interference causing multi-path induced fading.

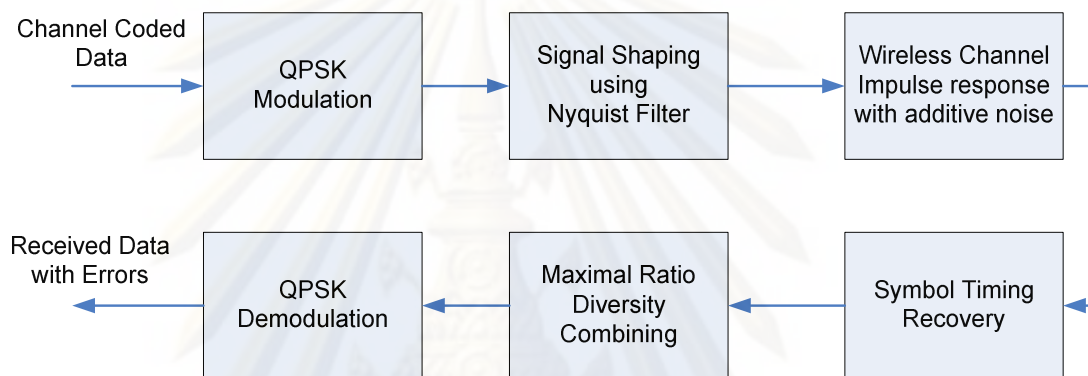
Slow and fast fading refers to the rate at which the magnitude and phase change imposed by the channel on the signal changes and is related to the coherence time. *Coherence time* is a measure of the minimum time required for the magnitude change of the channel to become uncorrelated from its previous value. Slow fading arises when the coherence time of the channel is relatively large compared to the delay constraints of the channel such that that amplitude and phase change imposed by the channel is roughly constant over the period of use. Fast fading occurs when the coherence time of the channel is small relative to the delay constraints of the channel; in this case the amplitude and phase change imposed by the channel varies considerably over the period of use.

Another channel phenomena associated with wireless communications is the *doppler spread*. Doppler spread may be interpreted as the measure of the variation of the shift in carrier frequency, or the measure of the rate at which the channel changes. Small doppler spreads imply a large coherence time,  $T_{coherence}$ , or a slowly changing channel. Wireless channel characterized by small Doppler shifts (0.3 – 6.1 Hz) are often called slow fading channels. If the wireless communication channel symbol duration,  $T_{symbol}$ , is large compared to the coherence time then the channel is subject to fast fading.

The change in the carrier frequency because of the doppler spread (or Doppler shift) is mainly due to the mobility of the receiver relative to the transmitter. A fast moving receiver, mobile user in a car, or train, will experience high doppler frequency shifts and hence subject to fast fading channel conditions. While a slow moving receiver, walking or in an elevator, are usually subject to slow fading channel conditions.

### 2.6.3 Wireless Channel Simulator

Wireless channel measurements under different possible environments can be difficult and time consuming. However, a wireless channel can be reasonably modeled using a multipath Rayleigh fading model. The effects of the channel under different environments can then be easily studied through computer simulations by varying the appropriate parameters. An example of the wireless channel simulator that can be used for this purpose is described in (Chen, 1995) with the block diagram as shown in Figure 2.9.



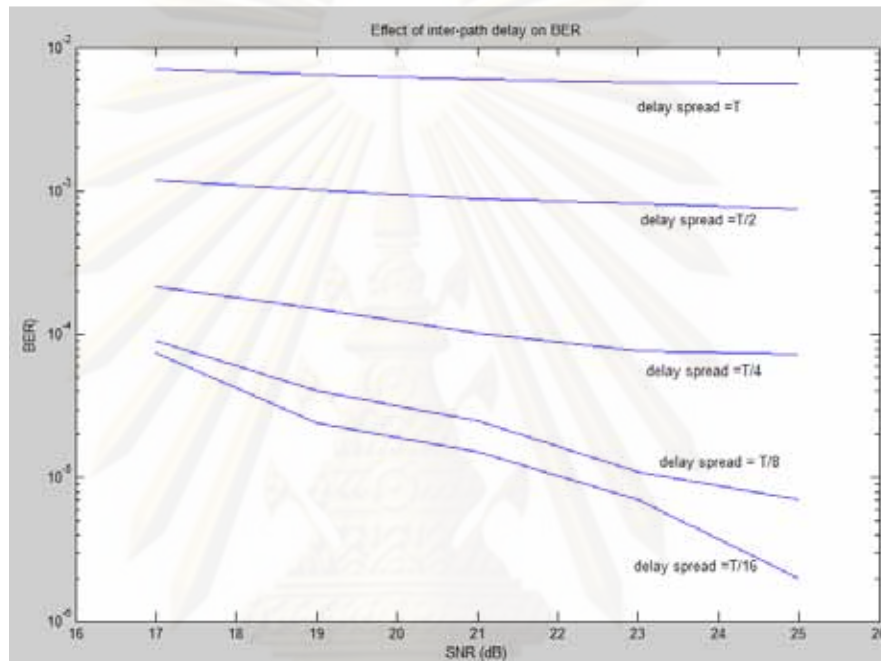
**Figure 2.9** Schematic of the wireless channel simulator

The parameters that can be changed in the simulator are the maximum doppler frequency ( $f_D$ ), the propagation power delay profile (modeled as  $n$ -rays with different inter-path delay and power), signal power and antenna diversity. The data rate is fixed at 256 kbps. A coherent receiver, with optimal symbol timing recovery and perfect carrier recovery is assumed. An ideal maximal ratio combiner for antenna diversity combining is used.

The wireless channel simulator is used to study the average bit-error rates (BER) at different signal-to-noise ratios (SNR) ranging from 17 to 25 dB, and the effects of different Doppler frequencies ( $f_D = 1\text{Hz}, 2\text{Hz}, 5\text{Hz}, 16\text{Hz}$  and  $40\text{Hz}$ ), inter-path delay ( $\tau=0, T/16, T/8$  and  $T/4$ ) and diversity (no diversity, two-branch and three-branch) are investigated.

The inter-path delay is due to the different distances traveled (multi-path) by the scattered rays received at the receiver. The effect of this delay is to increase the spread of the channel impulse response which leads to increased ISI and hence increases the BER as  $\tau$  is increased from 0 (flat fading) to  $\tau = T$ , where  $T$  is the symbol duration, as

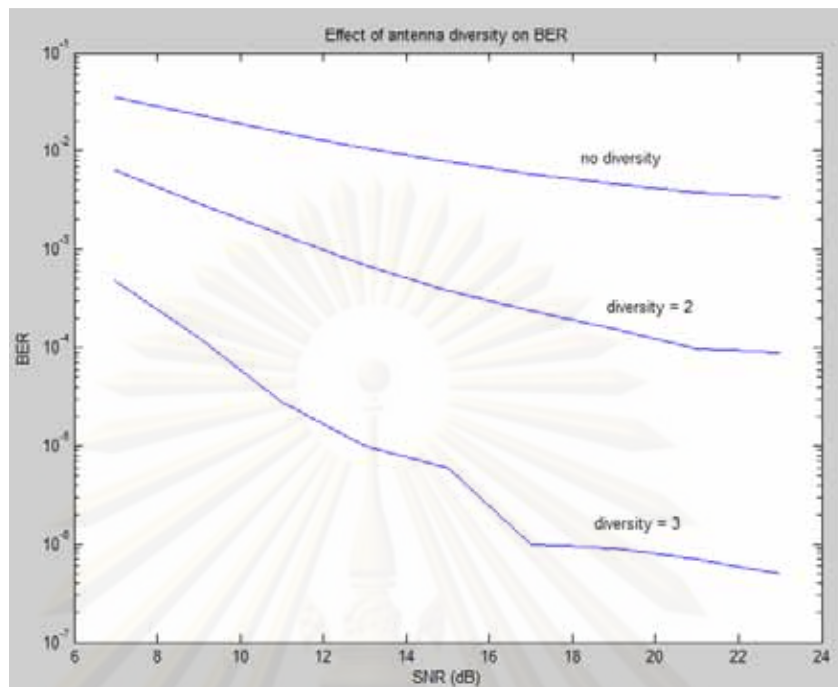
can be seen in Figure 2.10. One way to decrease the BER at a given inter-path delay is to increase the transmitted signal power as measure by the SNR. In Figure 2.10, increasing the SNR at  $\tau = T/4, T/8$  and  $T/16$  decreases the BER. Increasing the signal power also has limited effect for a certain value of  $\tau$ . For example, in Figure 2.10 at  $\tau = T/2$  and  $T$ , increasing the SNR does not significantly reduce the BER, this operating point is called *irreducible BER*.



**Figure 2.10** Effect of inter-path delay and SNR on BER

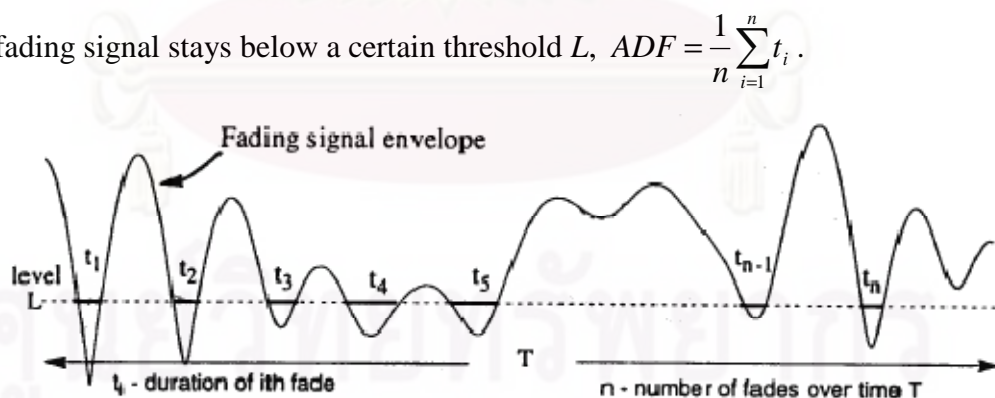
One way to improve the BER when  $\tau$  is at the point of having an irreducible BER even at increased SNR is to increase the diversity of the antenna. Increasing the number of receiver antennae that are spaced at a certain distance apart (antenna array) from each other is a well-known technique (space diversity combining) to combat the effects of multi-path fading. In the absence of antenna diversity, only one antenna at the receiver, the rate of fading is highly dependent on the doppler frequency. If there is more than one antenna at the receiver, the received signal strength at different antennae is relatively uncorrelated. The probability that the signals received in the antenna array being under fade decreases as the number of antenna in the array increases. A maximal ratio combining of the different signals received at the antenna array is a good strategy to reduce the BER. The BER of an  $N$ -array antenna (diversity  $N$ ) is of the order  $(P_e)^N$ , where  $P_e$  is the BER of one antenna. The effect of increasing the antenna diversity on the BER at  $\tau = T/4$  is shown in Figure 2.11.





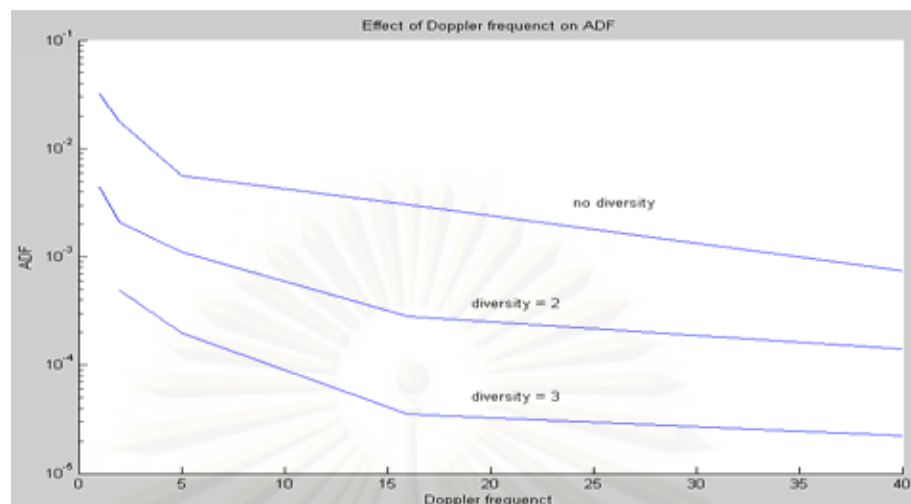
**Figure 2.11** Effect of antenna diversity on BER

The burst error nature of the wireless channel is related to the characteristic of the fading signal envelop. The level crossing rate (*LCR*) is the rate where the fading signal crosses a specified threshold  $L$  in a positive going direction.  $LCR = n/T$ , where  $n$  is the number of fades and  $T$  is the duration of the fade as illustrated in Figure 2.12. The average duration of fade (*ADF*) is a measure of the average amount of time the fading signal stays below a certain threshold  $L$ ,  $ADF = \frac{1}{n} \sum_{i=1}^n t_i$ .

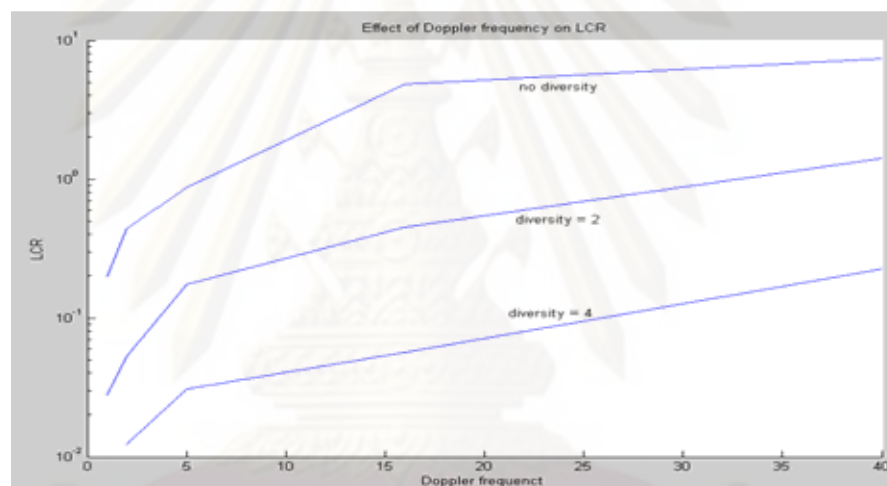


**Figure 2.12** Average duration of fade (ADF) and Level crossing rate (LCR)

The average BER is proportional to the product of the ADF and LCR at a given power level. The ADF is inversely proportional to the Doppler frequency  $f_D$ , as illustrated in Figure 2.13, at different antenna diversity. On the other hand, the LCR is directly proportional to  $f_D$ , as illustrated in Figure 2.14, at different antenna diversity.



**Figure 2.13** Effect of Doppler frequency  $f_D$  on ADF



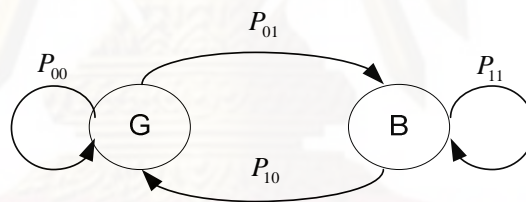
**Figure 2.14** Effect of Doppler frequency  $f_D$  on the frequency of error burst

Hence at a given power level, it is possible that  $f_D$  has no effect on the BER. However since the ADF corresponds to the length of burst errors and the LCR corresponds to the frequency of the burst, the Doppler frequency plays an important role in understanding the nature of errors in wireless channels. For small values of  $f_D$ , slow fading channel (slow moving receivers), the frequency of the fades are lower but the duration of the fades are longer. Thus error burst does not occur frequently in slow fading channel conditions but the errors burst have very long durations. For high values of  $f_D$ , fast fading channel (fast moving receivers), error burst occur more frequently but the duration of the error bursts are shorter. These effects are illustrated in Figure 2.13 and Figure 2.14, the figures also shows that increasing the antenna diversity can help reduce the number of times that the channel is under fade.

#### 2.6.4 Wireless Channel Prediction and Markov Modeling

Channel models have been extensively used in communication systems, for example to analyze the performance of link-layer protocols, and many different models have been used to describe the burst error nature of the wireless channel. To obtain accurate channel models requires significant computational complexity to capture the non-stationary characteristics inherent in wireless fading channels. Such models may not be suitable for low bandwidth and low-delay applications. The two-state Gilbert model is a tractable channel model have been used extensively and successfully in predicting channel condition in a number of different applications.

For this research we will use a two-state Markov model that is a simplified Gilbert-Elliot (Gilbert, 1960) channel model at the packet level; this model has been shown to be sufficient in modeling the burst nature of packet errors (Zorzi, 1996) (Aramvith, 2002). The model has two states, a good state ( $G$ ) and a bad state ( $B$ ) as shown in Figure 2.15. If the channel is in the good state, it is assumed that a packet can be transmitted successfully, and when the channel is in the bad state, a transmitted packet will experience some errors.



**Figure 2.15** Two-state Markov model of wireless channel

The channel state-transition probability matrix ( $P$ ) for this model is computed using Eq. 2.4, where  $P_{00}$ ,  $P_{01}$ ,  $P_{10}$  and  $P_{11}$  are the state transition probabilities.

$$P = \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix} = \begin{bmatrix} 1 - P_{01} & P_{01} \\ P_{10} & 1 - P_{10} \end{bmatrix} \quad (2.4)$$

From (Zorzi, 1997), the state transition probabilities,  $P_{01}$  and  $P_{10}$ , can be derived from the burst statistics of the channel, as in Eq. 2.5. The assumption is that the channel follows a Markov process having geometric distributions of run lengths error-bursts with mean  $1/P_{10}$ .

$$P_{10} = \frac{1}{M}; P_{01} = \frac{P_{10} * PER}{(1 - PER)} \quad (2.5)$$

In Eq. 2.5,  $M$  is the mean burst length in packets and  $PER$  is the packet error rate.

The packet error rate and mean burst length be computed from the error pattern generated by the wireless channel simulator, in order to model the burst error nature of channels with memory.

From the transition state probability matrix and the given initial state, with initial state probability ( $S_0$ ), we can calculate the expected channel throughput, i.e. the average probability of successful transmission in the next  $k$  packets according to the method in (Howard, 1971). The channel state probabilities ( $S_k$ ) at any discrete time interval  $k$  can be calculate from the initial state probability and the transition probability matrix as given by Eq. 2.6.

$$\begin{aligned} S_k &= S_0 * P^k \\ S_k &= [\pi_{k1} \quad \pi_{k0}] = S_0 * P^k \end{aligned} \quad (2.6)$$

Where  $S_0 = [1 \ 0]$ , if the channel initial state is good and  $S_0 = [0 \ 1]$  if the channel initial state is bad. The probability that the channel is in the good and bad state after  $k$  time interval is given by  $\pi_{k1}$  and  $\pi_{k0}$  respectively. The expected channel throughput ( $p_k$ ) in the next  $k$  packet interval can be computed by Eq. 2.7.

$$p_k = \frac{1}{i} \sum_{i=1}^k \pi_{i1} \quad (2.7)$$



## CHAPTER III

### **Error Resilient Video Coding in H.264 for Wireless Video Transmission**

In this work, we propose more efficient use of encoding tools available in the H.264.AVC encoder to improve error resiliency in wireless video transmission. Furthermore, the adaptive rate control is improved to minimize the impact of rate overhead.

We investigate the use of the explicit FMO as an error resilient tool in H.264/AVC for wireless video transmission. By exploiting the location of the macroblocks in the slice group, we compute a distortion-from-propagation measure as an indicator of macroblock importance. This macroblock importance parameter is based on the spatial and temporal characteristics of the video sequence, and introduces the idea of remapping an initial slice group map. Using a simple interleave sorting algorithm we generate explicit macroblock-to-slice group maps for FMO (STI-FMO) that is effective for wireless video transmissions. We also propose improvements to the macroblock classification process by adopting an improved sorting algorithm that minimizes the variance of a chosen macroblock parameter.

We propose enhancement to the H.264/AVC adaptive frame layer rate control that takes into consideration the trade-off associated with using explicit FMO. Also, a feedback based FMO selection scheme is investigated to make use of channel information in deciding the amount of FMO protection for a particular frame.

#### **3.1 Spatial-Temporal Indicator for Explicit FMO (STI-FMO) in H.264/AVC**

In this study we extend the previous works on FMO to propose a novel method, the STI-FMO (Cajote, 2007) that combines spatial and temporal indicators of macroblock importance. The method is based on the idea that if we have an initial macroblock-to-slice group map we can generate another slice group map that will have a lower *distortion-from-concealment* error. From an initial macroblock-to-slice group map a new *distortion-from-propagation* measure is computed using three parameters: 1) the individual *distortion-from-concealment* error of that macroblock, 2) the *distortion-from-concealment* error of the macroblocks in the same slice group and 3) the estimated probability that a macroblock will have an error within the slice.

Previous works on explicit FMO have investigated the use of the macroblock coded *bit-count* as a spatial indicator of macroblock importance (Hantanong, 2005) (Aramvith, 2006). Another indicator that was studied is based on the temporal characteristic of the video and on the error concealment method used at the decoder (Panyavaraporn, 2007). Hence, we call it as a *distortion-from-concealment* error, or simply distortion measure. This distortion measure and the bit-count measure will be used to derive a new macroblock importance parameter.

### 3.1.1 Effect of errors on FMO slice groups

In the current implementation of the H.264/AVC reference software that we used (JM 9.2); in using the explicit type of FMO we can only specify the assignment of a macroblock to a slice group. We cannot specify the order of the macroblock within the slice group; the current implementation of the encoder specifies that the macroblock within a slice must be arranged in ascending order of macroblock address (Wenger, 2002). The order of slice group transmission by default always starts with slice group 0, unless arbitrary slice group ordering (ASO) is specified. This information is useful in understanding the effect of error propagation in a slice when FMO is used.

Compressed video becomes vulnerable to transmission errors because of variable length coding. The operations of the encoder and decoder must be synchronized to successfully decode the bit stream. The decoder relies on the length of previously decoded symbols to decode the current symbol. A single bit error in the bit stream can cause loss of synchronization in the decoder such that succeeding symbols until the end of the slice will be in error. The decoder will regain synchronization at the beginning of the next slice, because a slice header is designed as a synchronization point. This has the effect of having a burst error, affecting several macroblocks due to a single bit error in the bit stream. Since we cannot control the arrangement of the macroblocks with the slice, we can only characterize its effect on error propagation. And effect of error propagation within a slice makes the macroblock at the beginning of the slice more important than the macroblock at the end of the slice.

### 3.1.2 Limitations of Bit-Count and Distortion Measure for Explicit FMO

In the previous works that use bit-count and distortion measure, the effect of error propagation within the slice has not yet been considered. Using bit-count and distortion measure for explicit FMO uses a two-pass encoding process. In the first

pass encoding, the video sequence is encoded without FMO and the bit-count and distortion measure of each macroblock is computed. The bit-count is the number of header and texture bits used to encode the macroblock, while the distortion depends on the type of error concealment used at the decoder. If we assume a non-motion compensated error concealment method is used at the decoder then the co-located macroblocks in the previous frame will be used as the reconstructed pixel value of the concealed macroblock in the current frame. During the decoding process of each macroblock, if a macroblock is in error or is undecodable it will be concealed using the co-located macroblock in the previous frame. The distortion incurred by error concealment,  $D_{CE}$ , is computed based on the sum of absolute difference (SAD) using Eq. 3.1.

$$D_{CE} = \sum_{(x,y) \in L} |f_k(x,y) - f_{k-1}(x,y)| \quad (3.1)$$

In Eq. 3.1, frame  $k$  and  $k-1$  are the current and previous frame,  $f(x,y)$  is the reconstructed pixel value at the coordinate  $(x,y)$  and  $L$  is area of the damaged MB.

The macroblocks with high bit count and high distortion values are classified as important macroblocks. A simple interleave sorting method is used to assign the macroblocks to slice groups. The macroblocks are sorted in descending order according to their bit count of distortion measure, and then macroblocks are assigned to different slice groups following the order of the sorted list. This is to spread the important macroblocks to different slice groups.

To illustrate the limitations when we do not consider the effect of error propagation, the distortion data for the 6<sup>th</sup> frame of Akiyo sequence is shown in Figure 3.16 for example.

0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	623	1118	566	0	0	0	0
0	0	0	12	1485	2305	1032	5	0	0	0
0	0	0	4	<b>4228</b>	<b>3362</b>	1627	283	3	0	0
0	0	0	0	1018	<b>3363</b>	1159	358	0	0	0
0	0	0	337	863	2868	2208	808	980	3	0
0	1	566	15	318	1589	1216	495	204	702	0
0	126	533	277	227	1573	782	607	250	929	0
0	156	542	151	6	753	1313	695	1127	332	0

**Figure 3.16** Distortion measure, 6<sup>th</sup> frame of Akiyo sequence

Applying interleave sorting method to the distortion data in Figure 3.16, the resulting macroblock-to-slice group assignment is shown in Figure 3.17. It can be seen that although macroblock address 37 has the highest distortion measure ( $D_{CE} = 4228$  in Figure 3.16) and is assigned to slice group 0, but because of the ascending order of macroblock address in the slice the macroblock with the highest distortion measure is the 5<sup>th</sup> macroblock in the slice. As a consequence, macroblock with address 4, 12, 23 and 36 are now more important than MB 37, because an error in these MBs can propagate until the end of the slice. This is also the case for the MB address 49 ( $D_{CE} = 3363$ ) and MB address 38 ( $D_{CE} = 3362$ ).

Slice group	Macroblocks
0	4 12 23 36 <b>37</b> 42 48 51 54 74 82 87 95
1	5 13 15 24 26 41 43 <b>49</b> 55 58 63 88 89
2	6 14 30 <b>38</b> 44 56 64 84 86 91 94 97 98
3	7 17 18 31 45 57 59 60 67 70 72 78
4	0 8 19 27 32 40 46 50 62 65 68 69
5	1 9 20 25 33 47 61 66 80 83 90 96
6	2 10 16 21 34 39 52 76 79 85 92 93
7	3 11 22 28 29 35 53 71 73 75 77 81

**Figure 3.17** Macroblock-to-slice group assignment, 6<sup>th</sup> frame of Akiyo

### 3.1.3 Combining Spatial and Temporal Indicators

The previous method, that uses bit-count and distortion measure, relies only on the individual characteristics of the macroblock and does not consider the effect of error propagation in a slice. In order to consider the effect of error propagation, we need to consider the location of the macroblock within the slice group and use this added information in computing a new *distortion-from-propagation* measure. The location of the macroblock within a slice is determined by an initial macroblock-to-slice group mapping, which can be derived using either bit-count or distortion measure. For this work we used as our initial map the slice group mapping derived from the distortion measure. The bit-count information, which is our spatial indicator, is used as a means to estimate the probability that a macroblock will be in error during transmission. These new set of information, the location of a macroblock in a slice group and the probability of error of a macroblock, are combined with the distortion measure of the macroblock within a slice group, to compute the *distortion-from-*



*propagation* measure and uses both spatial and temporal information, and is the macroblock importance parameter for the STI-FMO method.

### 3.1.3.1. Using the Initial Slice map

Because the encoder by default will arrange the macroblock within a slice in increasing macroblock address and considering the effect of error propagation on variable-length codes, the perceived importance of a macroblock now changes when it becomes a member of a slice group. The importance of a macroblock does not solely depend on the individual characteristics of a macroblock, such as bit-count or distortion, but on other macroblock in the same slice group. By first using an initial macroblock-to-slice group map, we can now determine a more accurate distortion-from-concealment error of a particular macroblock by taking into consideration its position in a slice group and the distortion measure of the other macroblock in the slice. This new distortion-from-concealment measure can now be used to generate a new slice group mapping.

### 3.1.3.2. Loss probability of a macroblock

First, we assume that single bit errors occur independently in the bit stream and we estimate the probability that the error will occur in a particular macroblock in a slice group. This assumption is valid because of the burst effect of a single bit error due to error propagation. The probability that the macroblock as position  $j$ , ( $MB_j$ ) will be in error in the event that a single bit error occurred during the transmission of one slice is estimated as using Eq 3.2.

$$\begin{aligned} P(MB_j \text{ error} | \text{single bit error}) &= \\ \frac{P(\text{single bit error} | MB_j \text{ error})P(MB_j \text{ error})}{P(\text{single bit error})} &\approx \binom{m}{k} \end{aligned} \quad (3.2)$$

In Eq. 3.2,  $m$  is the bit-count of  $MB_j$  and  $k$  is the total number of bits in the slice.

### 3.1.3.3. Distortion-from-propagation measure

Given the propagating effect of a single bit error to a macroblock that is assigned to a particular slice group and an estimate of the probability that a single bit error will affect a particular macroblock, we now compute a *distortion-from-propagation*,  $D_{prop}$ , measure that takes into consideration the initial slice group mapping using Eq. 3.3

$$D_{prop} = \sum_{i=j}^N (D_{CEi} \cdot P(MB_i \text{ error} \mid \text{single bit error})) \quad (3.3)$$

In Eq. 3.3,  $i$  denote the set of MBs in the slice group at position  $j$  up to  $N$ , where  $N$  is the last macroblock in the slice.  $D_{CEi}$  is the individual distortion of the macroblock computed using Eq. 3.1 and  $P(MB_i \text{ error} \mid \text{single bit error})$  is the probability the macroblocks will be in error as computed from Eq. 3.2.

The *distortion-from-propagation* measure using Eq. 3.3 are computed for all the macroblock in the frame, and the macroblock are also sorted in descending order and assigned to slice groups following the order of the sorted list to generate the new slice group map. One important issue at this point is that there is no guarantee that our new slice map will always perform better than our initial slice map. For this case we derive a single figure of merit to determine if the slice group that we derived using the *distortion-from-propagation* measure is better than our initial slice map. For simplicity we just compute the sum of distortion measure, using Eq. 3.3, for all macroblock in the frame using the initial map and the new slice group map. The slice map that will give a lower value will be used for that particular frame.

#### 3.1.3.4. Experimental Set-up

To analyze the effectiveness of using STI-FMO for wireless video transmission and to provide fair comparison with our other previous works, we used the same reference software, H.264/AVC JM 9.2. The decoder is modified to simulate non-motion compensated error concealment. The decoder simulates the effect of error propagation due to variable-length coding in the slice by discarding macroblocks starting from the macroblock that has error until the end of the slice, regardless of the coding mode of the macroblock.

Four standard video sequences are encoded: akiyo, foreman, claire and carphone, using the baseline profile at level 3.0. Each sequence is encoded for a total of 100 frames at a frame rate of 10 frames per second. Rate control is enabled and the bit rate is set at 32 kbps. The default encoder parameters are used with the exception of the FMO related parameters; see (Michael, 2004) for detailed information about the H.264/AVC JM encoder parameters.

To investigate the benefits of using FMO on wireless channel, the Rayleigh fading wireless channel simulator as described in the previous chapter is used in our simulation. The details of the simulator can be found in (Chen, 1995). To simulate the

effects of slow and fast fading channels, the maximum Doppler frequency parameter is set to 1 Hz, for slow-fading and 40 Hz, for fast-fading, respectively. The average bit error rate (BER) and average packet error rate (PER) for an 80-bit packet frame, are 0.06 and 0.09, respectively both for fast and slow fading channel conditions.

The PSNR and the total number of undecodable macroblock were used as the performance metrics to qualify the effectiveness of STI-FMO. PSNR is an objective measure of video quality, while the number of undecodable macroblocks is a quantitative measure of performance. Minimizing the number of undecodable macroblocks indicates robustness to transmission and propagation errors, but it does not necessarily translate to higher PSNR, because not all macroblocks have the same effect on the PSNR. The performance of the STI-FMO is compared with other works that uses the spatial and temporal indicator separately

#### 3.1.3.5. Results and Analysis

The following results compared the performance of our proposed STI-FMO with our works using explicit FMO. This is the only fair comparison to evaluate our proposed method since the previous works (Hantanong, 2005) have provided comparative results with the standard H.264/AVC FMO schemes.

Table 3.1 and Table 3.2 gives a comparison of the number of undecodable macroblock and average PSNR of the test video sequences under slow and fast fading conditions. The results show that using STI-FMO can effectively reduce the number of undecodable MBs under fast fading conditions. The *distortion-from-propagation* model correlates well with the error propagation effects of burst errors under fast fading conditions.

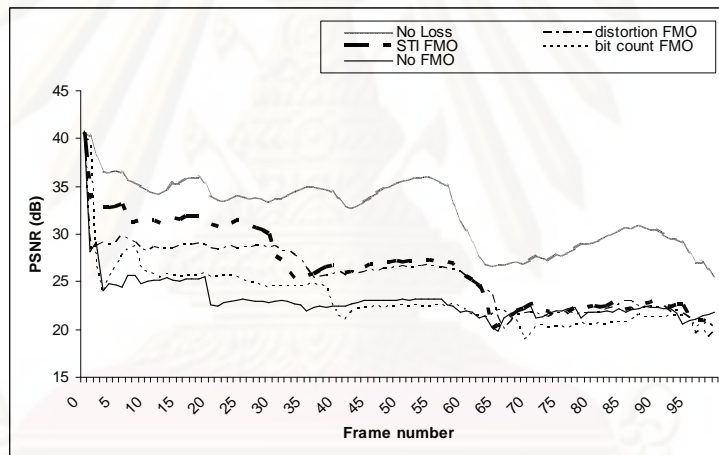
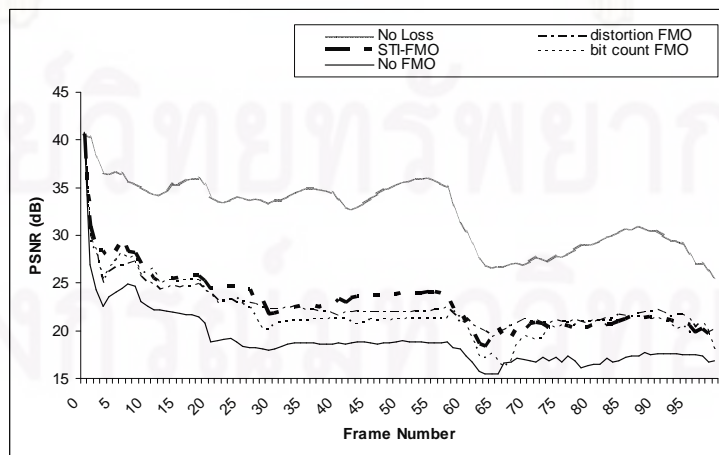
**Table 3.1** Comparison of Number of Undecodable Macroblocks

Loss macroblock	Akiyo		Foreman		Claire		Carphone	
	slow	fast	slow	Fast	slow	fast	slow	Fast
No FMO	1547	5367	1499	5740	1346	5269	1380	5634
Bit count	758	1210	<b>712</b>	1243	<b>780</b>	1257	742	1329
Distortion	<b>701</b>	1210	746	1187	833	1182	651	1202
STI-FMO	750	<b>1153</b>	734	<b>1185</b>	833	<b>1147</b>	<b>622</b>	<b>1176</b>

**Table 3.2** Comparison of Average PSNR (dB)

PSNR (dB)	Akiyo		Foreman		Claire		Carphone	
	slow	fast	slow	fast	slow	fast	slow	Fast
No FMO	30.71	27.63	17.11	15.12	30.82	24.31	23.04	19.01
Bit count	34.46	32.17	20.61	18.19	31.12	<b>30.07</b>	23.29	21.94
Distortion	34.75	31.58	<b>20.62</b>	<b>19.81</b>	30.66	29.59	25.49	22.62
STI-FMO	<b>35.03</b>	<b>32.37</b>	20.34	19.40	<b>31.79</b>	29.89	<b>26.45</b>	<b>23.06</b>

From the comparison of the average PSNR, the STI-FMO helps improve visual quality by reducing the number of undecodable macroblocks. But having a lower number of undecodable MBs does not directly translate to having lower average PSNR, because not all MBs have the same effect on the PSNR. A plot of the PSNR for carphone sequence under slow and fast fading is show in Figure 3.18 and Figure 3.19 respectively.

**Figure 3.18** Comparison of PSNR using carphone sequence under slow fading**Figure 3.19** Comparison of PSNR using carphone sequence under fast fading



### 3.1.3.6. Summary

The method of using both spatial and temporal information, the STI-FMO, to quantify the importance of a macroblock for use with explicit FMO has been shown to be effective in mitigating errors in wireless transmission of H.264/AVC video streams. The proposed method performs better than the works that use bit-count and distortion measure in terms of average PSNR and number of undecodable MBs. Using the new *distortion-from-propagation* measure gives a more accurate assessment of the importance of a macroblock from an error propagation point of view as compared to using the individual distortion measure. From the given initial slice group map, the proposed method takes into consideration the position of a macroblock within a slice group and uses this information to generate a new slice map that has the potential to perform better than the initial map.

The next step in the development of an error resilient video coding using explicit FMO is to improve the macroblock classification process, other than the interleave sorting method, to further improve the performance.

## 3.2 Improved Sorting Algorithm for Explicit FMO in H.264/AVC

This study focuses on the macroblock classification process of using FMO in designing macroblock-to-slice group maps in H.264/AVC. The motivation is to improve on the previous works that use interleave sorting algorithm, but also to make the classification process not too complex. The interleave sorting process used in (Hantanong, 2005) (Panyavaraporn, 2007) makes no consideration on the nature of macroblock importance parameters used, or on the effect of the classification on the resulting slice group map, we propose to make improvements on this aspect of the classification process.

### 3.2.1 Independence of Slice groups in Slice Structured Coding of H.264/AVC

The concept of using slices as a tool to stop spatial error propagation, it also promotes the concept of independence among slice groups. We are motivated to find a macroblock classification process that also adheres to the concept of slice independence. The interleave sorting method used in previous studies takes no consideration about the macroblock importance parameter or on the effect of the assignment on the slice groups.

Consider the case of using the macroblock coded bit-count as a parameter for explicit FMO, then by using interleave sorting method the resulting slice groups will have variable lengths. If the probability of a random bit error occurring in the bit stream is assumed to be independent. Then longer (more bits) slice group will naturally have a higher probability of error compared to a shorter (less bits) slice group. One way to improve the macroblock classification process is to assign macroblocks to slice groups in such a way that the number of bits representing the slice groups (the total bit-count of the macroblocks in the slice) should be more or less equal across all groups.

Consider another case where distortion,  $D_{CE}$ , is used as a parameter for explicit FMO. A certain distortion is incurred in the event that a MB is deemed undecodable due to transmission errors. If somehow we desire that the loss of any particular slice group due to transmission error will have more or less the same effect on the perceived distortion of the decoded output signal. Then the classification process must be designed to classify the MB into slice groups such that the total,  $D_{CE}$ , measure of the MBs within the slice must be equal among different slice groups.

### 3.2.2 Divide and Conquer Approach to Classification Process

The two cases mentioned earlier outlines the design objective of the classification process of assigning macroblocks to slice groups. In theory, we want to be able to classify macroblocks into slice groups that satisfied a certain criteria. To simplify the problem, we want to satisfy the criteria that the sum (or mean) of the chosen parameter (bit-count or distortion) must be more or less the same for all slice groups. Although we have defined a simple criterion, the task of finding the optimal grouping that will satisfy this criterion is still very difficult.

Consider the case of a QCIF resolution video input (144 x 176 pixels per picture), segmented into 16x16 pixel macroblocks, this will result in 99 different macroblocks for the QCIF image and each macroblock has either a bit-count or distortion parameter. For the simple case that we want to group the 99 different MBs into two slice groups such that the total number of bits (or distortion) in each group are equal is not as straight forward. There are many possible groupings that can satisfy the requirement of having equal number of bit per slice. In that case we can have several possible solutions, and how to choose among the possible solutions poses another problem. Compounded by the fact that we need to be able to group the macroblock

into eight different slice groups in order to be effective for FMO. The complexity of the grouping problem increases as the number of objects to group increases, but for our cases we restrict ourselves to QCIF video sequences so that we only have 99 different macroblock to process. Although the proposed solution will still work with any arbitrary number of macroblocks, having 99 different macroblocks to group is still a large number. The complexity of the problem also increases as the number of groupings is increased, for our case we want to be able to assign macroblocks of up to eight groups.

Finding the optimal grouping of 99 different macroblocks into eight slice groups such that the number of bits in each slice are more or less equal, is the ideal problem that we want to solve. An elegant algorithmic solution that is simple and tractable is needed since the brute force iterative method of grouping (and checking if the criterion is satisfied) is not advisable even with the use of sophisticated computer, because of the large number possible groupings. To make the problem tractable and solvable, we have to make simplifications on the requirements of the problem.

First, to be able to satisfy the requirement of assigning macroblocks to eight groups that satisfies the equal number of bits (or whatever parameter is available) per slice criteria, we must be able to satisfy the requirement for the simple case of having only two groups. That is we want to be able to classify MBs into two slice groups (group A and B) such that the number of bits in each slice are more or less equal. If we can solve this simplified problem, we can apply the same solution to each of the groups (group A and B) in order to have four groupings. And then apply the same solution again to each of the four groupings in order to satisfy the requirement of having eight groups. This is a divide-and-conquer approach, and it makes the solution for the case of having two groups scalable to the problem of having  $N$  groups, where  $N$  is a multiple of 2. Thus, if we can obtain an optimal solution for the classification problem with two groups, we can obtain an optimal or near optimal solution for the case of eight groups. So our classification problem is simplified to the case of having two groups.

Second, if finding the optimal solution is exhaustive and difficult, we can opt for a simpler and tractable sub-optimal solution. For the case of our two group classification problem, we want to assign macroblock into two different slice groups such that the number of bits in each slice is equal. If it is difficult to satisfy the “equal number of bits” criteria, we can opt to find a sub-optimal grouping that only

approximates the equality criterion. This sub-optimal classification problem is the basis for the modified sorting algorithm for assigning macroblocks to eight slice groups.

### 3.2.3 Improved Sorting Algorithm for FMO

At this point, our objective is to group the  $M$  number macroblocks (where  $M \in \mathbb{I}$ ,  $8 \leq M \leq 99$ ) into two (group A and B) slice groups. The lower bound for  $M$  is eight since we want to classify the macroblocks into eight slice groups so that we have at least one macroblock per slice group. The upper bound for  $M$  is 99 for a QCIF video sequence input. We want the two slice groups, A and B, to have approximately the same number of bits per slice.

The proposed approximate solution to the problem is a modification of the sorting algorithm. Instead of just assigning MBs in the sorted list to slice groups alternately in an interleaving fashion we impose a rule on how to assign the current MB being classified to either group A or B. The assignment rule is that the current MB being classified will be assigned to a slice group with the lower total MB bit count.

The algorithm is outlined as follows (Cajote, 2008):

1. Sort the macroblocks in descending order of bit count. Macroblocks with high bit counts must be on the top of the list, and macroblocks with lower bit-count at the bottom of the list
2. Assign the first two elements of the sorted list as the first elements of Group A and Group B and remove them from the sorted list
3. Compute the total bit count in Group A and B, denoted as  $SumGrpA$  and  $SumGrpB$ .
4. Classify the macroblock that is currently at the top of the list, denoted as  $curMB$ , (the unassigned macroblock with the highest bit count) to either Group A or Group B
5. IF ( $SumGrpA < SumGrpB$ ),  
     then  $curMB$  is assigned to Group A  
     ELSE  
      $curMB$  is assigned to Group B
6. Remove the classified macroblock from the sorted list
7. Repeat (Steps 3) to (Step 6) until all macroblocks in the sorted list have been processed

By continually assigning the macroblock with the next highest bit-count to the slice group that has the lower running total of bit-count, the resulting slice group maps (group A and B) will have total number of bits to be approximately equal or very



close. Using this modified sorting algorithm the difference in the resulting number of bits between the two slice groups is small when compared interleaving method of macroblock assignment.

The modified sorting algorithm provides an approximate solution to the classification problem with two groups. The same modified sorting algorithm can be applied to the macroblocks belonging to group A and B, to obtain four slice groups, and applying again will assign macroblocks to eight slice groups.

Some other enhancements to the modified sorting algorithm have been included to deal with macroblocks with 0 bit count; these macroblocks must be equally distributed to the eight slice groups, so as to balance the number of macroblocks within the slices.

An example of the MBA map of the 3<sup>rd</sup> frame of the carphone sequence using bit count and the improved sorting algorithm is shown in Figure 3.20(a), and the MBAmapping of using bit count and the interleave sorting algorithm is shown in Figure 3.20(b).

3	1	6	7	4	3	2	4	5	7	5
0	0	3	2	4	7	0	5	2	6	3
3	0	2	2	5	2	6	4	2	1	0
2	6	7	3	5	7	4	4	3	5	1
3	1	5	2	1	6	7	6	6	0	1
5	2	7	2	6	4	5	7	1	1	1
6	0	7	4	4	1	5	3	0	7	3
7	0	7	3	5	5	0	1	6	2	6
4	4	0	4	1	7	3	6	3	5	0

2	3	5	4	2	6	5	2	6	4	6
7	2	3	7	6	0	7	2	1	4	1
1	3	3	1	7	0	2	6	5	5	7
5	4	2	5	2	1	6	1	5	6	3
7	7	0	3	3	4	2	4	4	0	7
0	5	2	1	0	6	0	4	5	3	7
1	5	4	6	0	3	6	3	7	0	1
0	1	0	1	2	2	1	3	4	5	0
4	6	7	3	4	0	5	6	1	2	7

(a)
(b)

**Figure 3.20** From the 3<sup>rd</sup> frame of carphone sequence. (a) The MBA map using bit-count and the improved sorting algorithm, (b) the MBA map using bit-count and interleave sorting algorithm

### 3.2.4 Results and Analysis

We compared four different video sequences and used bit-count and  $D_{CE}$  measure as the parameters to classify the macroblock for explicit FMO in H.264. We compared the performance of the improved sorting algorithm with the interleave sorting algorithm used in previous works.

First, we gathered the macroblock statistics, mean and standard deviation, using the interleave sorting algorithm and compared the macroblock statistics when using the proposed improved sorting algorithm. For each frame, we compute the sum of the bit count and  $D_{CE}$  measure per slice group. We then compute the mean and standard deviation of the bit count and  $D_{CE}$  measure per frame, and then we compute the mean and standard deviation of the two parameters for all frames in the video sequence. We compared the statistics of the bit-count and  $D_{CE}$  measure using interleave sorting method and the proposed improved sorting algorithm. The summary of the statistics for four different video sequences is shown in Table 3.3.

**Table 3.3.** Summary statistics of bit-count and  $D_{CE}$  using simple sorting and the proposed modified sorting algorithm

Video Input	Stat	Interleave Sort		Improved Sort	
		Bit count	$D_{CE}$	Bit count	$D_{CE}$
Akiyo	mean	393.9	2969.9	390.11	2957.1
	std	150.71	1029.7	<b>16.24</b>	<b>242.18</b>
Foreman	mean	424.9	38765	390.71	36418
	std	104.51	2184.8	<b>3.305</b>	<b>121.72</b>
Claire	mean	392.9	4243.3	390.42	42374
	std	138.45	1336.5	<b>6.032</b>	<b>156.73</b>
Carphone	mean	397.96	16764	390.11	16505
	std	112.5	2399.4	<b>3.04</b>	<b>196.69</b>

From Table 3.3, we can observe that using the improved sorting algorithm, a conscious effort is made to decrease the variations in the bit-count and  $D_{CE}$  per slice group in each frame. This is clearly evident in the large reduction of the standard deviation, with almost the same mean, of the bit count and  $D_{CE}$  measure when we compared the interleave sorting algorithm and the proposed improved sorting method.

Second, we simulate the transmission of the video sequences over a wireless channel by using a wireless channel simulator, using FMO as an error resilient tool to combat transmission errors. We compared the performance of two macroblock classification algorithm, the interleave sort and the improved sorting algorithm, by using different explicit FMO maps. Both slow and fast fading channel conditions are

tested, so we have a total of eight different test cases. The average PSNR and number of undecodable macroblocks were used as performance metrics. Table 3.4 summarizes the results of the experiment. The columns labeled as *IntSort* (Interleave Sort) shows the number of undecodable macroblocks and average PSNR in dB of the four video sequences under slow and fast fading channel conditions using the interleave sorting algorithm. The columns labeled as *ImpSort* (Improved Sort) shows the same but using the proposed improved sorting algorithm.

**Table 3.4** Summary of MB loss count for bit count and Average PSNR for Distortion

Video Input	Bit count		Distortion	
	MB Loss Count		Ave. PSNR (dB)	
	IntSort	ImpSort	IntSort	ImpSort
Akiyo slow	758	<b>693</b>	34.49	<b>34.59</b>
Akiyo fast	1210	1272	32.63	31.59
Foreman slow	712	728	20.1	<b>20.43</b>
Foreman fast	1243	<b>1227</b>	17.72	<b>18.47</b>
Claire slow	780	<b>763</b>	30.66	30.3
Claire fast	1257	1292	29.59	29.31
Carphone slow	742	<b>669</b>	25.49	<b>26.44</b>
Carphone fast	1329	<b>1200</b>	22.62	<b>23.24</b>

From the summary Table 3.4 it can be shown that using the proposed improved sorting algorithm can help reduce the number of undecodable MBs in five out of eight video sequences tested if the bit count parameter is used for FMO. This follows from intuition that by making the slice groups have more or less equal number of bits, they have equal probability of having an error. This makes the effect of the random error uniform across all slices. Compared with the case where the number of bits in each slice is not taken into consideration using the interleave sorting algorithm, then slice groups with higher number of bits are have higher probability of error, and thus more bits can be affected by an error due to the effect of error propagation. Using the proposed improved sorting algorithm somehow reinforces the nature of slice structured coding by making the slice groups more or less equally prone to random bit errors.

Also from Table 3.4, it can be shown that using the proposed improved sorting algorithm and  $D_{CE}$  as the parameter for FMO can help improve the average PSNR in five out of eight video sequences tested. This result also follows intuition, by using

the modified sorting algorithm; slice groups will have more or less the same contribution to the distortion at the output of the decoder due to the non-motion compensated error concealment process. Then the expected drop in PSNR will be more or less equal if any particular slice group is lost, this makes the effect on the distortion and PSNR uniform across all slices. Compared with the case where the total distortion contribution of the slice group is not taken into consideration, then slice group with higher total distortion will contribute more to the degradation of the output (and PSNR) compared to other slice groups. So using the proposed improved sorting algorithm also reinforces the independence of slices in slice structured coding by making the distortion contribution of each slice more or less equal across all slice groups.

### 3.2.5 Summary

An improved sorting algorithm has been presented that attempts to promote the independence of slice groups in adherence to the principles of slice structured coding. The proposed algorithm does not attempt to find the optimal grouping of the macroblocks into slice groups following a certain criteria, but approximates the optimal solution and uses a divide-and-conquer approach to make the solution scalable and tractable for eight slice groups. The proposed improved sorting algorithm that considers the uniformity of the given macroblock parameter (bit count or  $D_{CE}$  measure) in assigning macroblocks to slice group maps have been shown to provide modest gains in PSNR and number of undecodable macroblocks compared to the interleave sorting method of macroblock classification.

The proposed sorting algorithm for macroblock classification not only applies to the bit count and  $D_{CE}$  parameters used in the experiments but can be extended to any macroblock parameter available that can be used for explicit FMO in H.264/AVC.



### 3.3 Improved FMO-based Frame Layer Rate Control for H.264/AVC

We discussed in the previous chapter the trade-offs in coding efficiency and overhead bits when using FMO. The trade-offs are most obvious at low bit rates where the header bits occupy a larger portion of the total bit budget compared to the source bits. Thus, when using FMO at low bit rates, when error rates are high and bandwidth is limited, careful consideration of the trade-off is essential.

The approach taken by other researchers modifies the FMO map to minimize the effect of the overhead bits; the rate control essentially remains the same. In this study we take a more pro-active approach by proposing enhancements to the H.264/AVC frame layer rate control regardless of the FMO mapping to better control the rate when FMO is enabled.

#### 3.3.1 Proposed Header Bits Model

Motion vectors of neighboring MBs are often correlated because object motion can extend over large regions in the frame. In H.264/AVC, this correlation is exploited by computing a motion vector prediction from the macroblocks on the left, upper and upper-right location of the current macroblock being encoded, since the motion vectors of these macroblocks are already known in a normal raster scan order. The motion vector difference between the prediction and the true motion vector of the current macroblock is then encoded and transmitted. However, using FMO for purposes of error resiliency, the macroblock ordering can be scattered to minimize the effect of error propagation. In most cases neighboring macroblocks are not available for inter-prediction if they belong to different slice groups. This affects the computation of the motion vector difference and hence affects the coding performance. In this paper, we analyze the relationship of the motion vector difference and the number of slice groups to develop a new header bits model that performs well when FMO is enabled.

Previous studies investigated the use of motion vectors to model header bits for purposes of rate control. In (Kwon, 2007), the motion vectors have been used to model the number of header bits of inter-MB and intra-MB. The total number of header bits for the frame is computed as the total number of header bits of inter- and intra-macroblocks. This has been shown to be an effective and accurate model for header bits when FMO is not used. But when FMO is enabled with different number

of slice groups the model in (Kwon, 2007) is no longer accurate, since using FMO greatly affects the motion vector difference and not the actual motion vector.

The header bits model in (Kwon, 2007) for inter-MB uses the number of motion vectors ( $N_{nzMVe}$ ) and the number of non-zero motion vectors ( $N_{MV}$ ) gathered from the first pass encoding as shown in Eq. 3.4, where  $\gamma$  and  $\omega$  are model parameters.

$$R_{hdr,inter} = \gamma(N_{nzMVe} + \omega \cdot N_{MV}) \quad (3.4)$$

It has been observed by comparing the number of non-zero motion vector difference in several video sequences encoded with different QP at different number of FMO slice groups, the number of total non-zero motion vector difference increases as the number of FMO slice group used increases. In Table 3.5, shows the total number of non-zero motion vector difference for the carphone sequence and the corresponding percentage increase as the number of FMO slice groups increase. The table shows that at for the carphone sequence at QP=32, using FMO with eight slice groups, the percent increase in the number of non-zero motion vector difference is about 16%. This shows that the number of non-zero motion vector difference can be used to model the increase in overhead bits when FMO is enabled. The numbers of non-zero motion vector difference of the other sequences tested are provided in the Appendix A-1.

**Table 3.5** Number of non-zero motion vector difference at different QP and different number of FMO slice groups for carphone sequence

Carphone					% Increase		
QP	NoFMO	FMO2	FMO4	FMO8	FMO2	FMO4	FMO8
8	60552	65613	67828	70269	8.36	12.02	16.05
16	46192	48123	48969	49604	4.18	6.01	7.39
24	30549	32648	33266	33761	6.87	8.89	10.51
32	16134	17480	18161	18739	8.34	12.56	16.15
40	6636	7586	7876	8178	14.32	18.69	23.24
48	1438	1666	1693	1778	15.86	17.73	23.64

In order to address the effect on the loss of coding efficiency when using FMO due to the reduced availability of macroblocks for inter motion prediction. We adapt the model in Eq. 3.4 to model the header bits of  $P$ -frames. The total number of non-zero motion vector difference ( $N_{nzMVD}$ ), the total number of motion vectors ( $N_{MV}$ ) and the number of slice groups ( $num\_slice$ ) for a particular frame is used to model the frame header bits ( $H_{Pframe}$ ) as shown in Eq. 3.5, where  $\alpha_1$  and  $\alpha_2$  are model

parameters. In this case, the effects of intra-macroblocks are not considered since the header information includes only the macroblock modes, they are not crucial to the accuracy of the model.

$$H_{Pframe} = \alpha_1 N_{nzMVD} + \alpha_2 (N_{MV} + num\_slice) \quad (3.5)$$

By using the number of non-zero motion vector difference and including the effect of slice header overhead in the prediction of the frame header bits, we were able to obtain a more accurate header bits model. In order to compare the accuracy of the two models the  $R^2$  parameter is computed, when  $R^2$  is close to 1 then the model data correlates well with the actual experimental data. Several QCIF video sequences were encoded with QP values from 8 to 40, frame rate of 10 fps for a total of 100 frames using different number of FMO slice groups and the average  $R^2$  value is computed. A comparison of the  $R^2$  values between the header models in (Kwon, 2007) using Eq.3.4 and our proposed model using Eq.3.5 is shown in Table 3.6. The columns labels indicate the number of FMO slice groups, i.e., using 2, 4 and 8 slice groups are designated as FMO2, FMO4 and FMO8 respectively. The proposed model has higher  $R^2$  values compared to the model in (Kwon, 2007) and thus can more effectively model the number of header bits when FMO is used. The comparison of  $R^2$  for different QP and different rate is provided in Appendix A-2.

**Table 3.6** Comparison of average  $R^2$  values between the model in (Kwon, 2007) and the proposed modified header bits model.

$R^2$	NoFMO		FMO2	
	Proposed	(Kwon, 2007)	Proposed	(Kwon, 2007)
Akiyo	0.798	0.785	0.806	0.774
Carphone	0.917	0.882	0.922	0.887
Claire	0.843	0.820	0.856	0.827
Foreman	0.753	0.668	0.715	0.607
	FMO4		FMO8	
Video	Proposed	(Kwon, 2007)	Proposed	(Kwon, 2007)
Akiyo	0.787	0.665	0.756	0.245
Carphone	0.931	0.901	0.937	0.907
Claire	0.854	0.789	0.842	0.634
Foreman	0.738	0.658	0.750	0.668

### 3.3.2 Proposed Frame Complexity Measure

The current implementation of the rate control algorithm in the JM reference software follow the adaptive scheme as described in JVT-G012r (Li, 2003). There is however some limitations on the adaptive rate control algorithm and improvements have been proposed by several researchers. In this study we improve the frame layer rate control by improving the estimate of the frame complexity and proposing QP adjustment schemes for the current frame.

We have shown previously that the number of non-zero motion vector difference is a useful parameter to model the header bits. And that the amount of motion vector information is also correlated with the complexity of the frame and consequently to the amount of bits used for the texture and motion information. Following the framework in (Jiang, 2005) (Jiang, 2004), we compute the ratio ( $N_{nzMVDratio,i}$ ) of the number of non-zero motion vector difference ( $N_{nzMVD,i}$ ) in the  $i^{th}$  frame and the average of non-zero motion vector difference of all previously coded frames as shown in Eq. 3.6.

$$N_{nzMVDratio} = \frac{N_{nzMVD,i}}{\frac{1}{(i-1)} \sum_{j=1}^{i-1} N_{nzMVD,j}} \quad (3.6)$$

The MAD ratio ( $MAD_{ratio,i}$ ) is computed as the ratio of the predicted MAD of the current frame ( $MADP_i$ ) to the average MAD of all previously coded  $P$ -frames in the GOP using Eq.3.7.

$$MAD_{ratio,i} = \frac{MADP_i}{\frac{1}{(i-1)} \sum_{j=1}^{i-1} MADP_j} \quad (3.7)$$

Then the frame complexity ( $FC_i$ ) measure for the  $i^{th}$  frame is computed by combining the MAD ratio and the  $N_{nzMVD}$  ratio as shown in Eq.3.8. The model parameter  $\beta$  is set empirically with a value of 0.7 for complex sequences and 0.3 for simple sequences.

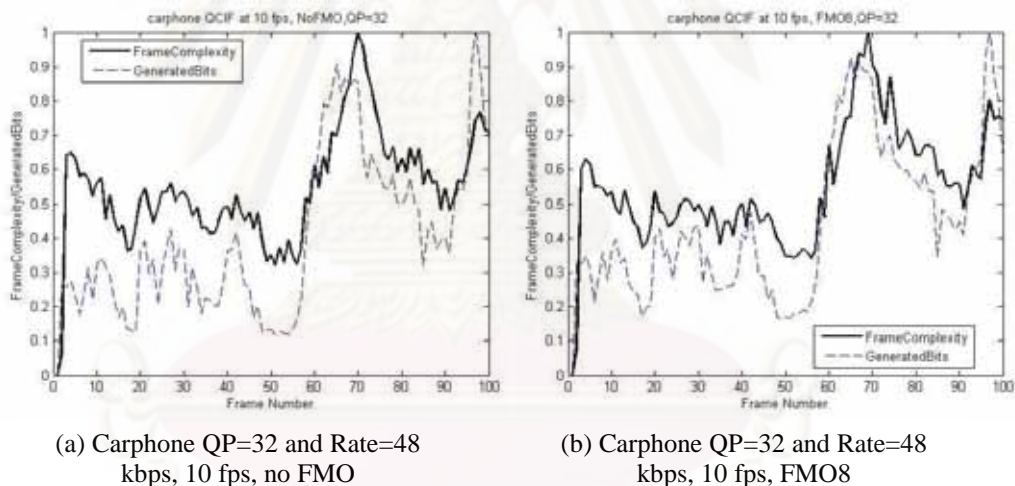
$$FC_i = \beta \cdot MAD_{ratio,i} + (1 - \beta) \cdot N_{nzMVDratio,i} \quad (3.8)$$

To determine the accuracy of the frame complexity model, we compare the actual generated bits and the computed frame complexity measure using Eq.3.8 for several test sequences. The carphone sequence (complex sequence) was encoded at fixed



QP=32 corresponding to a bit rate of approximately 48 kbps, so that the generated bits will be proportional to the frame complexity. The generated bits were compared to our modified rate control algorithm at a constant rate of 48 kbps and the frame complexity is measured using Eq.3.8. The comparison of the normalized generated bits and the frame complexity measure generated by a constant QP and constant bit rate encoder with no FMO and FMO with eight slice groups is shown in Figure 3.21(a)-(b).

As shown in Figure 3.21, the computed frame complexity from Eq. 3.8 correlates well with the actual number of generated bits. Similar trend is observed with other test sequences with different number of slice groups. Hence, the enhanced frame complexity measure using Eq. 3.8 is an accurate measurement of frame complexity and can be used to adjust the QP assignment to improve the frame layer rate control. For the complete plot of complexity measure with other sequences, the Figures are provided in Appendix B-1.



**Figure 3.21** (a)-(b) Carphone sequence encoded at QP=32 and Rate=48 kbps, no FMO and FMO with eight slice groups

### 3.3.3 Proposed Frame Layer Rate Control Enhancement

The purpose of rate control is to compute QP for all frames within the allowable rates. It can be assumed, without loss of generality, that the GOP structure is *IPPP...*, where *I* is an intra-coded picture and *P* is a forward-predicted picture. The adaptive rate control scheme in the H.264/AVC is composed of two layers: the GOP layer rate control and the frame layer rate control. For this study, the basic unit is selected as a frame so there is no need for an additional basic unit rate control.

The operations of the GOP layer rate control is described briefly as follows. At the beginning of the GOP, the GOP later rate control computes the total number of bits for the GOP and assigns an initial QP for the first  $I$ - and  $P$ -frame. For the succeeding  $P$ -frames, the number of remaining bits in the GOP is updated based on the generated bits of the previous frame. The details of the GOP layer rate control is given in (Li, 2003).

The frame layer adaptive rate control algorithm in H.264/AVC is composed of three parts: determine the target bits for each  $P$ -frame, compute the QP, and adjustment of QP. The operations of each component are discussed along with the proposed enhancements.

### 3.3.4 Computation of the Frame Layer Target Bits

To compute the target bits for each frame, the fluid flow traffic model is used based on linear tracking theory (Li, 2003). Target bits ( $T_{buf}$ ) for the  $i^{th}$  frame is computed based on the current buffer fullness ( $CBF$ ), target buffer level ( $TBL$ ), frame rate, and available channel bandwidth, as shown in Eq. 3.9.

$$T_{buf,i} = \left\lceil \frac{b_r}{f_r} - \Gamma(CBF_{i-1} - TBL_i) \right\rceil \quad (3.9)$$

In Eq.3.9,  $b_r$  and  $f_r$  denote the bit rate and frame rate respectively. The current buffer fullness and the target buffer level are denoted as  $CBF_{i-1}$  and  $TBL_i$  respectively. In the JM reference software  $\Gamma$  is a constant with typical values of 0.5 and initial value for  $CBF_{i-1}$  and  $TBL_i$  are computed at the GOP layer rate control.

Target bits ( $T_{rem}$ ) for the  $i^{th}$  frame is also computed based on the remaining bits in the GOP, as shown in Eq. 3.10.

$$T_{rem,i} = R_i/N_i \quad (3.10)$$

In Eq.3.10,  $R_i$  is the number of remaining bits in the GOP and  $N_i$  is the number of non-coded  $P$ -frames.

To obtain better estimate of the target bits, we adjust the computation of  $T_{rem}$  to consider the frame complexity  $FC_i$  (Section 3.3.2), we denote the modified target bits as  $T_{mod}$  and is given by Eq. 3.11.

$$T_{mod,i} = \begin{cases} FC_i \cdot T_{rem,i} & 0 < FC_i < 1.0 \\ 1.1 \cdot T_{rem,i} & 1.0 \leq FC_i < 1.2 \\ 1.2 \cdot T_{rem,i} & 1.2 \leq FC_i \end{cases} \quad (3.11)$$

The parameters in Eq. 3.11 are derived empirically from experiments. The idea is to allocate more bits for more complex frames and lower target bits for less complex frames.

The total number of bits allocated for the  $i^{\text{th}}$  frame ( $T_i$ ) is computed as a weighted combination of the target bits computed from the target buffer level and buffer occupancy ( $T_{buf,i}$ ), and target bit computed from the remaining bits in the GOP ( $T_{mod,i}$ ) as shown in Eq. 3.12. In Eq. 3.12, the typical value of  $\beta_r$  in the JM reference software is 0.5.

$$T_i = \beta_r \cdot T_{mod,i} + (1 - \beta_r) \cdot T_{buf,i} \quad (3.12)$$

### 3.3.5 Using the Proposed Header Bits Model

In the H.264/AVC rate control, after computation of the target bits, the number of bits allocated for texture is computed by subtracting from the total bits the estimate number of header bits. The estimate number of header bits is computed as the average number of the header bits of previously coded  $P$ -frames. It has been previously studied that the number of header bits varies greatly from frame to frame and a simple average is not a good estimate of the header bits (Kwon, 2007).

The proposed improvements to the frame layer rate control of H.264/AVC is the estimate of the header bits is modified to use the proposed header bits model as computed using Eq. 3.5 to consider the effect of FMO and slice header overhead. This allows more accurate estimate of the header bits and consequently makes the bit allocation for the texture bits more accurate as well. The number of bits allocated for texture ( $T_{txt,i}$ ) is computed as in Eq. 3.13.

$$T_{txt,i} = T_i - H_{Pframe,i} \quad (3.13)$$

After the estimate header bits are subtracted from the computed target bits, QP for the  $i^{\text{th}}$  frame is computed from the remaining texture bits using the quadratic rate-distortion model.

### 3.3.6 Using Frame Complexity and QP Adjustment

After computing QP using the quadratic rate-distortion model, QP is further adjusted to  $\pm 2$  of the previous QP to maintain smoothness of visual quality. This kind of adjustment is not sufficient in some cases, especially when FMO is used. We further adjust QP depending on whether the target bit is positive or negative and a lower bound on the texture bits is imposed.

When the computed target bits per frame are low (low bit rate and high complexity frame) there is high probability that target bits will fall below zero for the succeeding frames. In this case, the QP is adjusted to be larger than two from the previous frames resulting in poor video quality. The effect is severe when FMO is used with eight slice groups where the target bits are observed to be negative most of the time, especially in complex sequences. Thus, it is important to prevent negative target bits to maintain smooth visual quality. As an improvement, we use the computed frame complexity, the buffer status, and the number of slice groups to adjust QP to maintain positive target bits for improved performance.

Depending on the amount of header bits, the remaining bits for texture can be too small; in this case a lower bound is imposed on the texture bits given by Eq.3.14.

$$T_{texture} = \max \left\{ T_{texture}, \frac{b_r}{MINVAL \cdot f_r} \right\} \quad (3.14)$$

In the JM reference software MINVAL is a constant with typical value of 4. The QP value computed when using the lower bound usually does not meet the target bits for the current frame, the mismatch is higher when FMO is enabled with large number of slice groups. Thus, it is necessary to further adjust QP for such cases.

#### 3.3.6.1. Negative Target Bit

When the frame is complex and FMO is enabled the current buffer fullness tends to be significantly larger than the target buffer level. In such cases, the target bits tend to be negative, so the current buffer level must be reduced by increasing QP to maintain positive target bit levels. The amount of QP adjustment depends on the number of slice groups when FMO is used as shown in Eq.3.15.

$$QP = \begin{cases} QP + 2 & \text{num\_slice\_grp} < 4 \\ QP + 3 & \text{otherwise} \end{cases} \quad (3.15)$$

#### 3.3.6.2. Positive Target Bit

When the computed target bit is positive and the allocated bits for texture is greater than the minimum bound using Eq.3.10, then QP is computed using the quadratic rate-distortion model. To maintain smoothness of visual quality, QP is limited to be within  $\pm 2$  of the current value between pictures. As an improvement, QP is further adjusted depending on the current buffer fullness; frame complexity and the number of FMO slice groups as in Eq.3.16. The threshold values are set empirically based on the experiments.



$$QP = \begin{cases} QP - 1 & \left( \Gamma \cdot (CBF - TBL) < \frac{b_r}{f_r} \right) \text{ and } (FC < 0.4) \\ QP + 1 & \left( \Gamma \cdot (CBF - TBL) > \frac{b_r}{f_r} \right) \text{ and } (FC > 1.3) \\ & \text{and } (num\_slice\_grp < 4) \\ QP + 2 & \left( \Gamma \cdot (CBF - TBL) > \frac{b_r}{f_r} \right) \text{ and } (FC > 1.3) \\ & \text{and } (num\_slice\_grp > 4) \end{cases} \quad (3.16)$$

### 3.3.6.3. Lower Bound on Texture Bit

When the amount of bits allocated for texture is set to the minimum bound dictated by the bit rate and the frame rate as in Eq.3.17, QP is simply adjusted by adding 2. Otherwise QP is unchanged.

$$QP = \begin{cases} QP + 2 & T_{texture} < \frac{b_r}{MINVAL \cdot f_r} \\ QP & \text{otherwise} \end{cases} \quad (3.17)$$

### 3.3.7 Frame Skipping

After encoding the current frame, the number of generated bits is added to the buffer and the model parameters of the rate control are updated. If the current buffer level is above a certain threshold, then the encoder will skip encoding the incoming frame. The initial buffer size ( $B_s$ ) is set at  $3.0 \cdot (b_r/f_r)$  for low-bit rate and low delay application. The buffer occupancy threshold before skipping a frame is set at  $0.8 \cdot B_s$ . Frame skipping is implemented by performing a “virtual skip”, that is all the MBs in the frame are encoded in PSKIP mode. This mode generates the least number of bits for encoding a macroblock.

### 3.3.8 Results and Discussion

The PSNR and standard deviation are averaged at different rates using 20, 32, 48, 64 and 96 kbps, and also averaged for different number of FMO slice groups i.e. no FMO and FMO with 2, 4, and 8 slice groups. The results are summarized in Table 3.7; it shows that the proposed rate control enhancements can improve the PSNR especially for sequences with high motion activity such as carphone and foreman where the average gain in PSNR is 0.19 dB and 0.64 dB respectively. The average PSNR gain for the football sequence, another high motion activity sequence, is 0.02 dB. The small gain achieved for the football sequence can be attributed to the inherent complexity of the sequence, from Table 3.7 the average PSNR for football sequence is below 30 dB which suggest that the operated bit rates (20 kbps – 96 kbps) is not enough to encode the sequence with significant visual fidelity. This is illustrated by

observing the first four frames of football and foreman sequence encoded at 32 kbps is shown in **Figure 3.22**. The low visual quality due to low bit rate encoding is evident from the blurring of the images.

**Table 3.7.** Comparison of PSNR and PSNR standard deviations averaged over several bit rates and different numbers of FMO slice groups.

All rates	Avg. PSNR (dB)			Avg. STD PSNR	
	JM	Proposed	Gain	JM	Proposed
Akiyo	42.11	42.16	0.05	3.37	3.29
Claire	42.67	42.70	0.03	2.99	2.86
Carphone	33.49	33.69	0.19	3.65	3.21
Foreman	31.28	31.92	0.64	3.43	2.11
Football	26.68	26.70	0.02	0.83	0.77

The foot ball sequence is a high motion activity sequence similar to foreman, but with very different motion characteristics as illustrated by the motion field (the green lines in the figure) in Figure 3.22. For example, from the football sequence has a more localized motion field affecting a few macroblock in the frame due to the motion of the players with a fixed camera position. In contrast the motion field of the foreman sequence is due to both the movement of the camera which generates a global motion field affecting almost all macroblocks in the frame and also to the local motion of the of the head of foreman. The presence of both global motion and local motion fields in the foreman sequence affects the variation in video quality as compared to the football sequence as measured by the PSNR standard deviation. Although a comparative study of the complexity of foreman and football sequence is outside the scope of this study. From the point of view of the proposed rate control (and also the JVT rate control), the football sequence is easier to manage: smaller PSNR standard deviation, lesser number of frames skipped and overall PSNR is lower for the football sequence as compared to the other sequences tested. This can be attributed to a higher average QP values used in encoding the football sequence, resulting in lower visual quality.

In Table 3.7, the average PSNR standard deviation of the football sequence is also small in comparison with the other four sequences which suggest small variations in video quality (also small variations in QP values); the complex nature of the football sequence forces the encoder to use higher QP values which significantly reduce the PSNR.

Overall the average PSNR standard deviation is reduced for all sequences tested, this suggest that the proposed rate control scheme is able to attain a more stable buffer management. The complete comparison of PSNR, standard deviation, number of frames skipped and the total bit rate is shown in Appendix A-3. The results the comparison of rate control performance using the football sequence is shown in Table A-3.5.



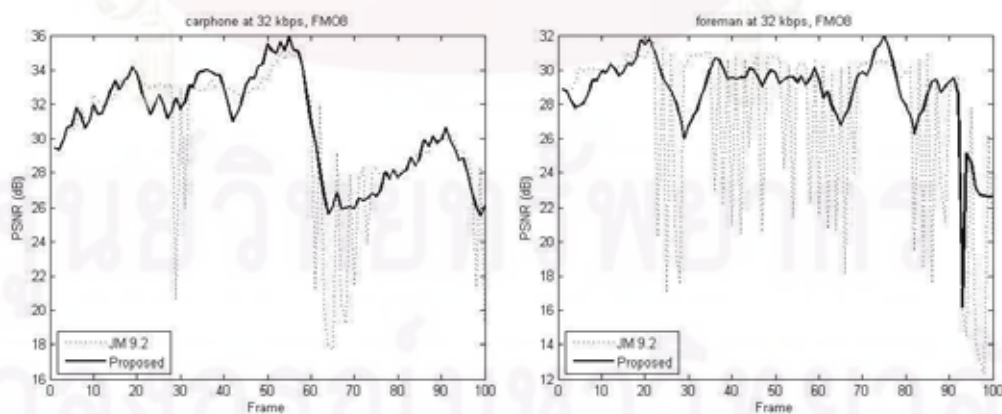
**Figure 3.22** First four frames of football and foreman sequence



The proposed rate control enhancements perform well at bit rates of 20 kbps and 32 kbps for carphone and foreman sequence. The performance of the JVT rate control and the proposed rate control method is almost the same in the football sequence. The low PSNR (below 30 dB) achieved using the football sequence suggests that the sequence is too complex to encode at the operated bit rates.

The average PSNR, average standard deviation, average generated bits and total number of skipped frames over all FMO slice group settings are shown in Table 3.8 and Table 3.9 for 20 kbps and 32 kbps respectively. Overall, experimental results suggests significant PSNR gains are achieved at lower bit rates when FMO is enabled, even for the case of sequences with high motion activity the improvement in PSNR at greater at lower bit rates. On the average the amount of bits utilized by the proposed rate control enhancements is lower compared to the JVT rate control, this indicates more efficient bit allocation. The PSNR standard deviation and number of skipped frames is also reduces indicating more stable buffer management. It is worthy to note that for the football sequence the number of frames skipped is comparable to the other low motion sequences like Akiyo and Claire. Again, the likely explanation is that encoding football at the operated rates results in much higher QP values which significantly affects the PSNR and PSNR standard deviation.

A comparison of the PSNR performance of the proposed rate control scheme compared with the JM 9.2 rate control for carphone and foreman at 32 kbps using FMO with eight slice groups is shown in Figure 3.23. The PSNR plots for the other sequences at 20 kbps and 32 kbps are provided in Appendix B-2.



**Figure 3.23** Comparison of PSNR at 32kbps using FMO with eight slice groups for Carphone and Foreman sequence



A comparison of the PSNR performance of the proposed rate control and JM 9.2 rate control at different FMO settings and at different rates for the foreman sequence is shown in

Table 3.10. Improvements are observed at almost all test cases. This is because the accuracy of the frame complexity model and header bits model depends on the motion vector difference when FMO is enabled. The PSNR gains for sequences with low motion content are comparable with the JM rate control but at a slightly lower bit rate. This means that the proposed scheme can allocate the bits more efficiently than the JM rate control. The number of frames skipped is also significantly reduced. The results at higher bit rates are not shown due to space constraints. But it can be generalized that at higher bit rates the gains in PSNR, standard deviation and number of skipped frames gradually decrease because the side effects of using FMO are less noticeable at higher bit rates. To illustrate this effect, the R-D curves of carphone and foreman sequence is shown in Figure 3.24 and Figure 3.25, the R-D curves of other sequences are shown in Appendix B-3 for reference.

**Table 3.8** Comparison of PSNR and PSNR standard deviation averaged over different number of FMO slice groups at 20 kbps.

20 kbps	Avg. PSNR (dB)		PSNR	Avg. PSNR std	
Video	JM	Proposed	Gain	JM	Proposed
Akiyo	36.76	37.02	0.25	2.47	2.12
Carphone	37.81	37.96	0.15	2.22	1.64
Claire	28.67	29.24	0.57	3.88	2.70
Foreman	25.80	26.97	1.17	4.60	2.35
Football	23.10	23.21	0.11	1.30	1.03
	Avg. Rate (Kbps)		Total Skip		
Video	JM	Proposed	JM	Proposed	
Akiyo	20.09	20.01	39	8	
Carphone	20.12	19.98	26	0	
Claire	20.30	20.07	86	6	
Foreman	20.33	20.19	143	18	
Football	20.19	20.01	59	6	

**Table 3.9** Comparison of PSNR and PSNR standard deviation averaged over different number of FMO slice groups at 32 kbps.

32 kbps	Avg. PSNR (dB)		PSNR	Avg. PSNR std	
Video	JM	Proposed	Gain	JM	Proposed
Akiyo	40.15	40.17	0.02	2.70	2.70
Carphone	40.99	40.96	-0.03	2.36	2.29
Claire	31.56	31.84	0.29	3.63	2.95
Foreman	28.91	30.21	1.30	4.46	1.94
Football	25.16	25.15	-0.01	0.83	0.82
	Avg. Rate (Kbps)		Rate	Total Skip	
Video	JM	Proposed		JM	Proposed
Akiyo	32.00	31.97		0	0
Carphone	32.06	31.98		2	0
Claire	32.23	32.09		23	1
Foreman	32.23	32.13		77	2
Football	32.08	31.96		0	0

**Table 3.10** PSNR for foreman sequence at different bit rates and different number of FMO slice groups.

Foreman (PSNR)	NoFMO		FMO2	
Rate (kbps)	JM	Proposed	JM	Proposed
20	27.323	29.041	26.034	27.365
32	29.028	31.465	28.522	30.621
48	33.043	33.203	32.315	32.753
64	34.306	34.623	33.743	34.121
96	36.473	36.473	36.029	36.101
	FMO4		FMO8	
Rate	JM	Proposed	JM	Proposed
20	24.063	26.143	22.548	23.961
32	27.462	29.957	26.454	28.608
48	31.882	32.142	29.315	31.626
64	33.739	33.761	32.767	33.338
96	35.612	35.802	35.443	35.455

### 3.3.9 Trade-off of the Proposed Rate Control Enhancements

The proposed rate control enhancements are able to achieve better bit allocation resulting in improved PSNR, the trade-off in doing so is the additional computation of modeling the header bits and the delay incurred in gathering the motion vector data after motion estimation. In this section we will discuss in detail the trade-off of the proposed rate control enhancements in terms of added computation and encoding delay.

The proposed enhancements to the JVT rate control as discussed in Section 3.3.3 consists of target bits computations and QP adjustment depending on the frame complexity, number of FMO slice groups are other rate control related parameters. The rate control enhancement themselves are not computationally expensive, but rather they depend on parameters that are derived from empirical experiments and modeling data. The most computationally expensive routine in the proposed method is the modeling of the headers bits using Eq. 3.5 which requires solving a linear regression problem in two variables. As opposed to the simple averaging method of previous header bits used in the JVT rate control.

In the JVT adaptive rate control, an estimate of the header bits is derived by averaging the header bits used in the previous frames:  $H_{pframe,k} = \frac{1}{N-1} \sum_{i=1}^{N-1} R_{hdr,i}$ . This operation requires  $(N - 1)$  additions and one division operation, where  $N$  is the number of frames. In contrast, using Eq. 3.5, as shown below requires solving the model parameters  $\alpha_1$  and  $\alpha_2$  using linear regression.

$$H_{pframe} = \alpha_1 N_{nzMVD} + \alpha_2 (N_{MV} + num\_slice)$$

The process of solving the model parameters  $\alpha_1$  and  $\alpha_2$  in Eq. 3.5 is a two step process. The first step is to gather the experimental data after the motion estimation process and motion vector encoding for each macroblock in the frame to get the number of header bits, number of non-zero motion vector difference and number of motion vectors for each macroblock. The amount of header bits contributed by residual coding of the macroblock is negligible since it contains only the macroblock mode, hence the macroblock header bits is mostly due to the motion related side information. The process of data gathering incurs encoding delay penalty, since motion estimation have to be done twice, the first pass the gather the data and the second pass during the frame encoding.

As an example, a table of computed motion estimation times and total encoding time for the sequences under test with 100 frames using different number of slice groups is shown in Table 3.11 at 20 kbps bit rate.

**Table 3.11** Computed motion estimation times (ME) and encoding times at 20 kbps

20 kbps	Motion Estimation Time (sec)				
Video	NoFMO	FMO2	FMO4	FMO8	Avg.
Akiyo	8.02	8.19	7.88	8.03	8.03
Claire	8.35	8.26	8.16	8.21	8.25
Carphone	8.23	8.22	8.29	8.39	8.28
Foreman	8.24	8.30	8.20	8.34	8.27
Football	8.86	8.76	8.91	8.89	8.85
	Encoding Time (sec)				
Video	NoFMO	FMO2	FMO4	FMO8	Avg.
Akiyo	38.54	32.21	29.57	29.10	32.36
Claire	38.59	33.11	29.77	29.11	32.64
Carphone	37.81	32.01	29.94	29.78	32.39
Foreman	39.49	31.46	30.13	29.97	32.76
Football	37.58	31.37	30.78	29.83	32.39

The motion estimation times are roughly 25% of the total encoding time, regardless of the video sequence and FMO slice groups. This means that on the average the incurred delay penalty due to the motion estimation process needed to gather modeling data for the proposed rate control enhancements is approximately 25% additional encoding time. This is just a rough estimate since motion estimation times depends other factors such as number of reference frames, size of search space and other encoder related parameters.

After the experimental data needed for modeling has been gathered from motion estimation of each macroblock in the frame, the second step is to solve for the values of the model parameters  $\alpha_1$  and  $\alpha_2$ . The model parameters can be solved by minimizing the error between the model estimates and the experimental data using linear regression<sup>1</sup>, which requires solving a set of linear equations of the form.

$$\begin{bmatrix} \sum_{i=1}^N (H_{pframe,i} \cdot N_{nvMVD,i}) \\ \sum_{i=1}^N (H_{pframe,i} \cdot (N_{MV,i} + slice)) \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^N (N_{nvMVD,i})^2 & \sum_{i=1}^N (N_{nvMVD,i} \cdot (N_{MV,i} + slice)) \\ \sum_{i=1}^N (N_{nvMVD,i} \cdot (N_{MV,i} + slice)) & \sum_{i=1}^N (N_{MV,i} + slice)^2 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}$$

We estimate the computational complexity by assuming that a multiplication operation is the same complexity as a division operation, and that an addition

<sup>1</sup> [http://en.wikipedia.org/wiki/Regression\\_analysis](http://en.wikipedia.org/wiki/Regression_analysis)



operation is the same complexity as a subtraction operation. There are six elements in the matrix of linear equations; each element can be computed at most by using  $N$  multiplications and  $N$  addition operations. To solve for the model parameters using Cramer's rule requires solving the determinant of two matrices followed by a division operations. We can assume that the cost of computing the matrix determinants are negligible compared to computing the elements of matrix.

Hence, the simple averaging method used by JVT requires  $(N-1)$  and one division operation. Whereas the proposed rate control enhancement requires  $6N$  multiplication operations and  $6N$  addition operations. Roughly an increase in more than twelve times in the computation of the header bits using a linear regression model as compared to averaging. This does not immediately conclude that the incurred delay will also be twelve times, other factors should be taken into consideration.

It should be noted that we only estimate the number of elementary operations involved, the process of computing or executing these elementary operations must also be considered. As an example, if we consider a Pentium IV<sup>2</sup> processor operating at 2.4 GHz as the computing engine, it can easily perform 7,295 Millions of Instruction per second (MIPS). Considering that this processors have several parallel computation capabilities, a separate floating point unit (FPU), special multimedia hardware support such as a multiply and accumulate unit (MMX), cache memory performance and SIMD (Single Instruction Multiple Data) architecture. It can be concluded that the overhead of more than  $6N$  multiplication and  $6N$  addition operations incurred by the modeling effort is negligible. The more important impact of the proposed rate control enhancement is the incurred delay of the additional motion estimation operation.

### 3.3.10 Summary

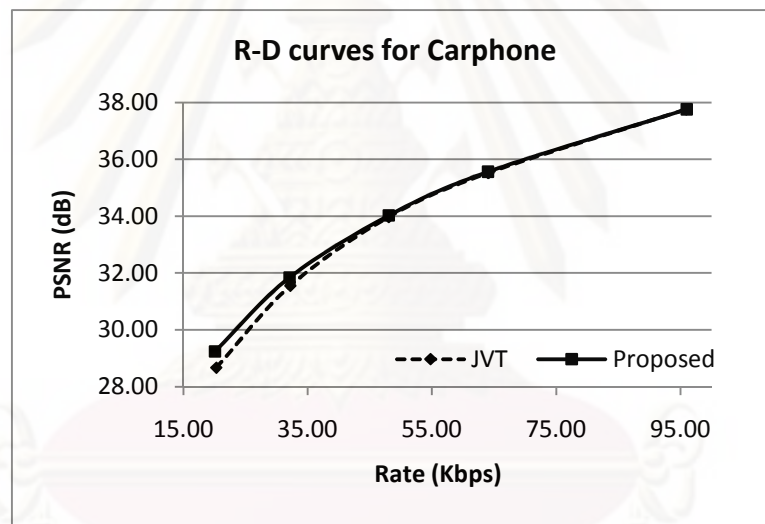
We have presented some improvements to the H.264/AVC frame layer rate control with consideration of using FMO for added error resiliency. We propose a new header bits model that uses the number of motion vector difference to more accurately model the header bits. A new frame complexity measure is proposed also using the number of motion vector difference to enhance the existing MAD-based frame complexity measure. We propose some target bits modification and QP adjustment schemes considering buffer fullness, frame complexity and number of

---

<sup>2</sup> [http://en.wikipedia.org/wiki/Pentium\\_4](http://en.wikipedia.org/wiki/Pentium_4)

FMO slice groups to generate QP that better allocates the bits for encoding the current frame.

It has been shown that the implemented FMO-based frame layer enhancements improves the PSNR significantly and can achieve the target bit rates more accurately as compared to the current H.264/AVC rate control for sequences with high motion and at low bit rates. The performance for sequences with low motion content is comparable to the JM rate control but achieves slightly lower bit rates. A smoother video quality is achieved because of a more stable buffer management. The number of skipped frames is also significantly reduced at low bit-rates and for high motion sequences, thus improving the overall PSNR. The proposed scheme is very useful for low-bit rate video streaming for wired and wireless networks. The computational overhead is negligible, but an incurred additional delay of roughly 25% in the total encoding time due to the motion estimation operation to gather the modeling data.



**Figure 3.24** Comparison of R-D curve with JVT reference rate control for Carphone

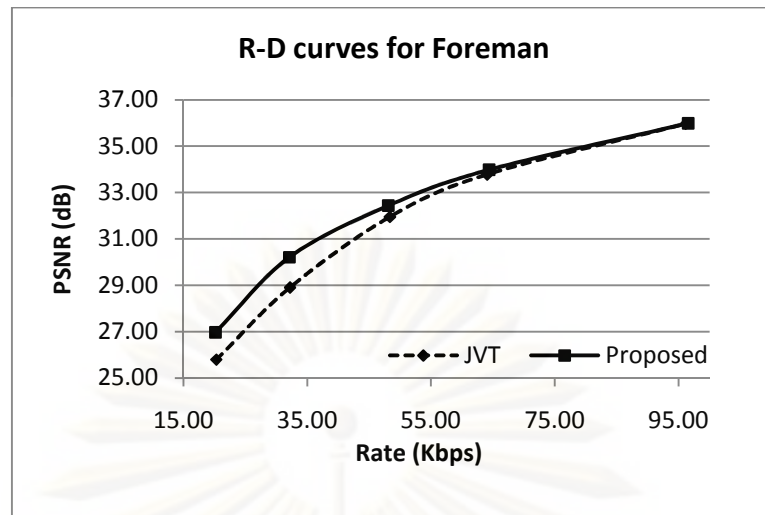


Figure 3.25 Comparison of R-D curve with JVT reference rate control for Foreman

ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย

### 3.4 Feedback based FMO selection strategy

In order to minimize the overheads incurred by using FMO under fading channel conditions with low-delay and low bandwidth constraints it is prudent to apply FMO only to certain frames that will most likely suffer from burst errors. Based on feedback information from the channel, in the form of ACK/NACK packets, the number of FMO slice groups that will be used in encoding the current frame and succeeding frame is adjusted to anticipate the effects of burst errors.

In order to incorporate the FMO selection strategy based on feedback information, certain assumptions must be made about the video transmission system. First, it is assumed that a dedicated error free feedback channel is available from the decoder to the encoder to report the number of packets lost or received during video transmission. This can be accomplished by utilizing a number of network transmission protocols, for example using ARQ protocols but without the option to retransmit the packets but only the protocols to report the lost packets. Another option is to use RTCP protocols for the feedback system for monitoring the transmission statistics to determine the conditions of the wireless channel. Second, it is assumed that the intended application is for low-bit rate and low-delay video transmission systems, where the round trip delay time is only a few milliseconds. This is necessary to allow for some time delay incurred in predicting the channel and make the necessary adjustments on the encoder without exceeding the required end-to-end transmission delay. Many low-tier radio transmission systems such as the Personal Access Communication Systems (Noerpel, 1996) (PACS) used for personal communications services (PCS) satisfy the low-bandwidth, low-delay application requirement.

Thus in the context of a low-delay, low-bandwidth video transmission system for personal communications services, where the channel statistics are monitored in the form of ACK/NACK packets we develop a feedback based FMO-selection strategy to improve performance of using FMO for wireless video transmission. The feedback information is used to monitor the packet error rate and instead of retransmitting the packets with error, we re-encode the frame with FMO enabled using different number of slice groups and retransmit the whole frame.



### 3.4.1 Simulated Video Transmission System

In this work we investigate the transmission of video sequences in a typical PCS environment such as the PACS system which is a low-tier radio system with throughput of 32 kbps. The payload of a PACS frame is 80 bits per packet and the roundtrip delay is in the order of a few tens of milliseconds suitable to simulate a low-delay, low bandwidth video transmission system. In this work we assume a fixed round-trip delay of 30 milliseconds. We assume error free feedback information is available from the decoder to encoder to report the number of packets lost in the form of NACK/ACK packets.

A wireless channel simulator is used to simulate Rayleigh fading channels using the techniques as described in Section 2.6. To simulate a slow fading wireless channel conditions the maximum Doppler frequency is set to 1 Hz. The simulator outputs an error pattern corresponding to the simulated wireless channel conditions.

We use the JM reference software version 9.2 as the video codec. A non-motion compensated error concealment process is simulated at the decoder by discarding macroblocks that have an error until the end of the slice regardless of the macroblock coding mode. Several video sequences are encoded using the baseline profile at level 3.0. The GOP structure is *IPPP...* with one reference frame, each sequence is encoded for a total of 100 frames at a frame rate of 10 frames per second. The encoder is modified to encode the macroblock-to-slice group map in the PPS header and inserted into the bit stream each time a frame is encoded with explicit FMO enabled. Rate control is enabled and the bit rate is set at 20kbps and 32 kbps for low bit rate transmission.

### 3.4.2 Proposed FMO-selection strategy

The objective is to estimate the severity of channel fading based on the number of packets lost during transmission. If many packets are lost during the transmission of the first few packets of the frame, then instead of retransmitting the lost packets, the frame is re-encoded with FMO enabled and retransmitted to provide some error resiliency. This is done to help the error concealment scheme at the decoder to minimize the degradation in visual quality due to transmission errors.

Initially, all frames are encoded with no FMO then based on the feedback information, for each frame we compute the ratio ( $E_{ratio}$ ) of the number of packets transmitted with error to the total number of packets transmitted. We then compute the average of this ratio ( $E_{avg}$ ) in the past  $N$  frames including the current frame. If the average ratio is greater than a threshold, we decide that the channel condition is bad; otherwise the channel condition is good state. If the channel condition is bad because of the high number of packets with error, the current frame is re-encoded with FMO

enabled with different number of slice groups ( $NumSlice$ ) depending on the threshold as given in Eq 3.18. These threshold values are determined by experiment and correspond to  $N=2$ , that is the average packet error ratio of the current and the previous frame. In anticipation of possible burst errors the next frame is also marked to be encoded with FMO corresponding to the number of slice group used in the current frame.

$$E_{avg} = \sum_{i=1}^N E_{ratio,i}$$

$$NumSlice_{i,i+1} = \begin{cases} 8 & 0.05 < E_{avg} < 0.1 \\ 4 & 0 < E_{avg} \leq 0.05 \end{cases} \quad (3.18)$$

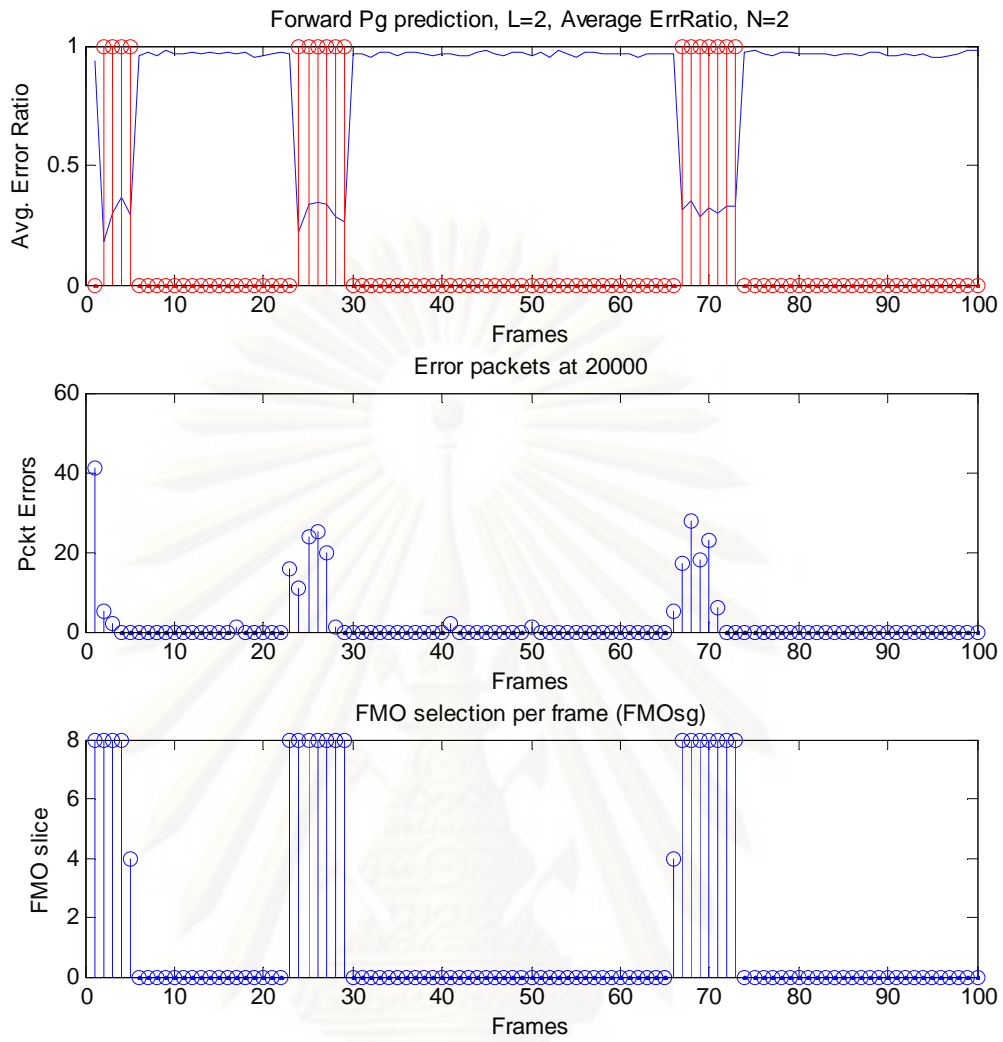
### 3.4.3 Experimental Results

We encode and transmit several video sequences using video transmission system and video codec settings as discussed in Section 3. We measure the PSNR of our proposed adaptive FMO (adFMO) selection scheme compared with videos encoded with no FMO and 2 (FMO2), 4 (FMO4) and 8 (FMO8) number of slice groups respectively. The summary of PSNR is shown in Table 3.12.

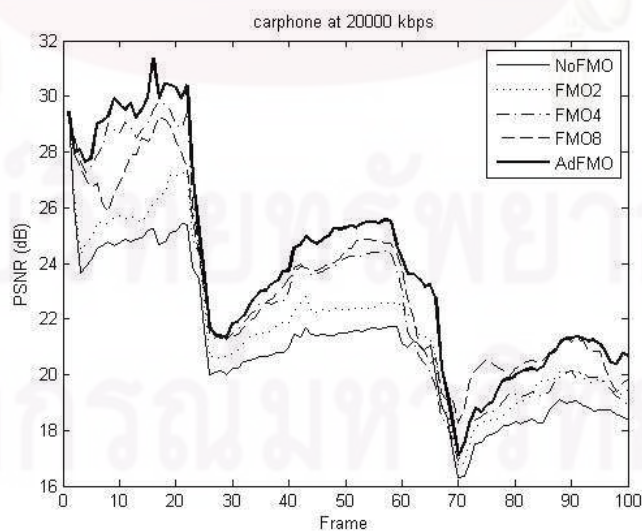
**Table 3.12** Average PSNR (dB) at 20 Kbps

	NoFMO	FMO2	FMO4	FMO8	AdFMO
Akiyo	30.32	30.22	30.43	30.31	30.95
Claire	26.83	27.93	28.41	29.11	29.67
Carphone	21.03	21.89	22.91	23.16	23.84
Foreman	15.42	17.14	17.80	18.74	18.94

The average PSNR gain of our proposed scheme at 20 kbps is 0.5 dB compared to using fixed number of slice groups. The gain in PSNR is due to the reallocation of bits used for source coding during periods of error free transmission that are otherwise allocated for FMO overheads if FMO selection is not used. A plot of the channel condition prediction, the packet errors per frame and the level of FMO protection assigned to each frame the carphone sequence is show in Figure 3.26.

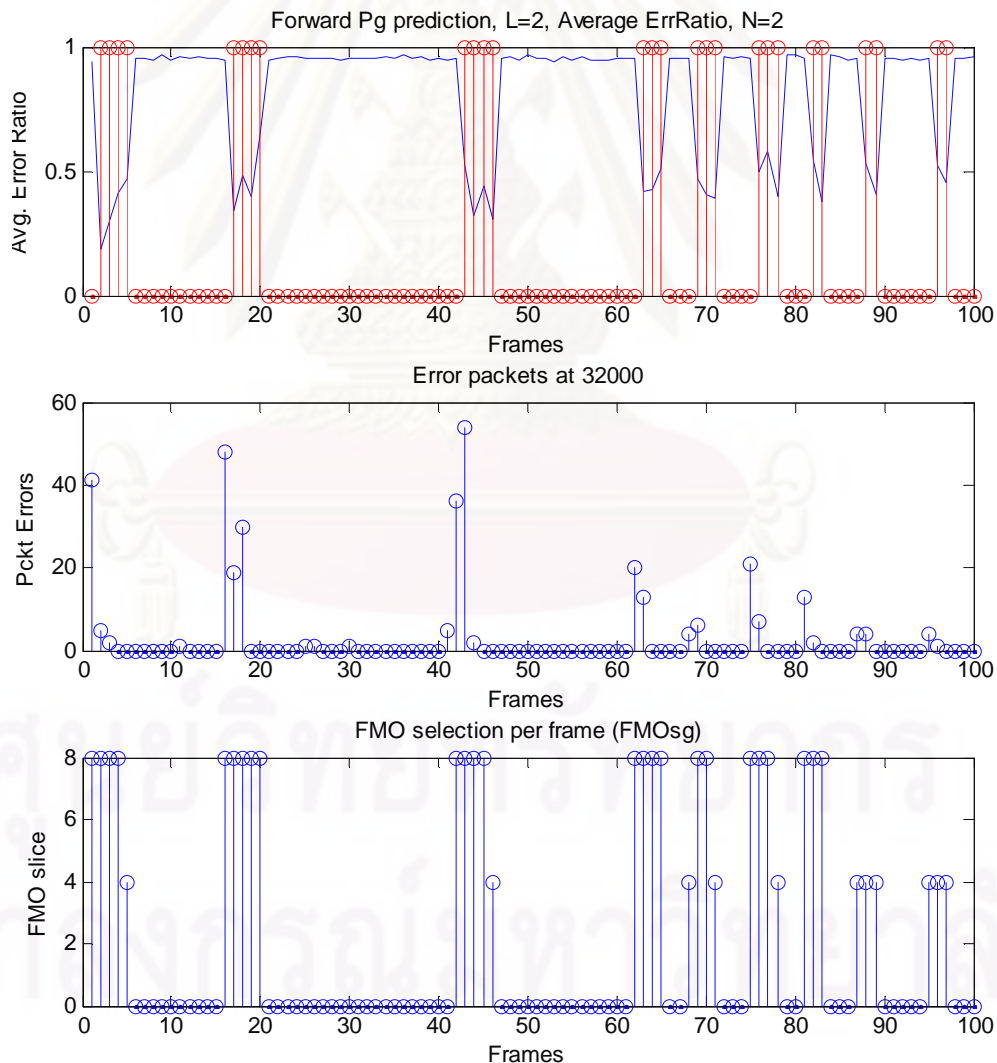


**Figure 3.26** Plots of Error ratio, packet errors and FMO mode selection for carphone sequence at 20 kbps.



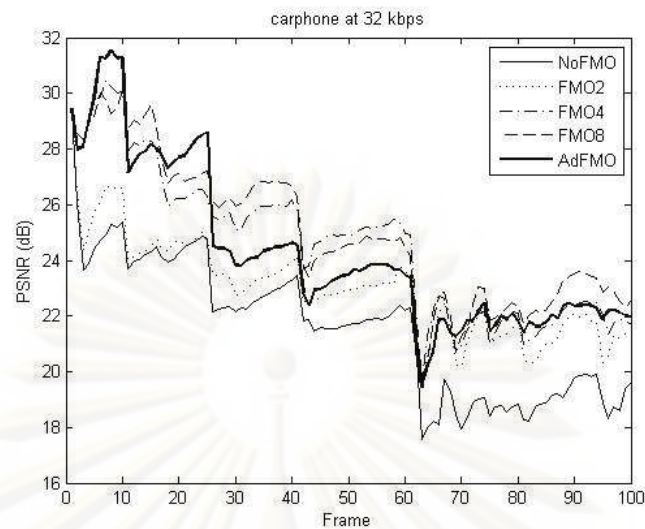
**Figure 3.27** Comparison of PSNR of Caprhone sequence at 20 kbps using different number of FMO slice groups and the adaptive selection method.

The use of FMO only during periods of burst errors and to minimize its use during periods of error free transmission as shown in Figure 3.26 can improve the overall PSNR. A comparison of the PSNR performance for the Carphone sequence at 20 kbps using different number of FMO slice groups compared the adaptive FMO-selection method is shown in Figure 3.27. By restricting the use of FMO only to frames that are experiencing burst errors, significant improvements in video quality can be obtained. But the trade-off is that the conditions of the channel must be accurately predicted from the feedback information. When the channel conditions is not accurately prediction, the frame will be encoded without FMO then the visual quality of the video will degrade because the frame have no protection against transmission errors, moreover the effect of propagation error is severely felt. This is illustrated Figure 3.28, for the carphone sequence at 32 kbps and the comparison of PSNR plot is shown in Figure 3.29.



**Figure 3.28** Plots of Error ratio, packet errors and FMO mode selection for carphone sequence at 32 kbps.





**Figure 3.29** Comparison of PSNR of Carphone sequence at 32 kbps using different number of FMO slice groups and the adaptive selection method.

The degradation in PSNR is due to a small packet error in frames 25, 26 and 30 which were not accurately predicted by the proposed method, because the error ratio is below the threshold. This small amount of packet errors can result in large visual degradation in the current frame due to the large number of undecodable macroblocks and the degradation also propagations to the succeeding frames due to motion prediction.

In such cases, the reconstruction quality of the decoded video largely depends on the performance of the non-motion compensated error concealment method.

#### 3.4.4 Summary

We have presented a feedback based FMO selection strategy where the number of FMO slice groups that will be used to encode the frame depends on the average ratio of packet errors. The proposed scheme works well for slow fading low bandwidth channel conditions where the channel error are characterized as having more burst errors and less isolated random errors. The improvement in PSNR is mainly due to the reallocation of source coding bits that are otherwise used for error resiliency by using FMO. Also, the proposed method depends on the accuracy of the channel prediction. In cases where there are there are isolated errors that are cannot be accurately predicted, the performance of the proposed scheme will depend more on the error concealment method used at the decoder.

## CHAPTER IV

### CONCLUSION AND FUTURE WORKS

#### 4.1 Summary and Conclusion

In this work, we present creative and practical solutions to improve the quality and error resiliency of transmitting compressed video over wireless channel environments.

In Section 3.1, we introduce the use FMO as an error resilient tool in H.264/AVC. Although a lot of research has been done about FMO, the issue of how to suitably arrange the macroblocks for effective transmission over wireless channels still requires significant research. There is no general framework available to analyze the effectiveness of a particular explicit FMO map under different application and transmission constraints. And because of an almost infinite number of possible macroblock-to-slice group mapping permutations, using practical macroblock importance parameters can help reduce the number of possible mappings to those that can theoretically improve the decoded video quality. The proposed *distortion-from-propagation* measure, using both spatial and temporal information, of macroblock importance gives a more accurate assessment of the importance of a macroblock in modeling the effect of error propagation and is used as a parameter for classifying macroblocks to slice groups. A novel explicit FMO remapping scheme is also proposed to generate a new slice group map that has the potential to perform better than the initial FMO map. The proposed STI-FMO scheme is shown to be more effective in improving the video quality as compared to other explicit FMO mappings based on spatial and temporal information alone as well as the fixed FMO mappings available in the H.264/AVC standard.

In Section 3.2, we focus on the development of a new macroblock classification process for explicit FMO. The problem of how to assign the macroblocks to slice groups to satisfy a certain criteria, for example achieve a minimum total distortion per slice, is also a very difficult problem to solve from a computational point of view. Exhaustive brute force minimization algorithms may not be practical considering the complexity involved and the potential gains that are achievable. Practical and yet effective macroblock classification schemes that are not too computationally complex are desirable. The proposed macroblock classification scheme is an improvement of

the simple interleave sorting algorithm but also computationally simple and yet effective in achieving modest gains. The proposed macroblock classification scheme uses a divide-and-conquer approach to minimize the variance of the macroblock importance parameter among different slice groups. This has been found to be effective in achieving modest gains in PSNR and reducing the number of undecodable macroblocks without necessarily changing the macroblock importance parameter, and thus extendable to other macroblock parameters as well.

In Section 3.3, the trade-offs in using FMO in terms of reduced coding efficiency and overhead bits are addressed. An analysis of the operations of the FMO seeks to identify the primary sources of the reduction in coding efficiency as well as the overhead in bits. The operations of the adaptive rate control of the H.264/AVC are also analyzed to find ways to improve the bit allocation scheme and at the same time taking into considerations the effect of FMO on the rate control. Based on the analysis, an improvement to the H.264/AVC frame layer rate control are proposed and consists of a new header bits model, an enhanced frame complexity measure, an improved target allocation and QP adjustment scheme. It has been shown that the proposed FMO-based frame layer rate control enhancements improves the PSNR significantly and achieves the target bits more accurately as compared to the current H.264/AVC rate control specially for sequences with high motion content and at low bit rates.

In Section 3.4, we highlight the importance of making judicious use of FMO under slow fading wireless channel environments. The idea is to maximize the use of the bit budget by not using FMO during periods of error free transmission, and enable the use of FMO only when the transmission channel is experiencing burst errors. A feedback based FMO mode selection scheme is proposed that makes use of a dedicated low-bandwidth feedback channel to monitor the channel packet error statistics. Based on the packet error statistics the channel conditions is predicted to be good or bad, and a decision is made to encode the frame with FMO enabled or disabled using different number of slice groups. The proposed scheme has been found to be effective for slow fading channels characterized by long but infrequent burst errors and small isolated random errors. In cases where the channel prediction fails to accurately determine the condition of the channel the performance of the proposed scheme will depend more on the error concealment method implemented at the decoder.

## 4.2 Future Works

The proposed methods of developing a more error-resilient video transmission system for wireless environments presents some of the possible practical solutions that are achievable within the given transmission and application constraints. Some other aspects of the work can be further explored as future studies.

The analysis and usefulness of explicit FMO map for wireless video transmission have not been thoroughly investigated yet. We presented some possible alternative ways to define a macroblock importance parameter, using bit count and distortion, and an improved sorting method as macroblock classification scheme. There are many other possible macroblock importance parameters that can be used considering for example, effects of error propagation, error concealment, macroblock coding modes, motion vector information, motion content, texture complexity, edge information, etc. Some other macroblock importance parameters can be used that takes into consideration channel/network related parameters such as packetization scheme and the use of FEC or ARQ protocols. Although we have presented a simple framework that uses feedback information that decides the number of FMO slice groups per frame it is also possible to use feedback information to dynamically change the MBAmapping each frame depending on the channel, i.e. slow or fast fading, conditions. Also the method macroblock classification is still largely unexplored and many classification schemes are heuristics and computationally complex, ultimately the classification scheme must depend also on network and application constraints and not only on the chosen macroblock importance parameter.

The proposed rate control scheme is able to properly exploit the weakness of the adaptive rate control when FMO is enabled by making use of the motion vector difference information that is found to be correlated with FMO. Another possible improvement is to consider the use of the MBAmapping information itself either in rate control, RDO or some other encoding decisions, because the map information directly affects the encoding and error resilient properties of the video encoder.

In this work we assume some rudimentary transmission constraints, fixed packetization, fixed delay and simple point-to-point transmission. Having a complete knowledge of the underlying transport/network constraints and analyzing its effects on FMO and rate control will be a step in the right direction in achieving a better joint source channel error resilient video transmission over wireless channels.



## References

- A.K.Kannue, & B.Li. (Oct. 2008). An Enhanced Rate Control Scheme with Motion Assisted Slice Grouping for Low Bitrate Coding in H.264. San Diego, California: ICIP 2008.
- A.Ortega, K.Ramchandran, & M.Vetterli. (1994). Optimal trellis-based buffered compression and fast approximation. *IEEE Trans. on Image Processing* , 3, 26-40.
- D.-K.Kwon, Shen, M.-Y., & Kuo, C.-C. J. (May 2007). Rate Control for H.264 with Enhanced Rate and Distortion Models. *IEEE Trans. Circuits and Systems for Video Technology*. 17 (5).
- Dhondt, Y., P.Lambert, & Walle, R. V. (2006). A Flexible Macroblock Scheme for Unequal Error Protection. ICIP 2006.
- E.N.Gilbert. (1960). Capacity of Burst-noise Channel. *Bell Systems Technical Journal* , 39, 1253-1266.
- Girod, B., & Farber, N. (2000). Chapter 12: Wireless Video. In M.-T.Sun, & A. Reibman, *Compressed Video Over Networks*. Mercer Dekker.
- Im, S., & Pearmain, A. (2005). Unequal error protection with the H.264 flexible macroblock ordering. *Proc. of SPIE Visual Comms. and Image Proc.*
- J.Lie, & K.N.Ngan. (2005). An Error Sensitivity-based Redundant Macroblock Strategy for Robust Wireless Video Transmission. Int'l Conf. on Wireless Networks, Comms. and Mobile Computing.
- Jiang, M., & Ling, N. (Feb. 2005). On Enhancing H.264/AVC Video Rate Control by PSNR-based Frame Complexity Estimation. 15 (1).
- K.N.Ngan, C.W.Yap, & K.T.Tan. (2001). *Video Coding for Wireless Communications*. New York: Marcel and Dekker, Inc.
- L.T.Ha, Kim, H.-S., Park, C.-S., Jung, S.-W., & Ko, S.-J. (Jan. 2009). Bitrate Reduction Using FMO for Video Streaming over Packet Networks. *PWASET* , 37.
- Lee, C., Lee, S., Oh, Y., & Kim, J. (2006). Cost-effective Frame-Layer H.264 Rate Control for Low Bit Rate Video. ICME 2006.
- M.Jiang, X.Yi, & N.Ling. (May 2004). Improved Frame Layer Rate Control Using MAD Ratio. Vancouver, Canada: IEEE Symp. on Ckts. and Syst.

- M.Zorzi, & R.R.Rao. (1996). Error Control Strategies for the Wireless Channel. 5th Int'l Conf. on Universal Personal Comms.
- M.Zorzi, R.R.Rao, & L.B.Milstein. (1997). ARQ Error Control for Fading Mobile Radio Channel. *IEEE Trans. on Veh. Tech.* , 46, 445-455.
- Michael, A., K.Suhring, & G.J.Sullivan. (Oct. 2004). *Proposed H.264/MPEG-4 AVC Reference Software Manual*. JVT-M012.doc, Joint Video Team (JVT), Palma,Spain.
- N.Thomos, S.Argyropoulos, N.V.Boulgouis, & M.G.Strintzis. (Nov. 2005). Error Resilient Transmission of H.264/AVC streams using Flexible Macroblock Ordering. EWIMT05. London.
- Noerpel, A., & Sherry, H. (May 1996). Personal Access Communication System A Flexible PCS Standard. NJ,USA: ELECTRO'96.
- P.Lambert, Neve, W., Dhondt, Y., & Walle, R. D. (2006). Flexible Macroblock Ordering in H.264/AVC. *Journal of Visual Reprs. and Image Comms.* , 17.
- Panyavaraporn, J., Cajote, R., & Aramvith, S. (Nov. 2007). Performance Analysis of Flexible Macroblock Ordering using Bit-Count and Distortion Measure for H.264 Wireless Video Transmission. SISB 2007. Bangkok, Thailand.
- R.A.Howard. (1971). *Dynamic Probabilistic Models*. John Wiley and Sons.
- R.D.Cajote, S.Aramvith, R.C.L.Guevara, & Y.Miyanaga. (may 2007). FMO Slice Group Maps using Spatial and Temporal Indicators for H.264 Wireless Video Transmission. ISCAS 2008. Seattle, Washington.
- R.D.Cajote, S.Aramvith, R.C.L.Guevara, & Y.Miyanaga. (2008). Improved Sorting Algorithm for Explicit FMO Macroblock Classification in H.264. ISCT 2008. Vientiane, Laos.
- S.Aramvith, & W.Hantanong. (De.c 2006). Joint Flexible Macroblock Ordering and FEC for H.264 Wireless Video Transmission. ISPACS.
- S.Aramvith, C.-W.Lin, S.Roy, & M.-T.Sun. (2002). Wireless Video Transport using Conditional Retransmission and Low-Delay Interleaving. *IEEE Trans. on Ckts. and SYs. for Video Tech.* , 12, 558-565.
- T.-C.Chen, L.-F.Chang, A.H.Wong, M.-T.Sun, & T.R.Hsing. (1995, April). Real-time software-based end-to-ned wireless visual communcations simulation platform. *Proc. of SPIE: Visual Comms and Image Proc.*
- T.Chiang, & Y.-Q.Zhang. (1997). A new rate control scheme using quadratic rate distortion model. *IEEE Trans. on Ckts. and Sys. for Video Tech* , 7, 246-250.

- T.Weigand, H.Schwarz, A.Joch, F.Kossentini, & G.J.Sullivan. (2003). Rate Constrained Coder Control and Comparison of Video Coding Standards. *IEEE Trans. on Ckts. and Sys. for Video Tech.* , 13, 688-703.
- W.Hantanong, & S.Aramvith. (Aug. 2005). Analysis of Macroblock-to-slice group mapping for H.264 Video Transmission over Packet-based Wirelss Fading Channel. 48th Midwest Symp. on Ckts. and Syst.
- Weigand, T., Sullivan, G., Bjontegaard, G., & Luthra, A. (2003). Overview of H.264/AVC video coding standard. *IEEE Trans. on Ckts. and Sys. for Video Tech.* , 13 (7), 560-576.
- Wenger, S., & Horowitz, M. (May 2002). FMO: *Flexible Macroblock Ordering*. ISO/IEC MPEG and ITU-T VCEG.
- Wu, Z., & Boyce, J. M. (Oct. 2006). Optimal Frame Selection for H.264/AVC FMO Coding. ICIP.
- Z.Li, F.Pan, K.P.Lim, G.Feng, X.Lin, & S.Rahardja. (March 2003). *Adaptive Basic Unit Rate Control for JVT*. Pattaya, Thailand: JVT 7th Meeting.



**APPENDICES**

ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย



## APPENDIX A

### A-1 Motion Vector Difference Data from Encoded Sequences

**Table A-1.1** Motion Vector Difference Data from four video sequences encoded at different QP and different number of FMO slice groups.

<b>Akiyo</b>					<b>%MVD</b>			
<b>QP</b>	<b>NoFMO</b>	<b>FMO2</b>	<b>FMO4</b>	<b>FMO8</b>	<b>NoFMO</b>	<b>FMO2</b>	<b>FMO4</b>	<b>FMO8</b>
8	14797	15273	15300	15746	0.00	3.22	3.40	6.41
16	11602	12039	12163	12307	0.00	3.77	4.84	6.08
24	8441	9012	8954	9094	0.00	6.76	6.08	7.74
32	4310	4456	4524	4609	0.00	3.39	4.97	6.94
40	1489	1649	1653	1745	0.00	10.75	11.01	17.19
48	134	162	147	140	0.00	20.90	9.70	4.48
<b>Carphone</b>								
<b>QP</b>	<b>NoFMO</b>	<b>FMO2</b>	<b>FMO4</b>	<b>FMO8</b>	<b>NoFMO</b>	<b>FMO2</b>	<b>FMO4</b>	<b>FMO8</b>
8	60552	65613	67828	70269	0.00	8.36	12.02	16.05
16	46192	48123	48969	49604	0.00	4.18	6.01	7.39
24	30549	32648	33266	33761	0.00	6.87	8.89	10.51
32	16134	17480	18161	18739	0.00	8.34	12.56	16.15
40	6636	7586	7876	8178	0.00	14.32	18.69	23.24
48	1438	1666	1693	1778	0.00	15.86	17.73	23.64
<b>Claire</b>								
<b>QP</b>	<b>NoFMO</b>	<b>FMO2</b>	<b>FMO4</b>	<b>FMO8</b>	<b>NoFMO</b>	<b>FMO2</b>	<b>FMO4</b>	<b>FMO8</b>
8	41561	44488	48024	49727	0.00	7.04	15.55	19.65
16	18594	19249	19372	19636	0.00	3.52	4.18	5.60
24	11307	12035	12212	12476	0.00	6.44	8.00	10.34
32	5737	6026	6170	6265	0.00	5.04	7.55	9.20
40	2283	2520	2561	2594	0.00	10.38	12.18	13.62
48	507	522	503	524	0.00	2.96	-0.79	3.35
<b>Foreman</b>								
<b>QP</b>	<b>NoFMO</b>	<b>FMO2</b>	<b>FMO4</b>	<b>FMO8</b>	<b>NoFMO</b>	<b>FMO2</b>	<b>FMO4</b>	<b>FMO8</b>
8	66021	69069	71937	73353	0.00	4.62	8.96	11.11
16	59033	62042	64286	64982	0.00	5.10	8.90	10.08
24	39903	43760	44428	45394	0.00	9.67	11.34	13.76
32	20868	23311	24216	24407	0.00	11.71	16.04	16.96
40	10175	11664	12459	12900	0.00	14.63	22.45	26.78
48	2827	3642	4057	4287	0.00	28.83	43.51	51.64

## A-2 Comparison of $R^2$ values at different QP and at different Rates.

**Table A-2.1** Comparison of  $R^2$  values of the proposed header bits model as compared with (Kwon, 2007) from four video sequences encoded at different QP and different number of FMO slice groups.

Akiyo	NoFMO		FMO2		FMO4		FMO8	
	Proposed	Kwon	Proposed	Kwon	Proposed	Kwon	Proposed	Kwon
8	0.956	0.950	0.958	0.953	0.949	0.942	0.930	0.914
16	0.965	0.961	0.959	0.951	0.958	0.948	0.925	0.900
24	0.943	0.937	0.945	0.938	0.938	0.923	0.905	0.862
32	0.806	0.797	0.791	0.770	0.796	0.738	0.723	0.489
40	0.320	0.282	0.375	0.259	0.291	-0.225	0.298	-1.941
Ave	0.798	0.785	0.806	0.774	0.787	0.665	0.756	0.245
Carphone	NoFMO		FMO2		FMO4		FMO8	
	Proposed	Kwon	Proposed	Kwon	Proposed	Kwon	Proposed	Kwon
8	0.909	0.820	0.915	0.818	0.920	0.843	0.923	0.850
16	0.937	0.896	0.944	0.913	0.954	0.924	0.957	0.930
24	0.947	0.931	0.950	0.927	0.948	0.941	0.950	0.939
32	0.927	0.910	0.927	0.916	0.938	0.923	0.947	0.934
40	0.866	0.853	0.875	0.858	0.896	0.876	0.909	0.879
Ave	0.917	0.882	0.922	0.887	0.931	0.901	0.937	0.907
Claire	NoFMO		FMO2		FMO4		FMO8	
	Proposed	Kwon	Proposed	Kwon	Proposed	Kwon	Proposed	Kwon
8	0.939	0.922	0.929	0.904	0.945	0.923	0.939	0.915
16	0.909	0.884	0.905	0.883	0.906	0.877	0.900	0.866
24	0.883	0.862	0.887	0.864	0.886	0.850	0.879	0.832
32	0.835	0.811	0.850	0.825	0.856	0.825	0.827	0.746
40	0.647	0.621	0.710	0.658	0.676	0.468	0.667	-0.190
Ave	0.843	0.820	0.856	0.827	0.854	0.789	0.842	0.634
Foreman	NoFMO		FMO2		FMO4		FMO8	
	Proposed	Kwon	Proposed	Kwon	Proposed	Kwon	Proposed	Kwon
8	0.884	0.792	0.850	0.740	0.840	0.726	0.823	0.695
16	0.854	0.758	0.812	0.701	0.835	0.727	0.817	0.726
24	0.813	0.708	0.771	0.644	0.798	0.716	0.791	0.715
32	0.639	0.520	0.677	0.543	0.667	0.590	0.691	0.625
40	0.577	0.564	0.465	0.408	0.549	0.533	0.629	0.578
Ave	0.753	0.668	0.715	0.607	0.738	0.658	0.750	0.668

**Table A-2.1** Comparison of  $R^2$  values of the proposed header bits model as compared with (Kwon, 2007) from four video sequences encoded at different Rates and different number of FMO slice groups.

<b>Akiyo</b>	<b>NoFMO</b>		<b>FMO2</b>		<b>FMO4</b>		<b>FMO8</b>	
<b>Rate</b>	<b>Proposed</b>	<b>Kwon</b>	<b>Proposed</b>	<b>Kwon</b>	<b>Proposed</b>	<b>Kwon</b>	<b>Proposed</b>	<b>Kwon</b>
<b>20</b>	0.871	0.869	0.848	0.841	0.832	0.809	0.792	0.686
<b>32</b>	0.902	0.896	0.904	0.894	0.893	0.876	0.878	0.830
<b>48</b>	0.929	0.923	0.933	0.928	0.926	0.915	0.906	0.873
<b>64</b>	0.920	0.912	0.928	0.921	0.937	0.928	0.916	0.891
<b>96</b>	0.947	0.942	0.945	0.937	0.941	0.931	0.918	0.894
<b>Carphone</b>	<b>NoFMO</b>		<b>FMO2</b>		<b>FMO4</b>		<b>FMO8</b>	
<b>Rate</b>	<b>Proposed</b>	<b>Kwon</b>	<b>Proposed</b>	<b>Kwon</b>	<b>Proposed</b>	<b>Kwon</b>	<b>Proposed</b>	<b>Kwon</b>
<b>20</b>	0.773	0.763	0.843	0.832	0.856	0.840	0.847	0.776
<b>32</b>	0.832	0.823	0.878	0.861	0.849	0.828	0.832	0.792
<b>48</b>	0.875	0.867	0.838	0.814	0.841	0.823	0.785	0.736
<b>64</b>	0.909	0.903	0.888	0.867	0.872	0.854	0.843	0.808
<b>96</b>	0.915	0.906	0.902	0.882	0.897	0.874	0.880	0.860
<b>Claire</b>	<b>NoFMO</b>		<b>FMO2</b>		<b>FMO4</b>		<b>FMO8</b>	
<b>Rate</b>	<b>Proposed</b>	<b>Kwon</b>	<b>Proposed</b>	<b>Kwon</b>	<b>Proposed</b>	<b>Kwon</b>	<b>Proposed</b>	<b>Kwon</b>
<b>20</b>	0.840	0.831	0.870	0.853	0.885	0.868	0.845	0.773
<b>32</b>	0.878	0.855	0.875	0.853	0.899	0.878	0.844	0.794
<b>48</b>	0.894	0.874	0.888	0.872	0.899	0.881	0.911	0.887
<b>64</b>	0.908	0.884	0.915	0.900	0.913	0.894	0.901	0.869
<b>96</b>	0.907	0.879	0.904	0.889	0.908	0.887	0.893	0.850
<b>Foreman</b>	<b>NoFMO</b>		<b>FMO2</b>		<b>FMO4</b>		<b>FMO8</b>	
<b>Rate</b>	<b>Proposed</b>	<b>Kwon</b>	<b>Proposed</b>	<b>Kwon</b>	<b>Proposed</b>	<b>Kwon</b>	<b>Proposed</b>	<b>Kwon</b>
<b>20</b>	0.639	0.621	0.606	0.588	0.787	0.768	0.820	0.763
<b>32</b>	0.669	0.591	0.700	0.689	0.740	0.728	0.786	0.758
<b>48</b>	0.659	0.549	0.633	0.549	0.683	0.618	0.605	0.538
<b>64</b>	0.687	0.564	0.630	0.505	0.700	0.616	0.619	0.534
<b>96</b>	0.739	0.669	0.731	0.640	0.758	0.663	0.696	0.614



**Table A-3.1** Comparison of SNR with the JVT Reference Rate Control

<b>SNR</b>	<b>JVTBaseline</b>					<b>SNR</b>	<b>Modified Frame Layer RC JVT</b>				
<b>Akiyo</b>	<b>NoFMO</b>	<b>FMO2</b>	<b>FMO4</b>	<b>FMO8</b>	<b>AvgSNR</b>	<b>Akiyo</b>	<b>NoFMO</b>	<b>FMO2</b>	<b>FMO4</b>	<b>FMO8</b>	<b>AvgSNR</b>
<b>20</b>	38.37	37.65	36.48	34.55	36.76	<b>20</b>	38.48	37.77	36.82	34.99	37.02
<b>32</b>	41.16	40.48	39.91	39.06	40.15	<b>32</b>	41.08	40.54	40.00	39.05	40.17
<b>48</b>	43.23	42.92	42.48	41.91	42.64	<b>48</b>	43.17	42.87	42.60	41.92	42.64
<b>64</b>	44.81	44.47	44.25	43.76	44.32	<b>64</b>	44.76	44.42	44.21	43.65	44.26
<b>96</b>	47.16	46.69	46.61	46.19	46.66	<b>96</b>	47.17	46.71	46.61	46.30	46.70
<b>Claire</b>	<b>NoFMO</b>	<b>FMO2</b>	<b>FMO4</b>	<b>FMO8</b>	<b>AvgSNR</b>	<b>Claire</b>	<b>NoFMO</b>	<b>FMO2</b>	<b>FMO4</b>	<b>FMO8</b>	<b>AvgSNR</b>
<b>20</b>	39.45	38.78	37.69	35.32	37.81	<b>20</b>	39.57	38.69	37.71	35.88	37.96
<b>32</b>	41.96	41.41	40.85	39.72	40.99	<b>32</b>	41.89	41.34	40.75	39.85	40.96
<b>48</b>	43.95	43.49	43.11	42.54	43.27	<b>48</b>	43.92	43.49	43.08	42.52	43.25
<b>64</b>	45.20	44.89	44.60	44.17	44.72	<b>64</b>	45.20	44.91	44.63	44.21	44.74
<b>96</b>	46.89	46.67	46.46	46.22	46.56	<b>96</b>	46.91	46.70	46.49	46.25	46.59
<b>Carphone</b>	<b>NoFMO</b>	<b>FMO2</b>	<b>FMO4</b>	<b>FMO8</b>	<b>AvgSNR</b>	<b>Carphone</b>	<b>NoFMO</b>	<b>FMO2</b>	<b>FMO4</b>	<b>FMO8</b>	<b>AvgSNR</b>
<b>20</b>	30.31	29.30	28.44	26.64	28.67	<b>20</b>	30.66	29.83	29.03	27.44	29.24
<b>32</b>	32.73	31.98	31.33	30.18	31.56	<b>32</b>	32.81	32.22	31.69	30.66	31.84
<b>48</b>	34.55	34.24	33.86	33.24	33.97	<b>48</b>	34.70	34.29	33.85	33.26	34.02
<b>64</b>	36.09	35.73	35.40	34.85	35.52	<b>64</b>	36.12	35.76	35.44	34.94	35.56
<b>96</b>	38.15	37.86	37.66	37.34	37.76	<b>96</b>	38.14	37.89	37.64	37.36	37.76
<b>Foreman</b>	<b>NoFMO</b>	<b>FMO2</b>	<b>FMO4</b>	<b>FMO8</b>	<b>AvgSNR</b>	<b>Foreman</b>	<b>NoFMO</b>	<b>FMO2</b>	<b>FMO4</b>	<b>FMO8</b>	<b>AvgSNR</b>
<b>20</b>	27.88	26.60	25.15	23.57	25.80	<b>20</b>	29.06	27.79	26.61	24.43	26.97
<b>32</b>	31.12	29.68	27.67	27.17	28.91	<b>32</b>	31.38	30.65	30.09	28.73	30.21
<b>48</b>	33.18	32.61	32.10	29.94	31.95	<b>48</b>	33.28	32.74	32.12	31.61	32.44
<b>64</b>	34.62	33.89	33.78	32.84	33.78	<b>64</b>	34.60	34.15	33.83	33.36	33.99
<b>96</b>	36.46	36.11	35.78	35.52	35.97	<b>96</b>	36.50	36.11	35.82	35.51	35.99





**Table A-3.2** Comparison of SNR standard deviation with the JVT Reference Rate Control

std SNR	JVTBaseline					std SNR	Modified Frame Layer RC JVT				
Akiyo	NoFMO	FMO2	FMO4	FMO8	AvgStd	Akiyo	NoFMO	FMO2	FMO4	FMO8	AvgStd
20	2.55	2.27	2.76	2.29	2.47	20	2.33	2.23	2.02	1.89	2.12
32	2.96	2.75	2.61	2.47	2.70	32	2.89	2.74	2.65	2.50	2.70
48	3.44	3.38	3.24	3.11	3.29	48	3.41	3.33	3.29	3.09	3.28
64	3.93	3.84	3.76	3.59	3.78	64	3.90	3.80	3.73	3.54	3.74
96	4.79	4.61	4.57	4.42	4.60	96	4.78	4.61	4.57	4.46	4.60
Claire	NoFMO	FMO2	FMO4	FMO8	AvgStd	Claire	NoFMO	FMO2	FMO4	FMO8	AvgStd
20	2.22	1.77	2.08	2.81	2.22	20	1.90	1.72	1.58	1.37	1.64
32	2.52	2.38	2.24	2.30	2.36	32	2.50	2.39	2.24	2.04	2.29
48	3.14	3.02	2.91	2.73	2.95	48	3.13	3.01	2.91	2.72	2.94
64	3.53	3.46	3.39	3.28	3.41	64	3.52	3.45	3.38	3.26	3.40
96	4.08	4.05	3.99	3.93	4.01	96	4.08	4.04	4.00	3.91	4.01
Carphone	NoFMO	FMO2	FMO4	FMO8	AvgStd	Carphone	NoFMO	FMO2	FMO4	FMO8	AvgStd
20	3.77	4.09	3.75	3.92	3.88	20	2.69	2.68	2.65	2.79	2.70
32	3.18	3.27	3.90	4.17	3.63	32	2.95	2.93	2.84	3.07	2.95
48	3.56	3.34	3.29	3.24	3.36	48	3.12	3.22	3.18	3.25	3.19
64	3.42	3.54	3.53	3.78	3.57	64	3.39	3.49	3.36	3.45	3.43
96	3.77	3.84	3.84	3.73	3.79	96	3.79	3.82	3.82	3.73	3.79
Foreman	NoFMO	FMO2	FMO4	FMO8	AvgStd	Foreman	NoFMO	FMO2	FMO4	FMO8	AvgStd
20	4.74	4.77	4.82	4.06	4.60	20	2.40	2.34	2.28	2.39	2.35
32	2.82	4.05	5.81	5.17	4.46	32	2.12	1.71	1.66	2.25	1.94
48	2.20	2.35	2.71	5.42	3.17	48	1.76	1.80	2.44	1.90	1.98
64	1.91	3.02	2.06	3.39	2.60	64	1.96	1.94	2.00	1.96	1.96
96	2.34	2.27	2.37	2.30	2.32	96	2.33	2.26	2.31	2.34	2.31



**Table A-3.3** Comparison of number of Frame skipped with the JVT Reference Rate Control

<b>Skip</b>	<b>JVTBaseline</b>					<b>Skip</b>	<b>Modified Frame Layer RC JVT</b>				
<b>Akiyo</b>	<b>NoFMO</b>	<b>FMO2</b>	<b>FMO4</b>	<b>FMO8</b>	<b>Total</b>	<b>Akiyo</b>	<b>NoFMO</b>	<b>FMO2</b>	<b>FMO4</b>	<b>FMO8</b>	<b>Total</b>
<b>20</b>	3	2	13	21	39	<b>20</b>	0	3	1	4	8
<b>32</b>	0	0	0	0	0	<b>32</b>	0	0	0	0	0
<b>48</b>	0	0	0	0	0	<b>48</b>	0	0	0	0	0
<b>64</b>	0	0	0	0	0	<b>64</b>	0	0	0	0	0
<b>96</b>	0	0	0	0	0	<b>96</b>	0	0	0	0	0
<b>Claire</b>	<b>NoFMO</b>	<b>FMO2</b>	<b>FMO4</b>	<b>FMO8</b>	<b>Total</b>	<b>Claire</b>	<b>NoFMO</b>	<b>FMO2</b>	<b>FMO4</b>	<b>FMO8</b>	<b>Total</b>
<b>20</b>	2	0	5	19	26	<b>20</b>	0	0	0	0	0
<b>32</b>	0	0	0	2	2	<b>32</b>	0	0	0	0	0
<b>48</b>	0	0	0	0	0	<b>48</b>	0	0	0	0	0
<b>64</b>	0	0	0	0	0	<b>64</b>	0	0	0	0	0
<b>96</b>	0	0	0	0	0	<b>96</b>	0	0	0	0	0
<b>Carphone</b>	<b>NoFMO</b>	<b>FMO2</b>	<b>FMO4</b>	<b>FMO8</b>	<b>Total</b>	<b>Carphone</b>	<b>NoFMO</b>	<b>FMO2</b>	<b>FMO4</b>	<b>FMO8</b>	<b>Total</b>
<b>20</b>	11	15	19	41	86	<b>20</b>	0	1	1	4	6
<b>32</b>	1	3	7	12	23	<b>32</b>	0	0	0	1	1
<b>48</b>	1	0	0	0	1	<b>48</b>	0	0	0	0	0
<b>64</b>	0	0	0	1	1	<b>64</b>	0	0	0	0	0
<b>96</b>	0	0	0	0	0	<b>96</b>	0	0	0	0	0
<b>Foreman</b>	<b>NoFMO</b>	<b>FMO2</b>	<b>FMO4</b>	<b>FMO8</b>	<b>Total</b>	<b>Foreman</b>	<b>NoFMO</b>	<b>FMO2</b>	<b>FMO4</b>	<b>FMO8</b>	<b>Total</b>
<b>20</b>	21	30	42	50	143	<b>20</b>	3	3	4	8	18
<b>32</b>	4	12	32	29	77	<b>32</b>	1	0	0	1	2
<b>48</b>	1	1	2	20	24	<b>48</b>	0	0	1	3	4
<b>64</b>	2	2	0	4	8	<b>64</b>	0	0	0	1	1
<b>96</b>	0	0	0	0	0	<b>96</b>	0	0	0	0	0



**Table A-3.4** Comparison of Bit rates with the JVT Reference Rate Control

Rate	JVTBaseline					Rate	Modified Frame Layer RC JVT				
Akiyo	NoFMO	FMO2	FMO4	FMO8	AvgRate	Akiyo	NoFMO	FMO2	FMO4	FMO8	AvgRate
20	199944	200296	201832	201656	200932	20	201344	200304	199816	198960	200106
32	320160	319768	320512	319472	319978	32	319888	319480	319584	319816	319692
48	480600	480208	480088	480280	480294	48	479968	479592	480896	479736	480048
64	641728	640248	639800	639424	640300	64	641400	641712	639240	639768	640530
96	958928	960216	959224	960016	959596	96	960912	959408	962352	959024	960424
<b>Claire</b>	<b>NoFMO</b>	<b>FMO2</b>	<b>FMO4</b>	<b>FMO8</b>	<b>AvgRate</b>	<b>Claire</b>	<b>NoFMO</b>	<b>FMO2</b>	<b>FMO4</b>	<b>FMO8</b>	<b>AvgRate</b>
20	200208	201008	201096	202376	201172	20	200296	199568	199328	200104	199824
32	320152	320720	320472	320984	320582	32	319824	319976	319688	319584	319768
48	481072	479384	479096	482592	480536	48	481288	480416	479840	481216	480690
64	639064	639632	639840	638864	639350	64	638536	639552	639800	639304	639298
96	959120	957880	959024	958408	958608	96	958560	958896	959272	959184	958978
<b>Carphone</b>	<b>NoFMO</b>	<b>FMO2</b>	<b>FMO4</b>	<b>FMO8</b>	<b>AvgRate</b>	<b>Carphone</b>	<b>NoFMO</b>	<b>FMO2</b>	<b>FMO4</b>	<b>FMO8</b>	<b>AvgRate</b>
20	202472	203496	203184	202816	202992	20	200288	200440	201256	200904	200722
32	322424	320816	321792	324176	322302	32	320024	320424	321328	321840	320904
48	480064	480400	481936	479752	480538	48	480480	480776	480744	480576	480644
64	640616	639840	639832	642920	640802	64	640520	640912	639976	640184	640398
96	960408	959392	960752	957456	959502	96	961048	961072	959144	959096	960090
<b>Foreman</b>	<b>NoFMO</b>	<b>FMO2</b>	<b>FMO4</b>	<b>FMO8</b>	<b>AvgRate</b>	<b>Foreman</b>	<b>NoFMO</b>	<b>FMO2</b>	<b>FMO4</b>	<b>FMO8</b>	<b>AvgRate</b>
20	203288	202760	203880	203216	203286	20	201536	201632	201336	203088	201898
32	319880	319640	324344	325240	322276	32	321320	320664	321920	321096	321250
48	480992	480160	486152	487368	483668	48	482264	480832	480248	480840	481046
64	641120	640536	638360	643480	640874	64	641296	645280	644696	644656	643982
96	960016	964664	958664	963448	961698	96	961736	967800	965536	967072	965536

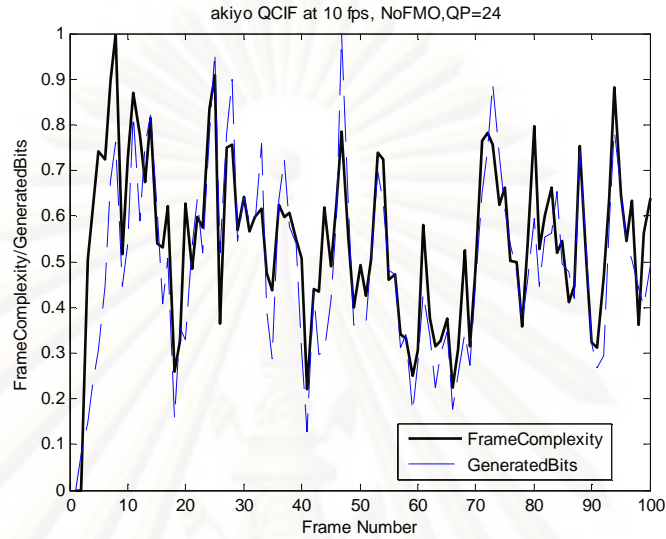
**Table A-3.5** Comparison of Rate Control Performance with JVT using Football Sequence

JVTBaseline						Modified Frame Layer RC JVT					
PSNR	NoFMO	FMO2	FMO4	FMO8	Avg.	PSNR	NoFMO	FMO2	FMO4	FMO8	Avg.
20	24.33	23.54	22.80	21.73	23.10	20	24.30	23.63	22.98	21.91	23.21
32	25.92	25.41	24.99	24.33	25.16	32	25.93	25.42	25.01	24.26	25.15
48	27.47	27.10	26.75	26.28	26.90	48	27.48	27.09	26.76	26.24	26.89
64	28.64	28.33	28.05	27.70	28.18	64	28.63	28.33	28.04	27.70	28.18
96	30.43	30.19	29.98	29.71	30.08	96	30.42	30.20	30.00	29.73	30.09
PSNR std	NoFMO	FMO2	FMO4	FMO8	Avg.	PSNR std	NoFMO	FMO2	FMO4	FMO8	Avg.
20	0.91	1.26	1.48	1.56	1.30	20	0.86	1.01	1.07	1.18	1.03
32	0.75	0.79	0.86	0.91	0.83	32	0.73	0.77	0.82	0.95	0.82
48	0.66	0.63	0.63	0.68	0.66	48	0.63	0.61	0.66	0.72	0.66
64	0.62	0.62	0.62	0.66	0.63	64	0.63	0.60	0.58	0.64	0.61
96	0.73	0.73	0.73	0.70	0.73	96	0.72	0.75	0.78	0.64	0.72
Skip	NoFMO	FMO2	FMO4	FMO8	Total	Skip	NoFMO	FMO2	FMO4	FMO8	Total
20	1	6	16	36	59	20	0	1	2	3	6
32	0	0	0	0	0	32	0	0	0	0	0
48	0	0	0	0	0	48	0	0	0	0	0
64	0	0	0	0	0	64	0	0	0	0	0
96	0	0	0	0	0	96	0	0	0	0	0
Rate	NoFMO	FMO2	FMO4	FMO8	Avg.	Rate	NoFMO	FMO2	FMO4	FMO8	Avg.
20	202208	201000	201976	202352	201884	20	200032	199640	200472	200240	200096
32	319712	320944	320224	322184	320766	32	319984	319936	319496	319120	319634
48	479464	479456	479672	480672	479700	48	479088	479848	479184	479624	479436
64	639680	638960	639656	638608	639226	64	639040	639080	639592	641120	639708
96	961864	958864	960256	959320	960076	96	960128	960648	960304	961864	960736

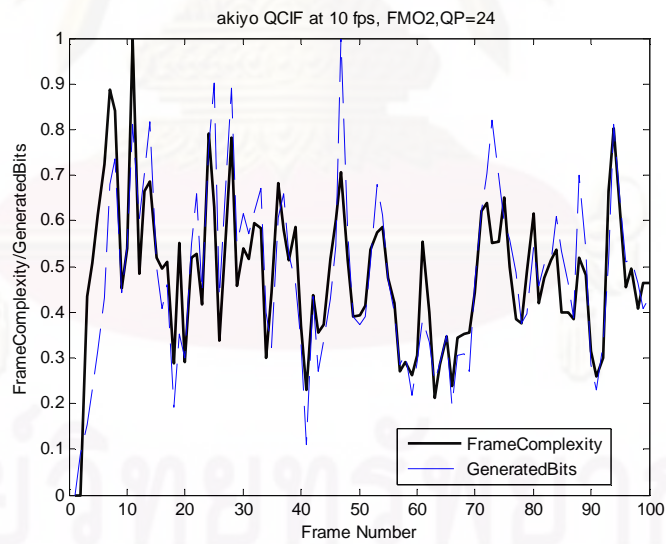


## APPENDIX B

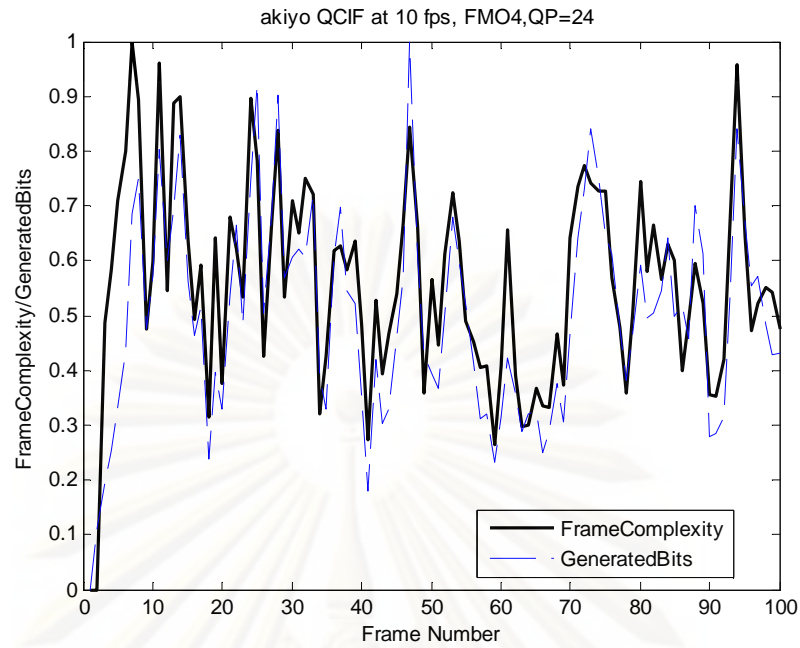
### B-1. Comparison of Complexity measure and Generated bits for Akiyo, Claire and Carphone and Foreman sequences, Figures B-1.1 – Figure B-1.16



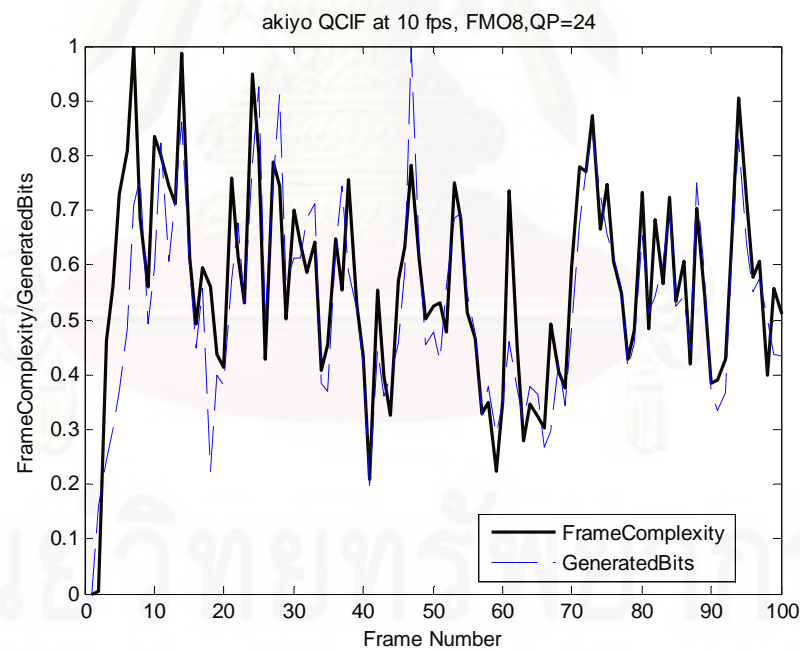
**Figure B-1.1** Akiyo sequence, NoFMO, Generated bits at QP=24 and Frame Complexity measure at Bit rate = 32 kbps



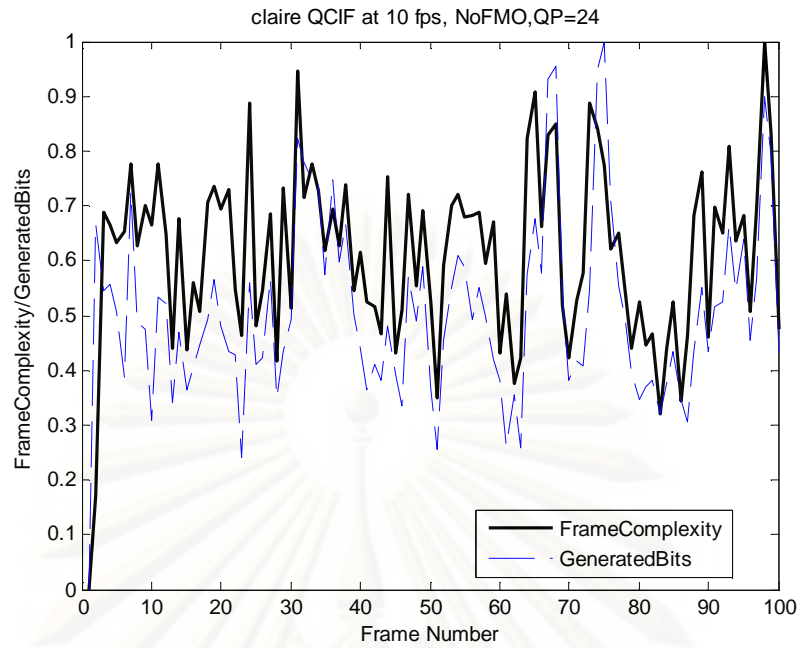
**Figure B-1.2** Akiyo sequence, FMO2, Generated bits at QP=24 and Frame Complexity measure at Bit rate = 32 kbps



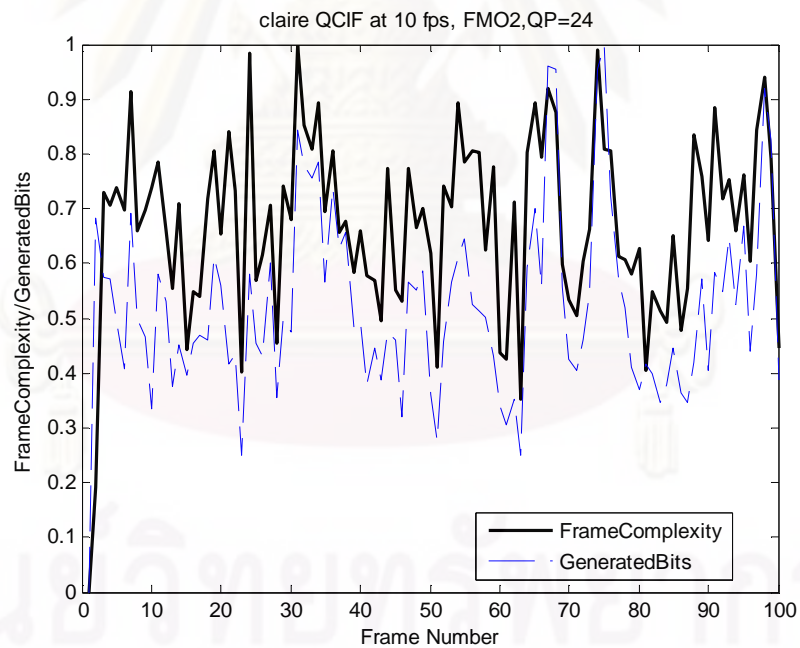
**Figure B-1.3** Akiyo sequence, FMO4, Generated bits at QP=24 and Frame Complexity measure at Bit rate = 32 kbps



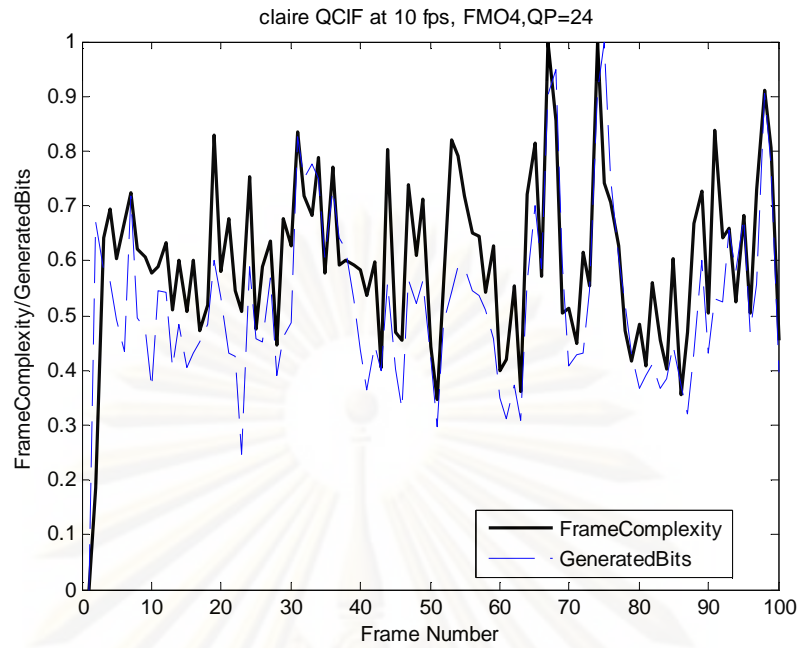
**Figure B-1.4** Akiyo sequence, FMO8, Generated bits at QP=24 and Frame Complexity measure at Bit rate = 32 kbps



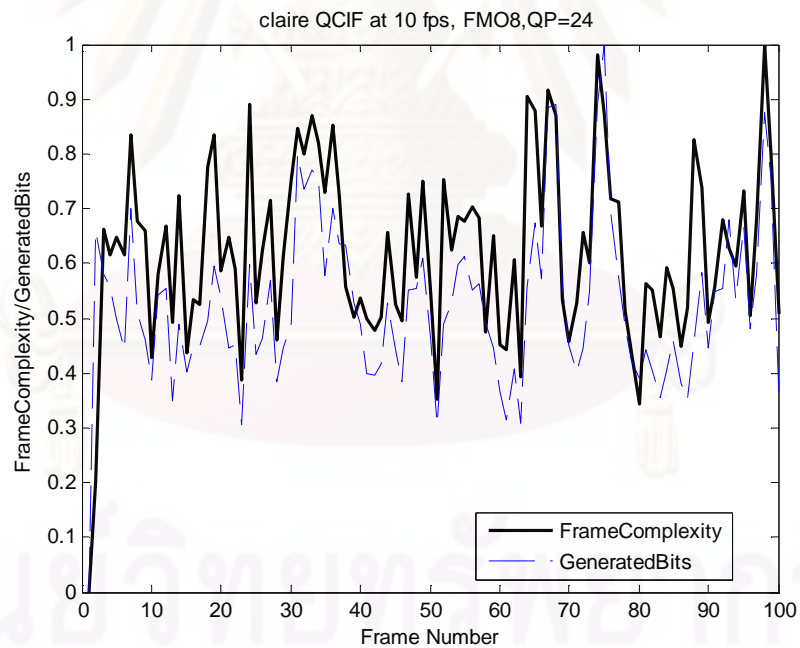
**Figure B-1.5** Claire sequence, NoFMO, Generated bits at QP=24 and Frame Complexity measure at Bit rate = 32 kbps



**Figure B-1.6** Claire sequence, FMO2, Generated bits at QP=24 and Frame Complexity measure at Bit rate = 32 kbps

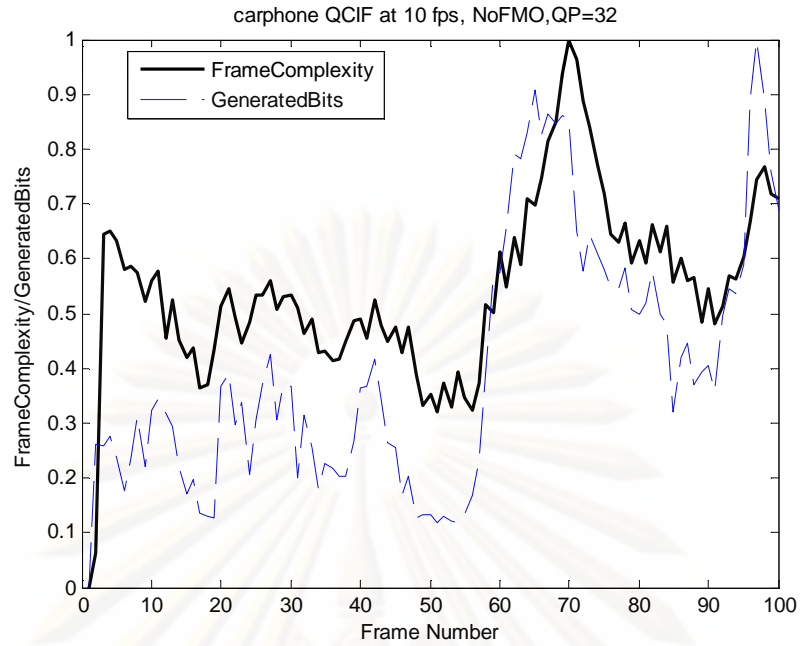


**Figure B-1.7** Claire sequence, FMO4, Generated bits at QP=24 and Frame Complexity measure at Bit rate = 32 kbps

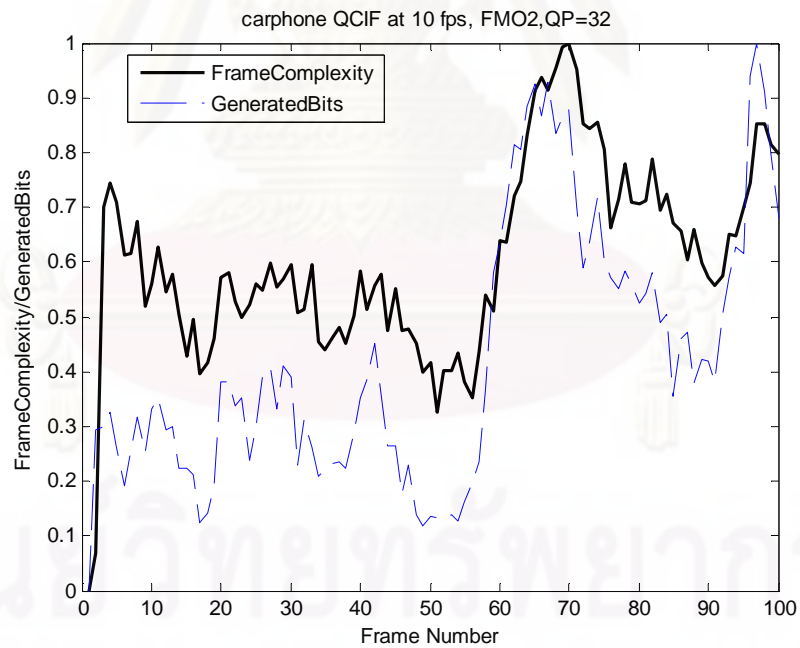


**Figure B-1.8** Claire sequence, FMO8, Generated bits at QP=24 and Frame Complexity measure at Bit rate = 32 kbps

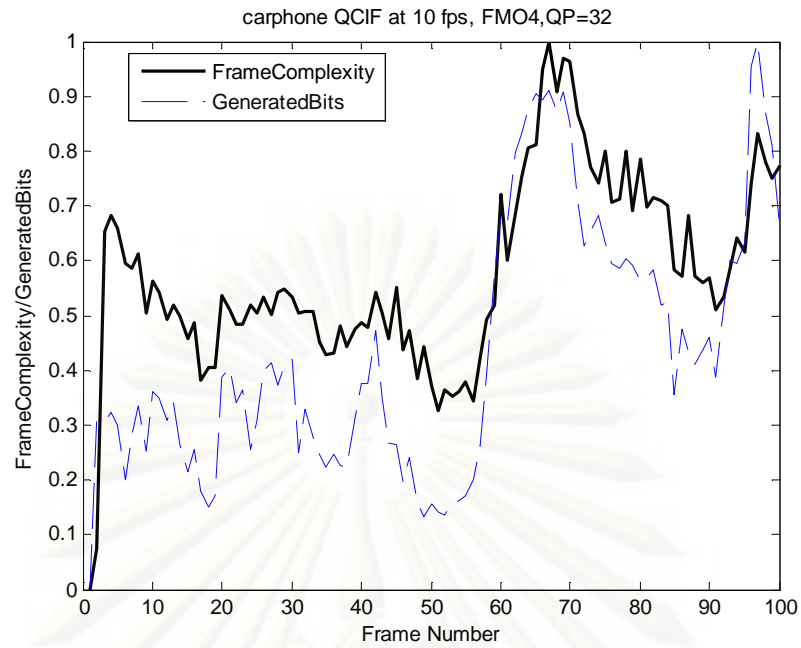




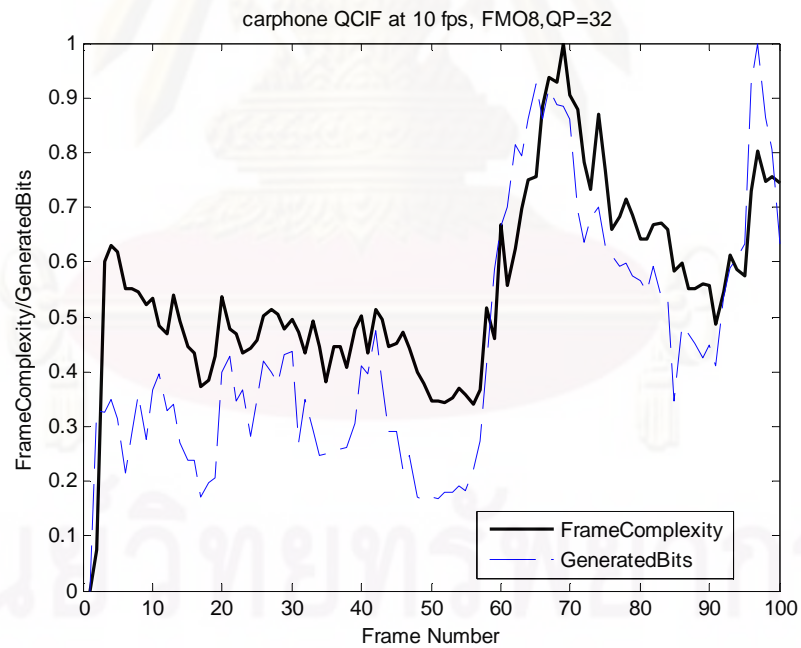
**Figure B-1.9** Carphone sequence, NoFMO, Generated bits at QP=32 and Frame Complexity measure at Bit rate = 48 kbps



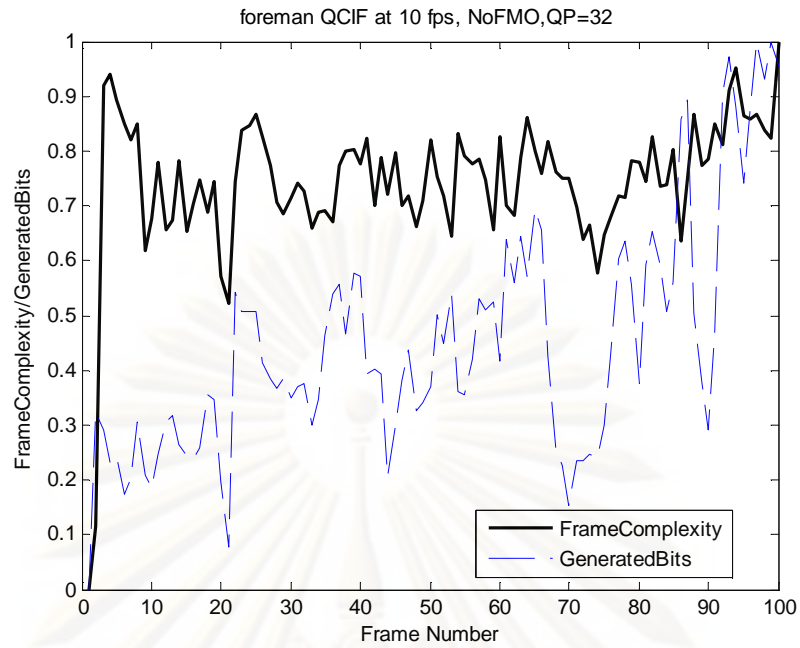
**Figure B-1.10** Carphone sequence, FMO2, Generated bits at QP=32 and Frame Complexity measure at Bit rate = 48 kbps



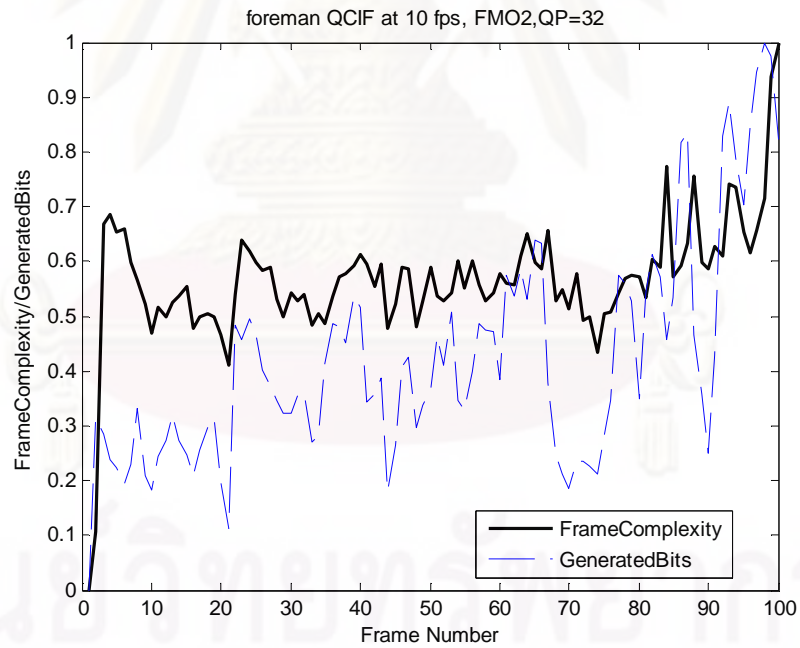
**Figure B-1.11** Carphone sequence, FMO4, Generated bits at QP=32 and Frame Complexity measure at Bit rate = 48 kbps



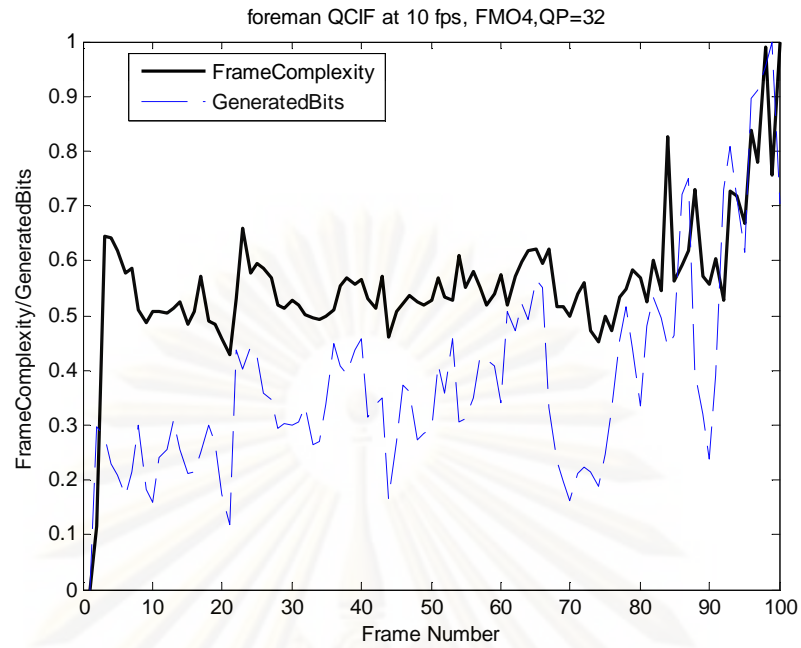
**Figure B-1.12** Carphone sequence, FMO8, Generated bits at QP=32 and Frame Complexity measure at Bit rate = 48 kbps



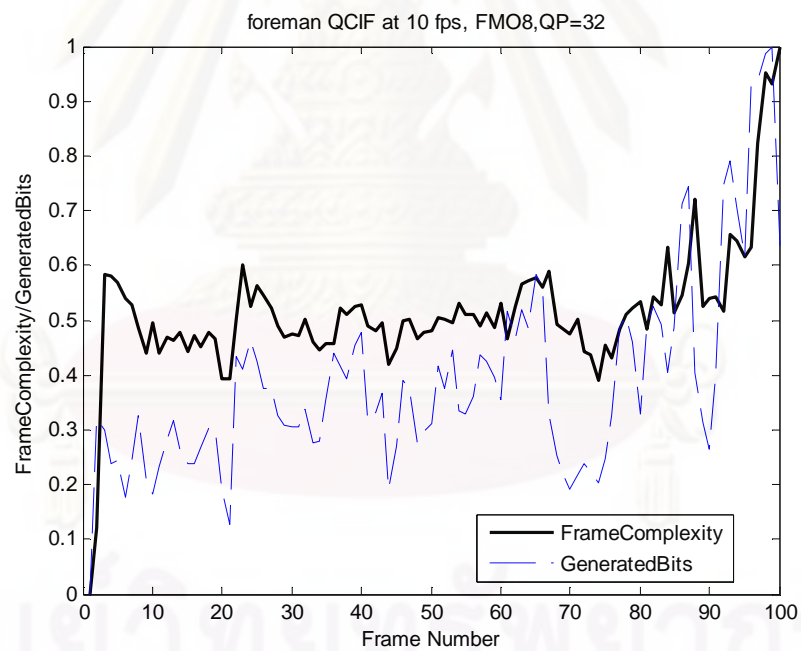
**Figure B-1.13** Foreman sequence, NoFMO, Generated bits at QP=32 and Frame Complexity measure at Bit rate = 48 kbps



**Figure B-1.14** Foreman sequence, FMO2, Generated bits at QP=32 and Frame Complexity measure at Bit rate = 48 kbps



**Figure B-1.15** Foreman sequence, FMO4, Generated bits at QP=32 and Frame Complexity measure at Bit rate = 48 kbps

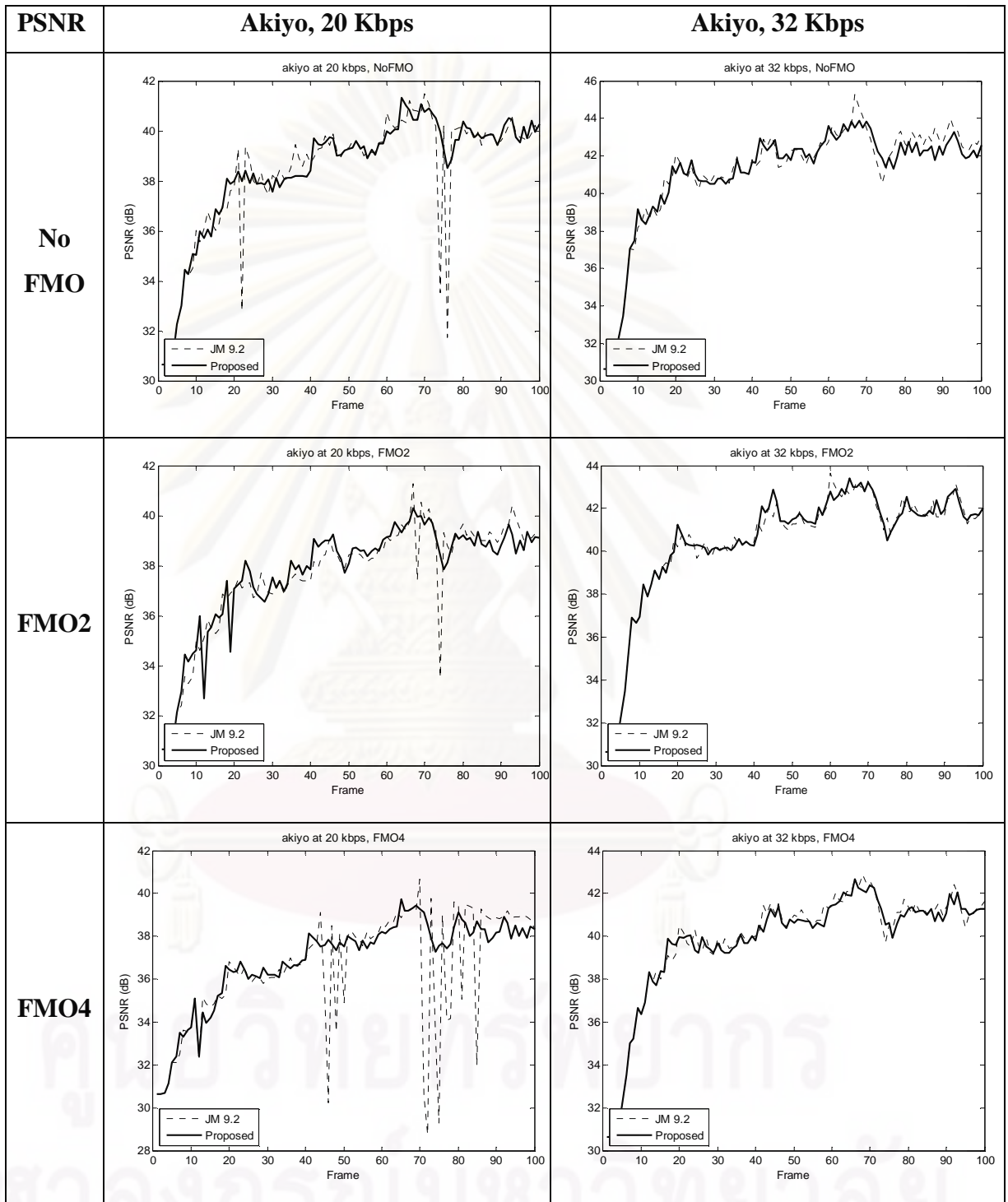


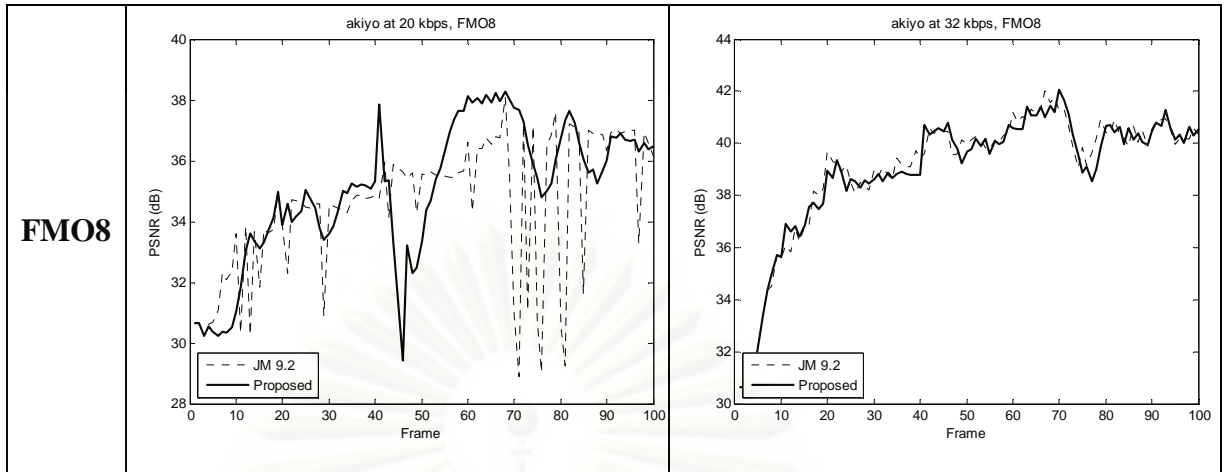
**Figure B-1.16** Foreman sequence, FMO8, Generated bits at QP=32 and Frame Complexity measure at Bit rate = 48 kbps



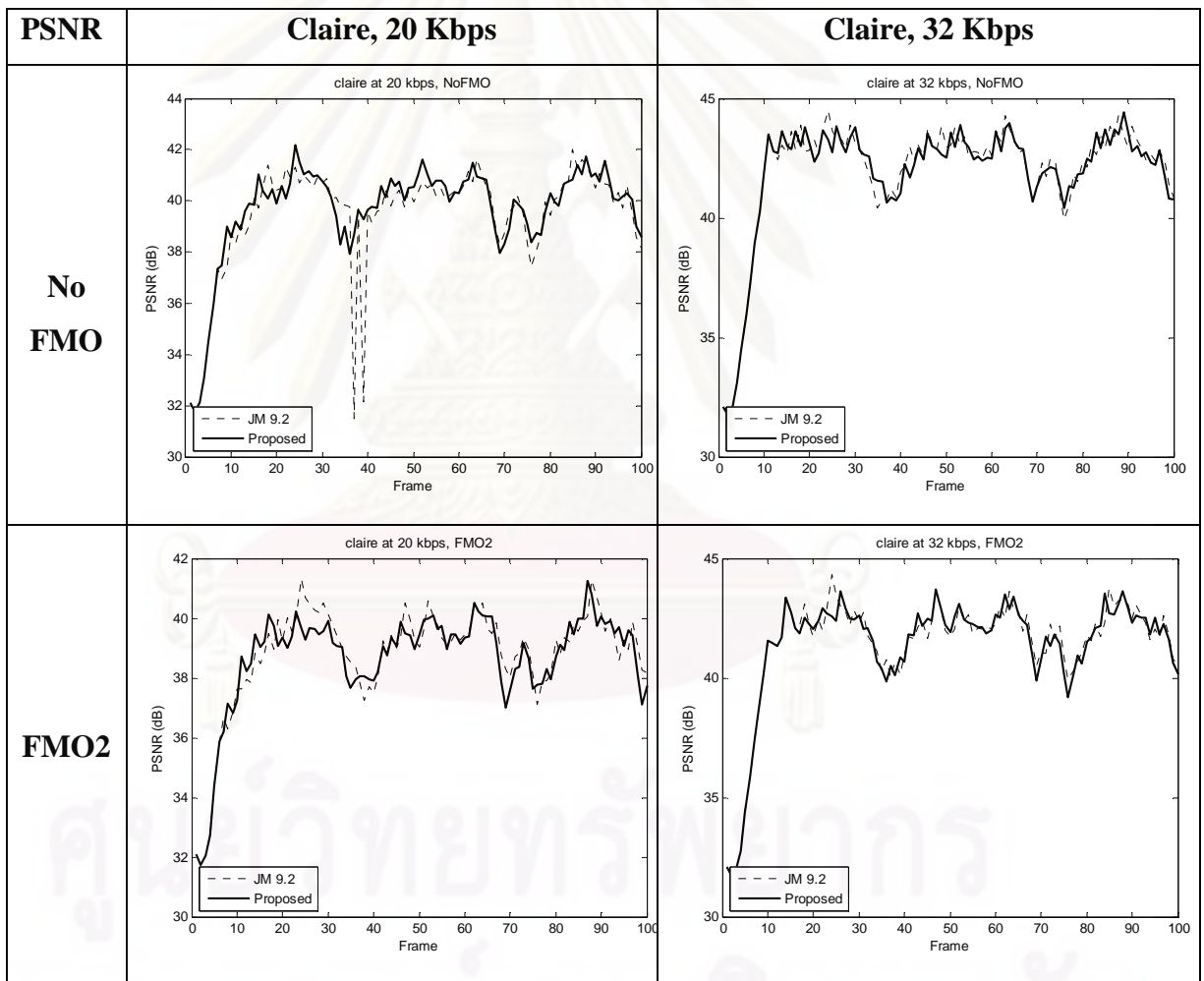
**B-2. Comparison of PSNR with JVT Rate Control at 20 kbps and 32 kbps for Akiyo, Claire, Carphone and Foreman sequences, Table B-2.1 – Table B-2.4**

**Table B-2.1.** Comparison of SNR for Akiyo sequence at 20 and 32 kbps

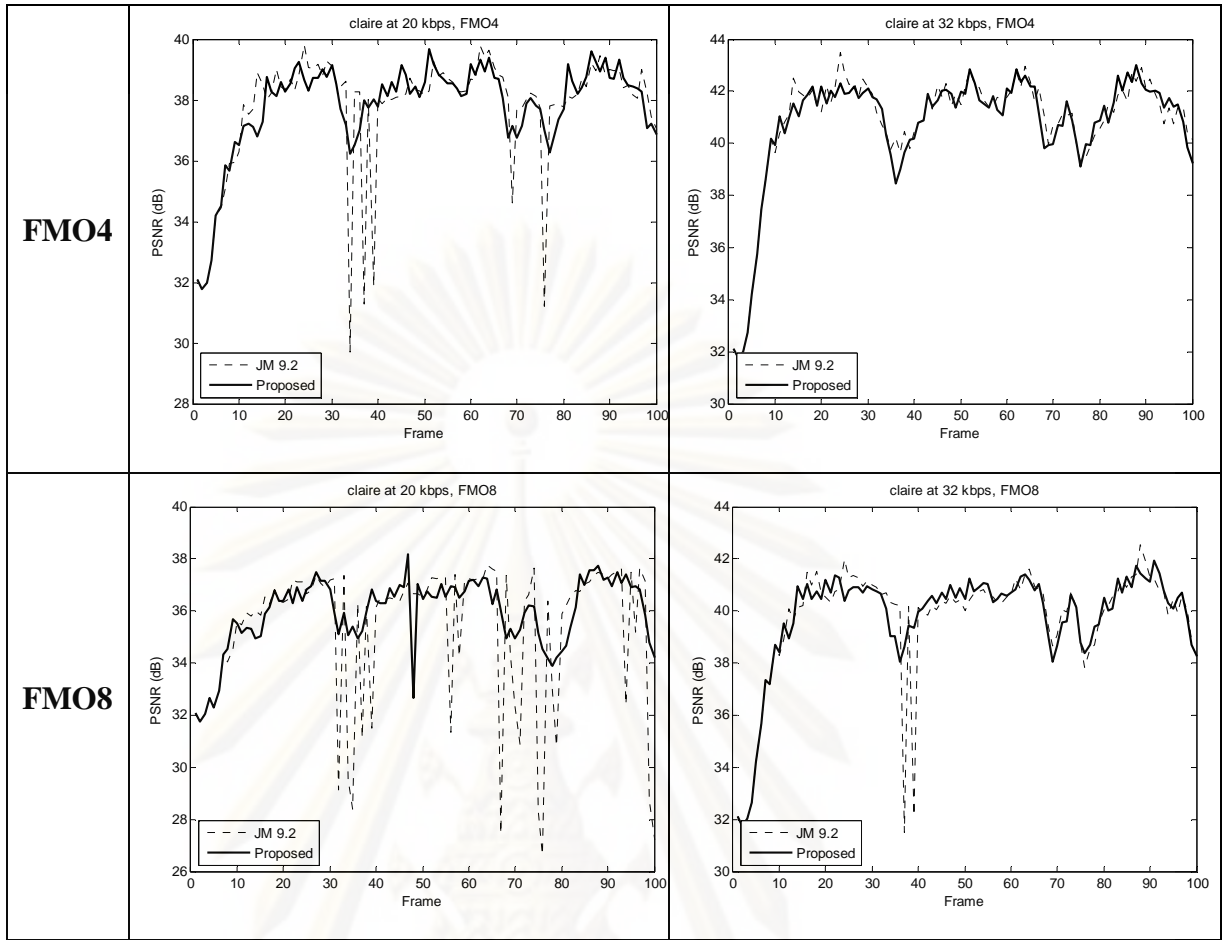




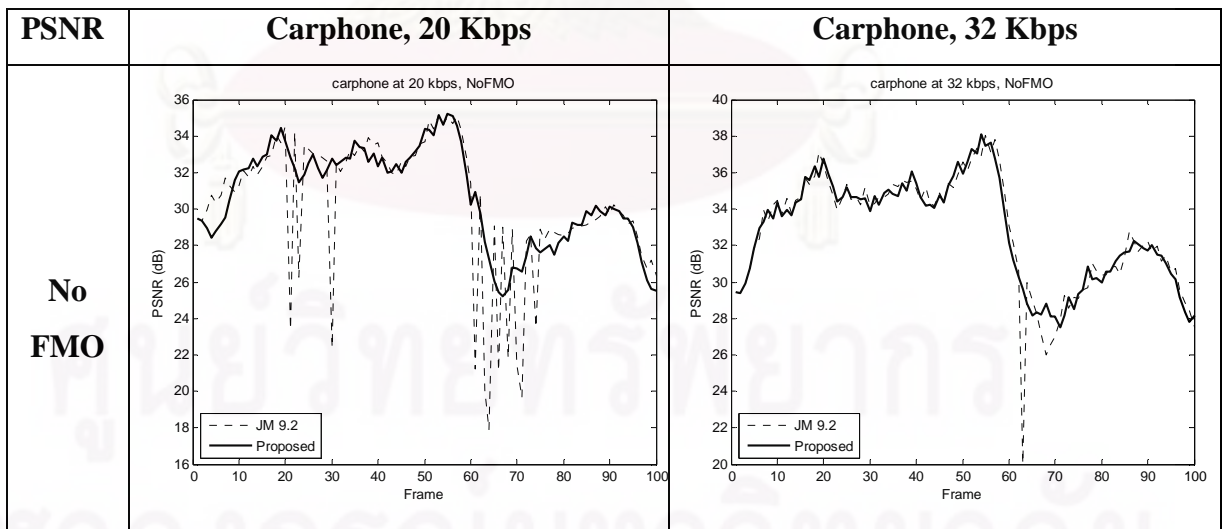
**Table B-2.2.** Comparison of SNR for Claire sequence at 20 and 32 kbps

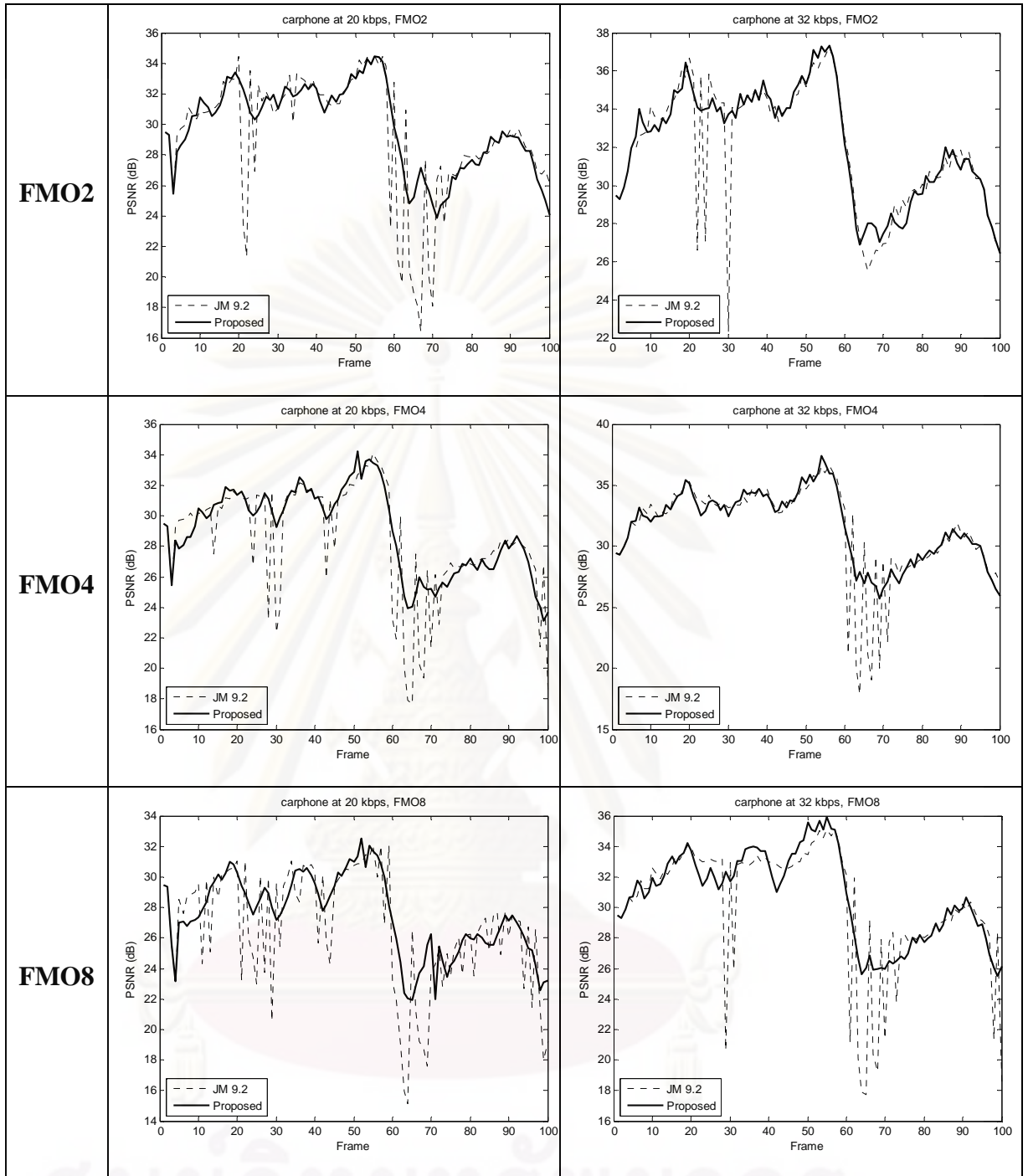


จุฬาลงกรณ์มหาวิทยาลัย



**Table B-2.3.** Comparison of SNR for Carphone sequence at 20 and 32 kbps

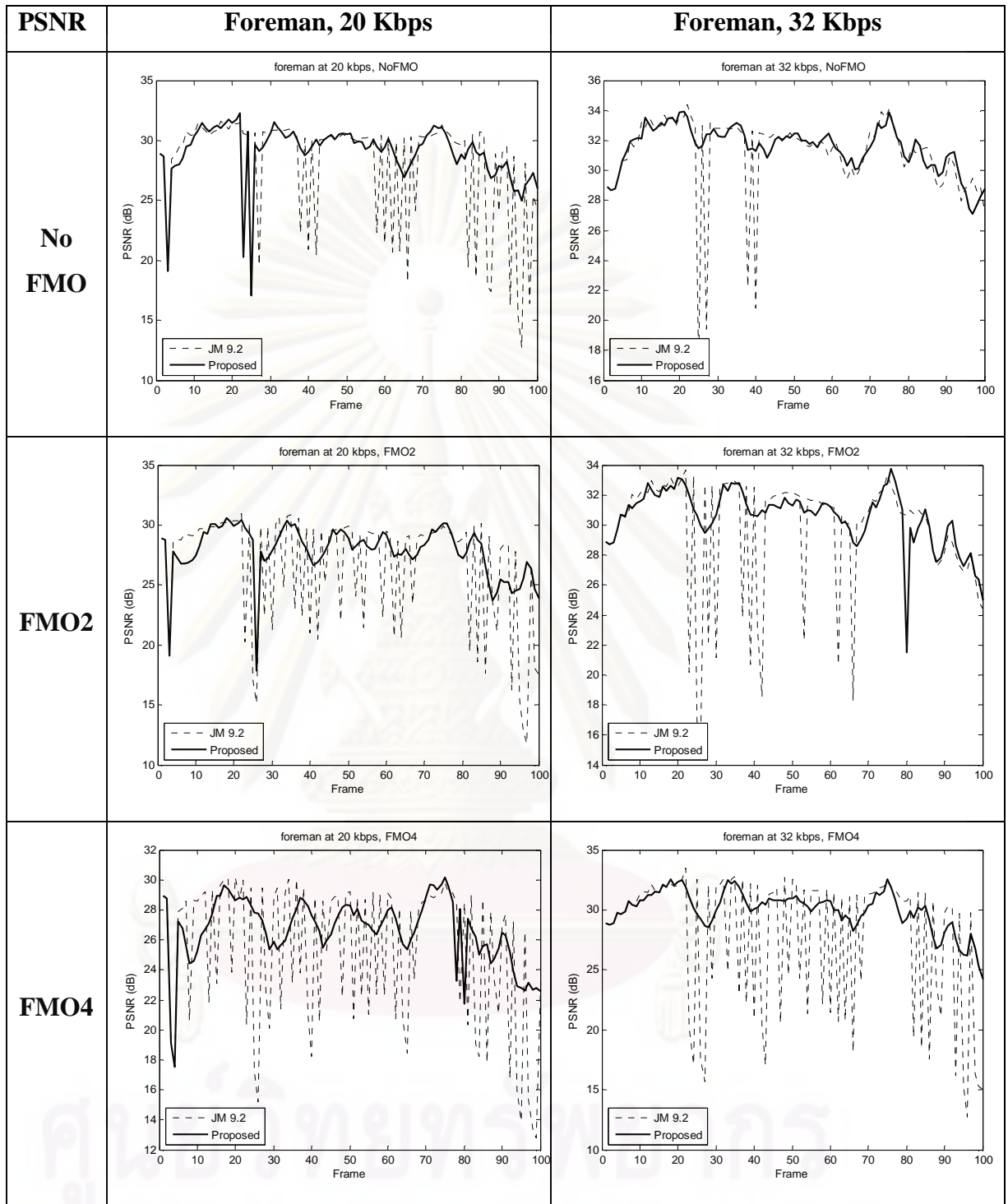


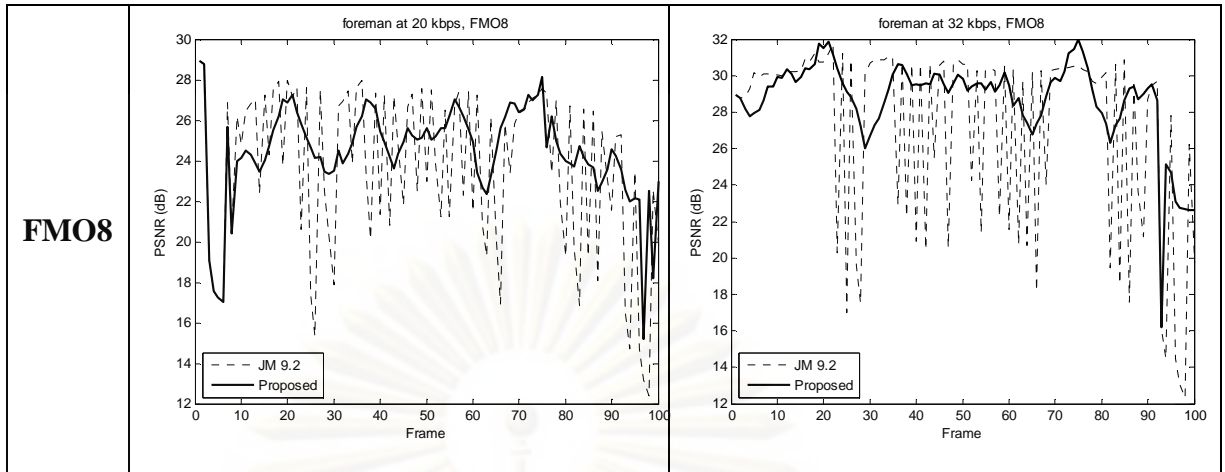


ศูนย์วิทยุโทรคมนาคม  
จุฬาลงกรณ์มหาวิทยาลัย

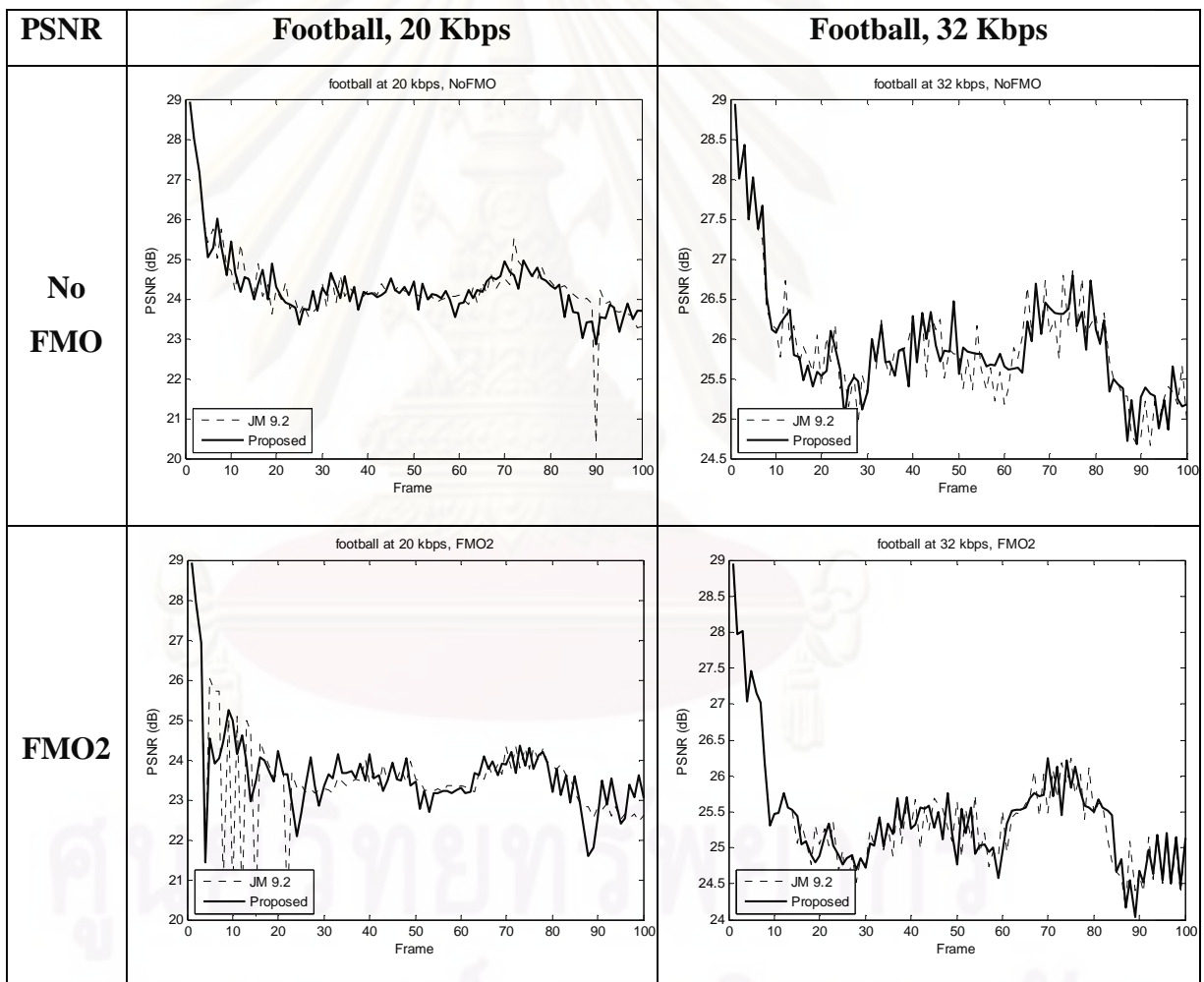


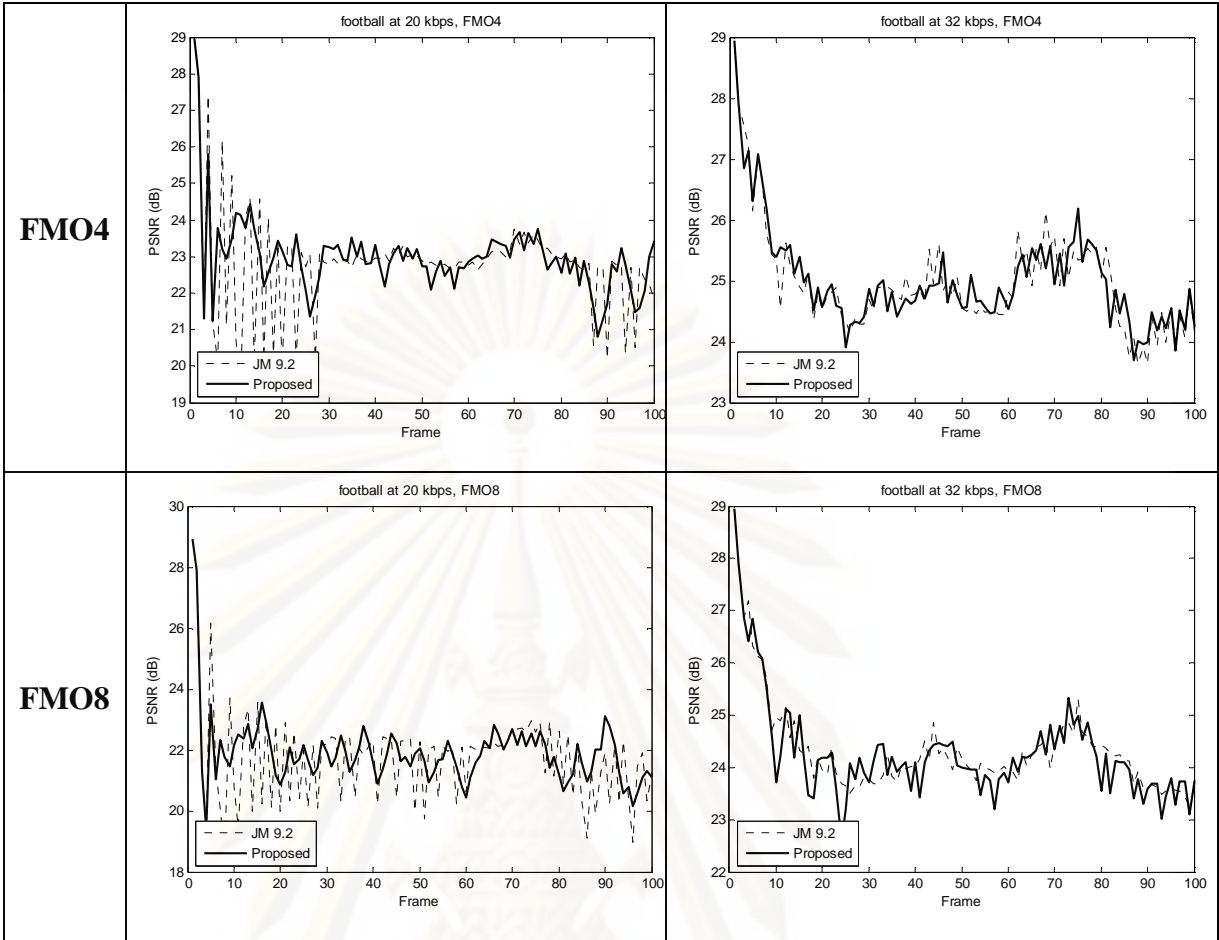
**Table B-2.4.** Comparison of SNR for Foreman sequence at 20 and 32 kbps



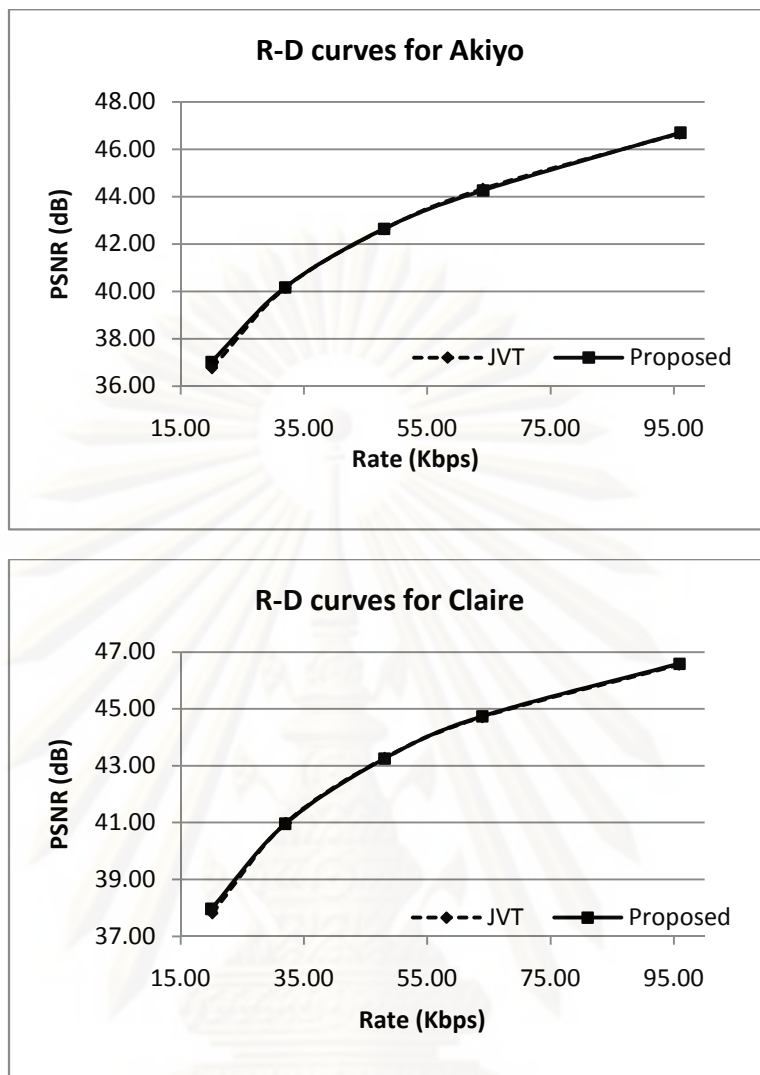


**Table B-2.5.** Comparison of SNR for Football sequence at 20 and 32 kbps





ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย



**Figure B.3** R-D curves comparison with JVT Rate Control for Akiyo and Claire

ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย



## VITAE

Mr. Rhandley D. Cajote received the Bachelor of Science (B.S.) degree in Electrical Engineering and the Master of Science (M.S.) degree in Electrical Engineering major in Computers and Communications from the Department of Electrical and Electronics Engineering, University of the Philippines, Diliman.



ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย