

ERROR RESILIENT CODING USING FLEXIBLE MACROBLOCK ORDERING IN
WIRED AND WIRELESS COMMUNICATIONS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ALİ MURAT DEMİRTAŞ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2008

Approval of the thesis:

**ERROR RESILIENT CODING USING FLEXIBLE MACROBLOCK
ORDERING IN WIRED AND WIRELESS COMMUNICATIONS**

submitted by ALİ MURAT DEMİRTAŞ in partial fulfillment of the requirements
for the degree of **Master of Science** in **Electrical and Electronics
Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, **Graduate School of Natural and Applied Sciences**

Prof. Dr. İsmet Erkmen _____
Head of Department, **Electrical and Electronics Engineering Dept.**

Prof. Dr. Gözde Bozdağı Akar _____
Supervisor, **Electrical and Electronics Engineering Dept.**

Examining Committee Members:

Assoc. Prof. Dr. Aydın Alatan _____
Electrical and Electronics Engineering Dept., METU

Prof. Dr. Gözde Bozdağı Akar _____
Electrical and Electronics Engineering Dept., METU

Assoc. Prof. Dr. Özgür Barış Akan _____
Electrical and Electronics Engineering Dept., METU

Assist. Prof. Dr. Çağatay Candan _____
Electrical and Electronics Engineering Dept., METU

Assist. Prof. Dr. Bülent Tavlı _____
Computer Engineering Dept., TOBB ETU

Date: September 4, 2008

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Ali Murat DEMİRTAŞ

Signature :

ABSTRACT

ERROR RESILIENT CODING USING FLEXIBLE MACROBLOCK ORDERING IN WIRED AND WIRELESS COMMUNICATIONS

Demirtaş, Ali Murat

M.Sc., Department of Electrical and Electronics Engineering

Supervisor: Prof. Dr. Akar, Güzde Bozdağı

September 2008, 104 pages

Error Resilient Coding tools are the methods to avoid or reduce the amount of corruption in video by altering the encoding algorithm. One of them is Flexible Macroblock Ordering (FMO) which provides us with ordering macroblocks of the frames flexibly. Six of them have definite ordering pattern and the last one, called explicit type, can get any order.

In this thesis two explicit type algorithms, one of which is new, are explained and the performance of different FMO types in wired and wireless communication are evaluated. The first algorithm separates the important blocks into separate packets, so it equalizes the importance of packets. The proposed method allocates the important macroblocks according to a checkerboard pattern and employs unequal error protection to protect them more. The simulations are performed for wired and wireless communication and Forward Error Correction is used in the second stage of the simulations. Lastly the results of the new algorithms are compared with the performance of the other FMO types. According to the simulations the Proposed algorithm performs better than others when the error rate is very high and FEC is employed.

Keywords: H.264, Flexible Macroblock Ordering, Error Resilient Coding, Wired Networks, Wireless Channels

ÖZ

ESNEK MAKROBLOK SIRALAMA KULLANARAK TELLİ VE TELSİZ KANALLARDA HATAYA ESNEK KODLAMA

Ali Murat Demirtaş

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Danışmanı: Prof. Dr. Gözde Bozdağı Akar

Eylül 2008, 104 sayfa

Kaynak kodlayıcıda kullanılmakta olan Hataya Dayanıklı Kodlama metotları, kodlayıcı algoritmasını değiştirerek videodaki bozulmayı azaltmak ya da engellemek için kullanılırlar. Bu metotlardan biri olan Esnek Makroblok Sıralama (EMS) çerçeveler içindeki makroblokların esnek biçimde sıralanmasını sağlar. Bunlardan altı tanesi belirli sıralama şekline sahiptir ve belirtik tip adı verilen sonuncu tip herhangi bir sıralama şekli alabilir. Bu çalışmada biri yeni olan iki belirtik tip algoritması anlatılacak ve değişik EMS tiplerinin telli ve telsiz iletimdeki performansları değerlendirilecektir. İlk algoritmada önemli bloklar farklı paketlere ayrılmakta, paketlerin önemliliklerini eşitlenmektedir. Önerilen yöntem, önemli makroblokları dama tahtası örüntüsünde ayırmakta ve onları eşit olmayan hata koruma kullanarak daha çok korumaktadır. Benzetimler, telli ve telsiz iletim için gerçekleştirilmektedir. Benzetimlerin ikinci aşamasında ileri hata düzeltme kullanılmaktadır. Son olarak yeni algoritmaların sonuçları diğer EMS tipleri ile karşılaştırılmaktadır. Simülasyon sonuçlarına göre , önerilen metot ileri hata düzeltme ile kullanıldığında, hata oranının yüksek olduğu ortamlarda diğer EMS tiplerinden daha iyi performans göstermektedir.

Anahtar Sözcükler: H.264, Esnek Makroblok Sıralama, Hataya Dayanıklı Kodlama, Telli Ağlar, Telsiz Kanallar

To My Family

ACKNOWLEDGEMENTS

I would like to thank all who have contributed one way or the other to make this thesis possible. Special thanks to Prof Dr. Gözde Bozdağı Akar, supervisor of this thesis, for her guidance, enthusiasm, outstanding support and patience. I would also thank Anil Aksay and Ahmet Serdar Tan for helping me in the development, for the valuable suggestions and technical support.

Thanks to all colleagues from Multi Media Research Group, for their suggestions and availability to help me whenever I need.

Finally I would like to thank my family, especially my brother. Their encouragement and patience give me the power to overcome the problems throughout my life.

TABLE OF CONTENTS

ABSTRACT.....	iv
ÖZ	vi
ACKNOWLEDGEMENTS.....	ix
TABLE OF CONTENTS.....	x
LIST OF TABLES.....	xii
LIST OF FIGURES.....	xiv
LIST OF ABBREVIATIONS	xvii
CHAPTERS	
1 INTRODUCTION	1
1.1 General Overview	1
1.2 Scope of the Thesis.....	3
1.3 Outline of the Dissertation	4
2 H.264 VIDEO CODING STANDARD	5
2.1 Comparison of H.264 with Other Standards	5
2.2 General Features	5
2.2.1 VCL (Video Coding Layer)	6
2.2.2 Network Abstraction Layer (NAL)	15
3 ERROR RESILIENT VIDEO CODING IN H.264.....	17
3.1 Overview	17
3.1.1 Encoder Error Resilient Methods	17
3.1.2 Decoder Error Resilient Methods	19
3.1.3 Encoder and Decoder Interactive Error Resilient Methods	20
4 FLEXIBLE MACROBLOCK ORDERING	21
4.1 Introduction.....	21
4.2 Types of Flexible Macroblock Ordering	23

4.2.1 Interleave Type	23
4.2.2 Dispersed Type.....	24
4.2.3 Foreground with Leftover Type	25
4.2.4 Box-Out Type	26
4.2.5 Raster Scan Type.....	28
4.2.6 Wipe Type.....	23
4.2.7 Explicit Type	31
4.3 Related Work on Explicit FMO.....	31
4.4 Explicit Type 1 Algorithm	34
4.5 Proposed Algorithm.....	34
5 SIMULATION ENVIRONMENT.....	39
5.1 Simulator	39
5.2 Wired Network Simulation Model	39
5.3 Wireless Channel Simulation Model	39
5.3.1 Channel Modeling	40
6 EXPERIMENTAL RESULTS	50
6.1 Test 1	53
6.1.1 Encoder Performance	53
6.1.2 Objective Quality	54
6.2 Test 2	67
6.2.1 Encoder Performance	67
6.2.2 Objective Quality	68
6.3 Test 3	76
6.3.1 Encoder Performance	76
6.3.2 Objective Quality	77
6.4 Test4	91
6.4.1 Encoder Performance	91
6.4.2 Objective Quality	92
7 CONCLUSIONS AND FUTURE WORK.....	100
REFERENCES	102

LIST OF TABLES

Table 5.1: SNR state boundary values of Fast Fading Channel	43
Table 5.2: SNR state boundary values of Slow Fading Channel.....	44
Table 5.3: State transition probabilities of fast fading channel	45
Table 5.4: State transition probabilities of slow fading channel	46
Table 6.1: Main characteristics of the video test sequences	50
Table 6.2: Encoder performance measurements of coded Foreman sequences using different FMO types for Test1	53
Table 6.3: Encoder performance measurements of coded Akiyo sequences using different FMO types for Test1	54
Table 6.4: Objective Quality Measurements of Foreman and Akiyo sequences for different FMO types when packet loss rate is %3.....	55
Table 6.5: Objective Quality Measurements of Foreman and Akiyo sequences for different FMO types when packet loss rate is %5.....	58
Table 6.6: Objective Quality Measurements of Foreman and Akiyo sequences for different FMO types when packet loss rate is %10	60
Table 6.7: Objective Quality Measurements of Foreman and Akiyo sequences for different FMO types when packet loss rate is %20	63
Table 6.8: Encoder performance measurements of coded Foreman sequence using different FMO types for Test2	67
Table 6.9: Encoder performance measurements of coded Akiyo sequence using different FMO types for Test2	68
Table 6.10: Objective Quality Measurements of Foreman sequence for different FMO types when wireless channel is slow fading.....	69
Table 6.11: Objective Quality Measurements of Akiyo sequence for different FMO types when wireless channel is slow fading.....	71
Table 6.12: Objective Quality Measurements of Foreman sequence for different FMO types when wireless channel is fast fading.....	72
Table 6.13: Objective Quality Measurements of Akiyo sequence for different FMO types when wireless channel is fast fading.....	73
Table 6.14: Encoder performance measurements of coded Foreman sequence using different FMO types for Test3	76

Table 6.15: Encoder performance measurements of coded Akiyo sequence using different FMO types for Test3	77
Table 6.16: Objective Quality Measurements of Foreman and Akiyo sequences for different FMO Types when packet loss rate is %3 and FEC is employed	78
Table 6.17 Objective Quality Measurements of Foreman and Akiyo sequences for different FMO Types when packet loss rate is %5 and FEC is employed	80
Table 6.18 Objective Quality Measurements of Foreman and Akiyo sequences for different FMO Types when packet loss rate is %10 and FEC is employed	83
Table 6.19 Objective Quality Measurements of Foreman and Akiyo sequences for different FMO Types when packet loss rate is %20 and FEC is employed	85
Table 6.20: Objective Quality Measurements of Foreman and Akiyo sequences for different FMO Types when packet loss rate is %20 and FEC is employed	88
Table 6.21: Encoder performance measurements of coded Foreman sequences using different FMO types for Test4.....	91
Table 6.22: Encoder performance measurements of coded Akiyo sequences using different FMO types for Test4	92
Table 6.23: Objective Quality Measurements of Foreman sequence for different FMO types when wireless channel is slow fading and FEC is employed	93
Table 6.24: Objective Quality Measurements of Akiyo sequence for different FMO types when wireless channel is slow fading and FEC is employed	93
Table 6.25: Objective Quality Measurements of Foreman sequence for different FMO types when wireless channel is fast fading and FEC is employed	96
Table 6.26: Objective Quality Measurements of Akiyo sequence for different FMO types when wireless channel is fast fading and FEC is employed	97

LIST OF FIGURES

Figure 1.1: Typical Video Communication System [2]	1
Figure 2.1: A QCIF frame with 3 slice	7
Figure 2.2: Sampling Format 4:2:0 [7]	8
Figure 2.3: Structure of VCL [6].....	8
Figure 2.4: Five Intra_4x4 prediction modes [6]	10
Figure 2.5: Segmentation of macroblocks for motion compensation [6]	12
Figure 2.6: Integer Transform Coding Matrix [6].....	13
Figure 2.7: Header of a NAL Unit [4]	15
Figure 4.1: Slice structure without using FMO [6].....	21
Figure 4.2: Interleave FMO Type (3 Slices Groups)[7]	23
Figure 4.3: Dispersed FMO Type (4 Slice Groups) [7]	25
Figure 4.4: Foreground with Leftover FMO Type (4 Slice Groups) [7]	26
Figure 4.5: Box-Out FMO Type (2 Slice Groups) [7].....	27
Figure 4.6: Box-Out Coding Sequence	28
Figure 4.7: Raster Scan FMO Type (2 Slice Groups) [7]	29
Figure 4.8: Wipe FMO Type (2 slices).....	30
Figure 4.9: Determining bit count of macroblocks and sorting their addresses	35
Figure 4.10: Step1, macroblocks are splitted into 2 groups according to their bit count numbers	36
Figure 4.11: Splitting the foreground and background regions using checkerboard pattern and bit count information, respectively	37
Figure 4.12: Adjusting slice group is of an isolated macroblock according to macroblock nubers of the groups	38
Figure 5.1: 2 state Gilbert Elliot Model [36]	40
Figure 5.2: SNR states of the wireless channel.....	43
Figure 5.3: FSMC representation of the wireless channel	44
Figure 5.4: SNR sequence of the slow fading channel	47
Figure 5.5: SNR sequence of the fast fading channel	47
Figure 6.1: PSNR values of different FMO types for Foreman sequence when packet loss rate is %3	56
Figure 6.2: PSNR values of different FMO types for Akiyo sequence when	

packet loss rate is %3	57
Figure 6.3: PSNR values of different FMO types for Foreman sequence when packet loss rate is %5	58
Figure 6.4: PSNR values of different FMO types for Akiyo sequence when packet loss rate is %5	59
Figure 6.5: PSNR values of different FMO types for Foreman sequence when packet loss rate is %10.....	61
Figure 6.6: PSNR values of different FMO types for Akiyo sequence when packet loss rate is %10.....	62
Figure 6.7: PSNR values of different FMO types for Foreman sequence when packet loss rate is %20.....	64
Figure 6.8: PSNR values of different FMO types for Akiyo sequence when packet loss rate is %20.....	65
Figure 6.9: Average PSNR values of FMO types for Test 1 Foreman sequence	66
Figure 6.10: Average PSNR values of FMO types for Test 1 Akiyo sequence ..	66
Figure 6.11: PSNR values of different FMO types for Foreman sequence when the channel is slow fading wireless channel	70
Figure 6.12: PSNR values of different FMO types for Akiyo sequence when the channel is slow fading wireless channel.....	71
Figure 6.13: PSNR values of different FMO types for Foreman sequence when the channel is fast fading wireless channel	73
Figure 6.14: PSNR values of different FMO types for Akiyo sequence when the channel is fast fading wireless channel.....	74
Figure 6.15: Average PSNR values of FMO types for Test 2 Foreman sequence	75
Figure 6.16: Average PSNR values of FMO types for Test 2 Akiyo sequence ..	75
Figure 6.17: PSNR values of different FMO types for Foreman sequence when packet loss rate is %3 and FEC is employed	79
Figure 6.18: PSNR values of different FMO types for Akiyo sequence when packet loss rate is %3 and FEC is employed	79
Figure 6.19: PSNR values of different FMO types for Foreman sequence when packet loss rate is %5 and FEC is employed	81
Figure 6.20: PSNR values of different FMO types for Akiyo sequence when packet loss rate is %5 and FEC is employed	82
Figure 6.21: PSNR values of different FMO types for Foreman sequence when packet loss rate is %10 and FEC is employed.....	84
Figure 6.22: PSNR values of different FMO types for Akiyo sequence when	

packet loss rate is %10 and FEC is employed.....	84
Figure 6.23: PSNR values of different FMO types for Foreman sequence when packet loss rate is %20 and FEC is employed.....	86
Figure 6.24: PSNR values of different FMO types for Akiyo sequence when packet loss rate is %20 and FEC is employed.....	87
Figure 6.25: PSNR values of different FMO types for Foreman sequence when packet loss rate is %30 and FEC is employed.....	88
Figure 6.26: PSNR values of different FMO types for Akiyo sequence when packet loss rate is %30 and FEC is employed.....	89
Figure 6.27: Average PSNR values of FMO types for Test 3 Foreman sequence	90
Figure 6.28: Average PSNR values of FMO types for Test 3 Akiyo sequence ..	90
Figure 6.29: PSNR values of different FMO types for Foreman sequence when wireless channel is slow fading and FEC is employed	94
Figure 6.30: PSNR values of different FMO types for Akiyo sequence when wireless channel is slow fading and FEC is employed	95
Figure 6.31: PSNR values of different FMO types for Foreman sequence when wireless channel is fast fading and FEC is employed	96
Figure 6.32: PSNR values of different FMO types for Akiyo sequence when wireless channel is fast fading and FEC is employed	97
Figure 6.33: Average PSNR values of FMO types for Test 4 Foreman sequence	98
Figure 6.34: Average PSNR values of FMO types for Test 4 Akiyo sequence ..	99

LIST OF ABBREVIATIONS

ARQ	Automatic Repeat Request
AVC	Advanced Video Coding
BER	Bit Error Rate
CAVLC	Context Adaptive Variable Length Coding
CABAC	Context Adaptive Binary Adaptive Coding
CRC	Cyclic Redundancy Check
DCT	Discrete Cosine Transform
ER	Error Resilience
EEP	Equal Error Protection
FEC	Forward Error Correction
FMO	Flexible Macroblock Ordering
FPS	Frame per Second
GOP	Group of Pictures
IDCT	Inverse Discrete Cosine Transform
IDR	Instantaneous Data Refresh
IEC	International Electrotechnical Commission
IP	Internet Protocol
ISDN	Integrated Switch Digital Network
ISO	International Organization for Standardization
ITU	International Telecommunication Union
ITU-T	Telecommunication Standardization Sector
JVT	Joint Video Team
LC	Layered Coding
MBAmap	Macroblock to Slice Allocation Map
MPEG	Moving Picture Experts Group
MDC	Multiple Description Coding
MTU	Maximum Transmission Unit
NAL	Network Abstraction Layer
NALU	NAL Unit
PSNR	Peak Signal-to-Noise Ratio
QCIF	Quarter Common Intermediate Format
PSTN	Public Switch Telephone Network
POCS	Projection onto Convex Sets
PPS	Picture Parameter Set
RPS	Reference Picture Selection
RS	Reed-Solomon
RTP	Real Time Transmission Protocol
RTCP	RTP control
SAD	Sum of Absolute Differences
SGid	Slice Group id
SNR	Signal to Noise Ratio
SPS	Sequence Parameter Set
SVC	Scalable Video Coding
UDP	User Datagram Protocol

UEP	Unequal Error Protection
VCEG	Video Coding Experts Group
VCL	Video Coding Layer
VLC	Variable Length Coding
XML	Extended Markup Language

CHAPTER 1

INTRODUCTION

1.1 General Overview

In our era, the use of multimedia in our daily life increases day by day. Many TV channels begin to broadcast their programs on the Internet. Besides that, video conferences reduce the time for transportation, which can be used for another aim, and videophone infrastructure is ready for most of the developed and developing countries. Moreover the popularity of YouTube and the studies of companies for 3G in mobile phones are significant evidences that show the tendency of people to visual communication. A simple communication architecture can be seen in Figure 1.1[2].

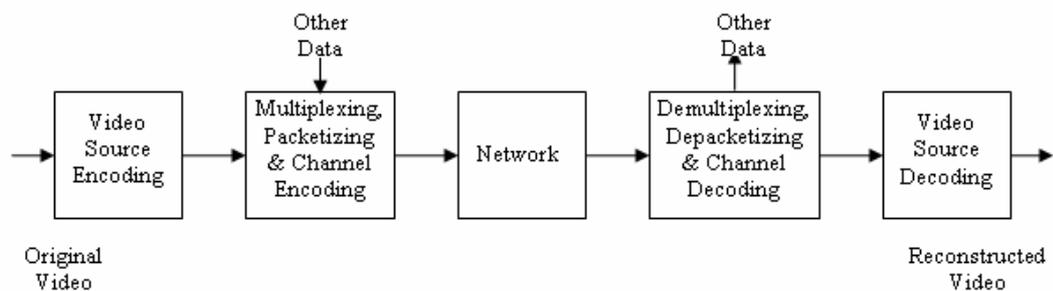


Figure 1.1: Typical Video Communication System [2]

There are several factors which accelerate the increase of multimedia usage. The improvement in the video codecs is one of the most important of them.

ITU (International Telecommunication Union) and ISO (International Organization for Standardization) are the two main bodies that develop video coding standards. Recently, they joined their efforts under the group of JVT (Joint Video Team) and created the H.264 AVC codec which is the most popular in the recent years [1].

H.264 codec can compress the video data two times better than the former codecs. Besides that, it has a special section to adapt the coded video to the transmission protocol which makes it a network friendly codec. However, the communication channels that are used to transmit the video data are error prone. Moreover, encoder utilizes all the redundancy in the video to compress it more. Hence, occurrence of an error causes crucial corruption in the reconstructed video.

In order to decrease the effect of error, several methods have been found. They are classified in three groups as follows. Methods applied at the encoder, methods applied at the decoder and methods where both encoder and decoder are used. Error Correction Coding is employed at the encoder, Error Concealment is applied at the decoder, and both encoder and decoder have to be used in order to apply ARQ. In addition to these methods, there are also Error Resilient methods of the coding standard. These methods are also member of one of the mentioned groups. All of these methods can be applied separately or together. However, they are not generally used concurrently because the channel has some strict limitations like delay and rate. Error Correction Coding is applied in the channel coding stage which is after the video coding [3]. The aim of this method is detecting or correcting errors that are found in the received video. The detection or correction can be achieved by the extra information which is calculated according to a known procedure and added to the transmitted video. The goal of Error Concealment [3] is decreasing the effect of damaged blocks by using the other blocks. These blocks are mostly the neighboring blocks of the corrupted blocks or the blocks of the previous or next frame which are at the same position with the corrupted blocks. ARQ is basically sending the packet again if it is lost or it is not received accurately. Although this method is the most successful method for rescuing the frames it can not be used in the real time communication because of its strict time constraints [2]. Lastly, there are error resilient coding tools. Main aim of these tools is to make the video data more robust to the errors in the communication channel by altering the encoding

procedure. There are several Error Resilient Coding methods. The methods of H.264 [4] are Intra Placement, Picture Segmentation, Reference Picture Selection, Data Partitioning, Redundant Slices, Parameter Sets, and Flexible Macroblock Ordering (FMO). Among these tools, FMO is one of the novel features of H.264/AVC codec standard.

Allocation of macroblocks to slices is performed using raster scanning pattern in regular encoding procedure. Hence, error concealment cannot be used efficiently if a packet is lost. FMO is developed to provide flexible allocation of macroblocks in order to provide more robust streams.

1.2 Scope of the Thesis

In the thesis, the performance of different types Flexible Macroblock Ordering [4] which is one of the Error Resilience tools in the H.264 is analyzed. When the macroblocks are distributed to the packets they are generally distributed in the raster scan order. Since raster scan separate neighboring blocks into the same packets, there will be a problem if an error occurs. On the other hand flexible macroblock provides more error robust allocation. Therefore the coded video data becomes more immune to the errors in the communication channel.

This solution has 7 different variations. The first 6 types are Interleave, Dispersed, Foreground with Leftover, Box-Out, Raster Scan and Wipe. They have definite macroblock allocation patterns. The last one is called Explicit type which is the most flexible kind among all of them. In this thesis, two explicit type algorithms are implemented. A new method is exploited during the implementation of the second algorithm. The main goal of this thesis is to compare the performance of these two algorithms with the first 6 definite patterns of the FMO in the wireless channel and wired networks platforms. Error correction coding is used in the second part of the simulation

The results are evaluated according to the PSNR values of the decoded videos. Besides that, their affects on the performance of the encoder and decoder are also examined.

1.3 Outline of the Dissertation

This thesis consists of 7 chapters. In the first four chapters a hierarchical structure is followed from H.264 AVC to FMO. The remaining chapters mention the simulation environment, results of the simulations and conclusion

In Chapter 2 H.264 standard is briefly explained. Firstly, its history is mentioned. Then two important parts of the standard which are Video Coding Layer (VCL) and Network Abstraction Layer (NAL) are described.

Chapter 3 summarizes the concept of Error Resilience Coding. The error resilience tools of the H.264 which are Intra Placement, Picture Segmentation, Reference Picture Selection, Data Partitioning, Redundant Slices, and Parameter Sets are briefly explained in this chapter.

In Chapter 4 Flexible Macroblock Ordering is described in detail. Different FMO types are explained. Then an FMO algorithm which is implemented in the thesis is described. Lastly, the proposed method is described

In Chapter 5 information is given about simulations. Firstly, the standards which are used for video communication are briefly explained. Then the characteristics of the wired and wireless communication platforms are mentioned and the models used for the simulation environment are described.

In Chapter 6 the properties of the coded video are stated. Then the results of the experiments are analyzed from different aspects. PSNR value is used for evaluated the objective quality. In addition to that, the amount of extra payload which the explicit algorithms cause is examined, and the success of algorithms according to the environments is explained utilizing the results. Finally, in Chapter 7 the results are summarized and future works are mentioned.

CHAPTER 2

H.264 VIDEO CODING STANDARD

2.1 Comparison of H.264 with Other Standards

H.264/AVC or MPEG 4 Part 10 [5] is a coding standard which was designed by JVT (Joint Video Team). The development of the standard was initiated by VCEG and it was firstly called as H.26L. MPEG and VCEG formed the JVT in 2001. VCEG called the standard as H.264 AVC and MPEG referred it as MPEG 4 Part 10. It was published in 2003. H.264 achieved 2 significant goals. First of all, it has a better compression rate than the previous codecs. According to the results of experiments, it is stated that it compresses the video at least 2 times better than other video codecs. This consequence can be accomplished by modifying the functional blocks of the codec. For instance prediction methods, transform methods and the entropy coding methods were improved. Secondly, this coding standard is especially developed for communication purposes. It had some features which made the coded video more robust to the errors in the communication medium. Flexible Macroblock Ordering is one of these features. Besides, it also had a layer for adapting it to the transmission protocol. This layer was called NAL (Network Abstraction Layer). These two properties made it very useful to use it in various applications. Some of them are streaming over different environments, storing videos, performing videophone and streaming multimedia.

2.2 General Features

Like all other video coding standards, H.264 [6] uses 3 statistical redundancies. The first one is spatial correlation which depends on the similarity among the neighboring macroblocks. The second one is temporal correlation which comes from the fact that the consecutive frames are alike. The third redundancy is the repetition of certain symbols in the bit stream. Compression can be achieved by getting use of these redundancies which will

be explained in the following subsections.

H.264 also provides an interface for communication. The former codec standard's aim is only compressing video. After this process, the next stage determines how to adapt the incoming data to the appropriate protocol. On the other hand, H.264 has a layer which adjusts the compressed video data to the appropriate communication protocol. The first layer which exploits different redundancy types to compress video is called Video Coding Layer (VCL). The second layer which provides an interface between source encoding and network is called Network Abstraction Layer. There are 4 H.264 profiles which are Baseline, Main, Extended and High.

2.2.1 VCL (Video Coding Layer)

VCL is the part of the H.264 standard where the compression is done. VCL performs the encoding process using a set of functional blocks. They are [6] Intra Prediction, Inter Prediction, Motion Estimation, Motion Compensation, Transform Coding, Quantization, Entropy Coding, Inverse Transform Coding, Dequantization and Deblocking Filter. Figure 2.4 illustrates these functional blocks. These blocks do not encode the picture as a whole. The building blocks of the frame and functional units of VCL are explained briefly in the following subsections.

2.2.1.1 Building Blocks of Frame

The frames are composed of 16×16 blocks which are called macroblocks and slices are group of macroblocks which are encoded and decoded independently from the macroblocks of the same frame [7]. However slices do not have a fixed size. They can be as small as one macroblock or they can be as big as one frame. There are 5 types of slices. They are Intra (I), Predictive (P), Bipredictive (B), Switching Intra (SI) and Switching Predictive (SP) slices. One of the important features of slices is that motion estimation and prediction can not be applied at the slice boundaries. Although this property decreases the compression efficiency, it increases the robustness by eliminating the dependency of slices to each other. In spite of increasing robustness, using too many slices for a frame also decreases efficiency because each slice has its own header which increases the payload. Each

frame can have no more than 7 slices. Figure 2.1 illustrates a frame which is composed of 3 slices.

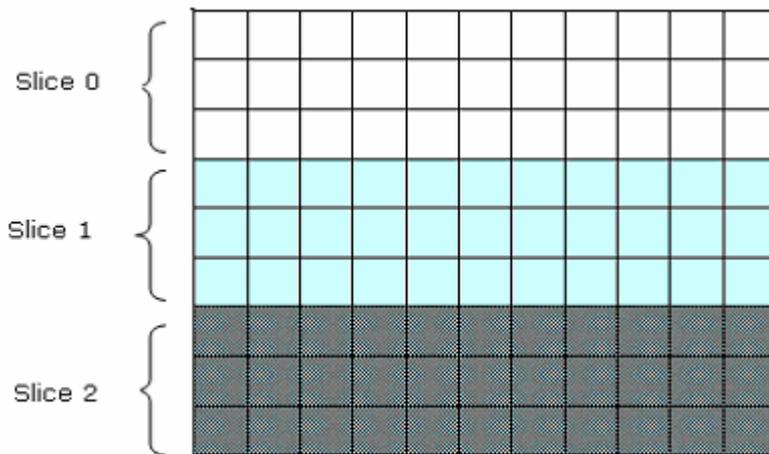


Figure 2.1: A QCIF frame with 3 slice

As mentioned before, slices composed of macroblocks, but the size of macroblocks can change according to color component. Because of the color perception characteristic of the human eye number of luminance pixels is more than chrominance pixels. Luminance macroblocks contain 16×16 pixels chrominance macroblocks consist of 8×8 blocks. Since there are 4 luminance blocks for each chrominance block the sampling format becomes 4:2:0. The structure of the components can be seen in Figure 2.2.

2.2.1.2 Main Functional Blocks of Video Coding Layer

Since the mission of VCL is very similar to the former video coding standards it also has similar functional blocks. The main difference between VCL and the

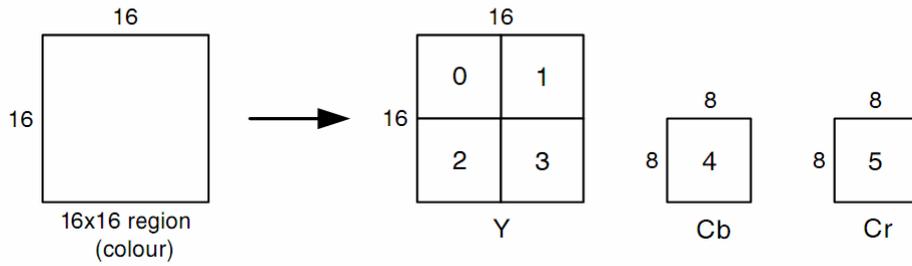


Figure 2.2: Sampling Format 4:2:0 [7]

former standards is that functional blocks of VCL are more complex than others. More detailed information can be found in [6]. In Figure 2.4, the structure of VCL can be seen. These several functional blocks can be called as follows: Intra Prediction, Inter Prediction, Transformation, Quantization and Entropy Coding, Inverse Transform, Dequantization and Deblocking Filter.

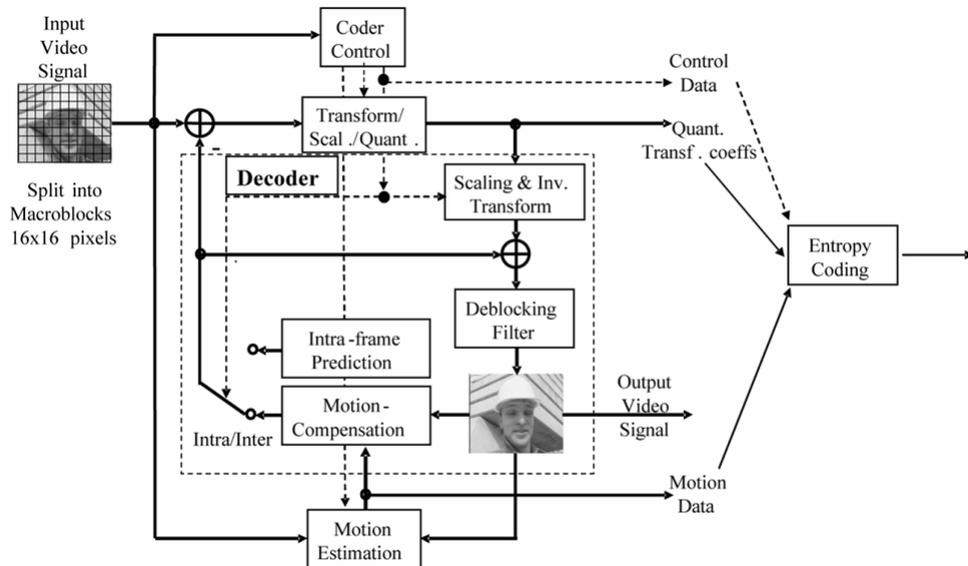


Figure 2.3: Structure of VCL [6]

2.2.1.2.1 Intra Frame Prediction

This functional block basically uses the spatial correlation among neighboring macroblocks to form intra slices. Since pixel values of the neighboring macroblocks have similar intensity values, instead of sending all the pixel values of them to the transformation stage, reference adjacent macroblocks which are located left or above of the others, are chosen and the remaining ones are predicted from these reference macroblocks [6]. Frames whose macroblocks are predicted using intra prediction can be used to resynchronize the video if drift occurs. Furthermore, intra prediction stops error propagation since intra predicted macroblocks only need the macroblocks which are located in the same frame. Although Intra prediction has such good features and it provides a better PSNR value than inter prediction, it is not frequently used during real time communication because size of intra predicted macroblocks is bigger than the size of inter predicted macroblocks. Hence intra prediction may not support the bandwidth constraints. Moreover the coded frame has to be fragmented before transmission which is an undesired case and the error probability of losing the packets, which contain these macroblocks, increase. As a result, these macroblocks have to be sent in a periodical manner.

There are 4 intra prediction methods which can be used during encoding. Two of them can be applied to luma components, one of them is used to predict chroma macroblocks and I_PCM is used to skip prediction and transform coding stages. Two Luminance prediction modes are 16x16 intra prediction and 4x4 intra prediction [6]. Intra 16x16 and intra 4x4 mode predictions are performed using the neighboring macroblocks and blocks respectively. 9 modes can be applied while predicting 4x4 macroblocks. Five of them are Horizontal, Vertical, Mean, Diagonal Down /Left and Diagonal Down/Right which can be seen in Figure 2.4. There are four prediction methods for 16x16 blocks which are Horizontal, Vertical, Mean (DC) and Planar. If the spatial correlation is high, 16x16 macroblocks can be used; otherwise, the area contains important detail, and 4x4 intra prediction mode should be used. [6]

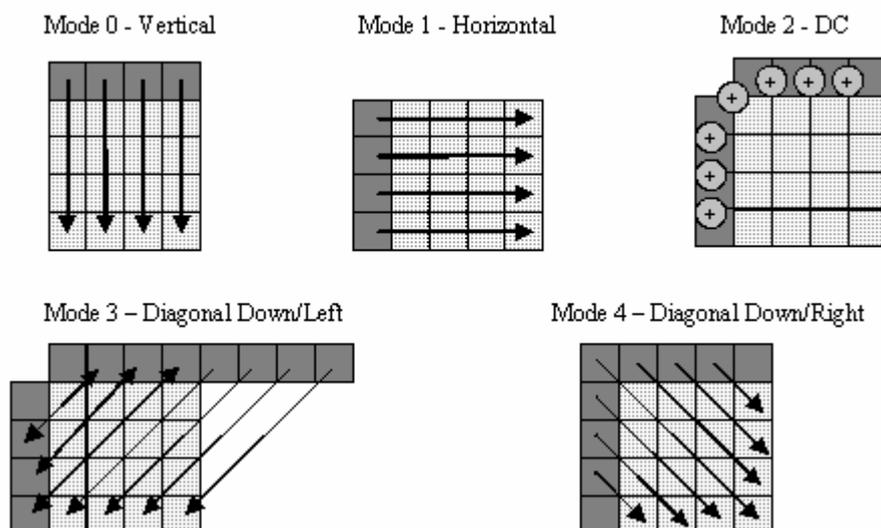


Figure 2.4: Five Intra_4x4 prediction modes [6]

The pixel values of the predicted macroblock are calculated using a weighted combination of neighboring macroblock pixels. Chroma intra prediction is employed using a similar method to Intra_16x16 prediction of luminance component.

2.2.1.2.2 Inter Frame Prediction

Like spatial correlation among neighbor macroblocks of the frames, there is also a correlation among the consecutive frames of the videos. Inter frame prediction is used to exploit this redundancy. It consists of three steps. The first step is motion estimation. In this step, the movement of objects is found. The second step is motion compensation. The difference between the actual frame and the estimated one is calculated at this step. The last step is inter prediction of motion vector blocks. Motion vectors are predicted from adjacent motion vectors. Predictive (P) and Bipredictive (B) macroblock types are created using motion compensated prediction.

Motion estimation is the most complex part of the encoder. The duration of motion estimation changes according to the applied algorithm. Like most of the codec standards, H.264 AVC also uses block based motion estimation [6]. Although this method may not find the best motion vectors, it is one of the

fastest algorithms which can be used for motion estimation. Despite using the same motion estimation method, H.264's resolution for detecting motion is better than former standards. Luminance and chrominance components can be estimated up to quarter pixel accuracy [6]. In order to make the motion prediction using half pixel accuracy, 6 tap FIR filter is used to double the resolution. Quarter pixel components are obtained by bilinear interpolation of the half pixel components. Half pixel and quarter pixel values of chroma components are obtained using linear interpolation. In addition to the enhanced accuracy, H.264 can also use multiple reference frames [6]. 4 frames can be used as reference frames. This feature can provide better compression efficiency especially on periodically changing scenes and it also increases error resilience by distributing reference dependency among different frames. However, it also has some disadvantages. In order to decode the actual frame, 4 reference frames have to be used which can increase the required memory. Besides that, finding the best motion vectors among the combination of frames increase the complexity of the encoder substantially.

Motion estimation can be carried out using blocks which have different size. The size is firstly determined according to the type of block. The block can be a partition of a macroblock which has 16x16 pixels or it can be a partition of submacroblock which consists of 8x8 pixels [7]. 4 different types of block size can be used for prediction which are 16x16, 16x8, 8x16, 8x8). If 8x8 block size is selected, there are also 4 different types of block size can be used for submacroblock partitioning which are 8x8, 8x4, 4x8, 4x4. They are illustrated in Figure 2.5. Hence the number of motion vectors which can be sent for one macroblock can change from 1 to 16.

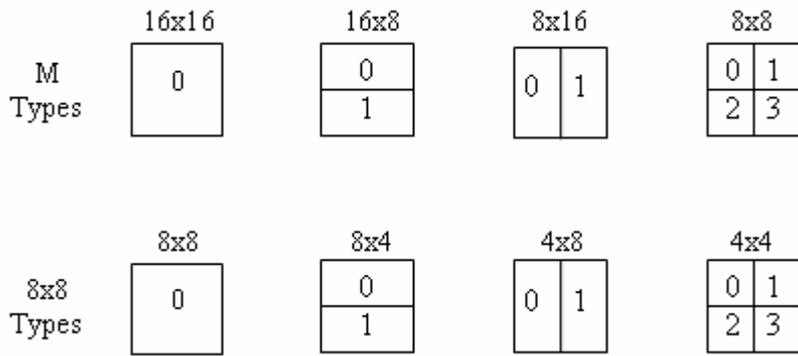


Figure 2.5: Segmentation of macroblocks for motion compensation [6]

This property increases the efficiency of the encoder substantially because the size of the block which is used for estimation can be changed according to the motion characteristic of this block. If the motion complexity of the block is low then macroblock partitioning can be used to encode faster. On the other hand, if the motion complexity of the area is high, one of the submacroblock partitions can be used to increase the accuracy of estimation. In addition to the temporal correlation, spatial correlation of motion vectors is also used during inter prediction. Instead of sending the actual motion vector values, predicted values are sent.

In order to perform inter prediction the neighboring blocks are used. The motion vectors are calculated differentially from the other motion vectors. Two methods are used for calculating the estimated motion vectors. They are directional segmentation and median prediction. Directional segmentation is used for 8x16 or 16x8 blocks [7]. If the macroblock consists of two 16x8 submacroblocks, the motion vector of submacroblock, which is placed on the top, is predicted using the macroblock above. Motion vector of the other submacroblock is predicted using the macroblock on the left. On the other hand, if the macroblock consists of submacroblocks, which contain 8x16 pixels, then the motion vector of submacroblock on the left is predicted using the macroblock on the left and the partition on the right is predicted using the neighboring macroblock on the right. Median prediction is performed using all the available motion vectors of neighboring macroblocks. Median of these motion vectors is used for prediction.

2.2.1.2.3 Transform Coding

This process is executed after intra or inter prediction stages. After prediction is performed, the output is sent to the transform coding block. This output is nothing but the residual after subtracting the predicted block from the reference block. Since the predicted block is similar to the reference block, there is a high correlation among the pixels of residual block. Hence, this information can be represented using fewer bits if it is transformed into another domain.

2-D Discrete Cosine Transform (DCT) was used to be applied for this aim. This process is performed using 8x8 blocks. In H.264, Integer Transform [6] is used instead of 2D DCT. Indeed, Integer transform is nothing but a simplified version of 2D DCT. The 4x4 transform matrix of Inverse Transform can be seen in Figure 2.6

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}.$$

Figure 2.6: Integer Transform Coding Matrix [6]

Integer Transform uses simple arithmetic operations which are additions and shifts. Furthermore 16 bit is enough to perform all the operations. These two features make integer transform an easily applied and a fast method to carry out transformation. Moreover, in inverse integer transform, mismatch does not occur because operations like division and multiplication which cause rounding error are not used during transformation. Besides that, integer transform is applied to 4x4 blocks instead of 8x8 blocks which decreases noise around edges. The DC coefficients of the chrominance components are also transformed using 2x2 Hadamard Transform [6] in addition to Integer Transform.

2.2.1.2.4 Quantization

In order to increase the compression efficiency, quantization is applied to the transformed coefficients. Unlike the other functional blocks, this process causes lossy compression because the coefficients are adjusted according to some predefined levels and distortion occurs as a result of this adjustment. H.264 standard employs Scalar quantization [6]. Since low frequency components are more important than high frequency components they are quantized with more levels. As the frequency of the components increase the step size of the quantization increases. The value of transformed coefficients is getting closer to zero as the frequency increases. Hence these values become zero after a predefined threshold. There are 52 quantization levels in H.264 from 0 to 51. 1 step increase in quantization value means %12 increase in quantization step size [6]. After quantization, the coefficients are reordered according to zigzag scan pattern in order to increase the efficiency of entropy coding. Only the 2x2 DC chroma components are ordered using raster scanning.

2.2.1.2.5 Entropy Coding

This functional block is the last stage of VCL. The aim of this block is to exploit the statistical redundancy of the transformed and quantized coefficients. Since this coding is lossless, the input stream can be reconstructed without any distortion. Two methods [6] which are Context Adaptive Variable Length Coding (CAVLC) and Context Adaptive Binary Adaptive Coding (CABAC) can be used in H.264 to carry out this procedure. CAVLC is mainly based on VLC. VLC represents frequently repeated codeword with shorter codeword and less frequent symbols with longer codewords. VLC uses codeword tables in order to match the incoming symbol. Hence the number of bits in the stream reduces. Using CAVLC provides switching between different tables according to the previously transmitted elements. On the other hand, CABAC [7] is an arithmetic coding as its name implies. However this arithmetic coding uses finite elements to decrease the complexity. These finite elements form the binary alphabet. CABAC consists of three steps. The first step is binarization which converts symbols that are not binary to binary. There are several methods to carry out binarization. Second part is determining the context model according to bit positions of

symbols. This context model is used to select the appropriate binary arithmetic coding parameters. The last step is updating the symbol probability parameters adaptively according to the incoming symbols.

2.2.2 Network Abstraction Layer (NAL)

Network Abstraction Layer is the second layer of H.264 standard. There is not a similar layer in the former codec standards. The main aim of this layer is adjusting the output of VCL to different communication or storage protocols. Some of the possible layers which are supported by NAL are RTP/IP, ISO MP4 and H.32x. The basic unit which carries the coded video data is called NAL unit (NALU) [4]. NAL units consist of two parts. The first part is header which is 1 byte long. Figure 2.7 demonstrates the structure of a NALU header. The second part is payload and its size changes according to the type of the information. First 5 bits of NAL header carries the type (T) of information carried in the payload. First 12 types are currently known. The next 2 bits are called nal reference idc [4], and they show the importance of the payload. If the value is zero then this NALU does not contain important payload like reference blocks. However, if the number is bigger than 0 then this NALU includes significant data. The last bit is called forbidden (F) bit and it illustrates the validity of the unit. NALU's can be transmitted either in bit stream format or in packet format [4].

0	1	2	3	4	5	6	7
T					R		F

Figure 2.7: Header of a NAL Unit [4]

NAL units which are transported in byte stream format are separated from other NAL units using a unique three byte pattern which is called start code prefix. The start code prefix bytes are 00 00 01. Since this pattern should only be seen at the start of the units, the byte sequences, which have the same pattern, has to be altered using emulation prevention bytes. On the

other hand, when the NAL units are transmitted using packet oriented format, there is no need to use start code prefix. Most of the packet oriented transmission protocols have a layered structure which adds header information that separate NALUs from each other.

CHAPTER 3

ERROR RESILIENT VIDEO CODING IN H.264

3.1 Overview

Error Resilient Video Coding can be employed in three ways [2]. It can be used in the encoder to create more robust stream before transmission. It can be also employed in the decoder to decrease the effects of errors. Lastly it can be used in the combination of source coder and decoder to select the reference frames.

3.1.1 Encoder Error Resilient Methods

H.264 can employ 6 error resilience tool in the encoder. They are Intra Placement, Picture Segmentation, Data Partitioning, Parameter Sets, Redundant Slices and Flexible Macroblock Ordering. The last three methods are only found in H.264 coding standard [4]. Reference Picture Selection is a member of both Encoder Error Resilient Methods and Encoder-Decoder Error Resilient Methods. Hence, it is explained in subsection 3.1.3. Moreover, Parameter Sets are not actually used as an error resilience tool by themselves. However, they can be required while using other error resilience tools. For instance, macroblock allocation map is used for decoding the frames which are encoded using FMO. This information is sent in Picture Parameter Set (PPS) which carry the coding parameters of a frame [4]. Remaining methods are explained in the following subsections.

3.1.1.1 Intra Placement

Since temporal prediction is applied during inter coding there is a dependency among macroblocks of consecutive frames. Thus, if an error occurred during the transmission of reference macroblock, the predicted macroblock is not

accurately decoded. Sending intra coded frames in regular intervals prevents drifting because macroblocks of intraframe is only dependent to itself. Therefore the errors seen in the previous frames do not have a negative effect on intra coded frame. The number of optimum intra coded frame can be determined according to the characteristics of medium which can be found using physical interface and communication protocol features. Detailed information about deciding the number of frames adaptively is found in [8].

3.1.1.2 Picture Segmentation

Since the size of the encoded frame is bigger than MTU in some cases, it has to be distributed into more than one packet. However if one of the packets is lost the remaining packets become useless. H.264 uses slices [4] in order to solve this problem. As mentioned in Chapter 2, slices are self-dependent frame partitions. Hence, they can be decoded without using other slices. Macroblocks are allocated into the slices in raster scan order.

3.1.1.3 Data Partitioning

Data Partitioning [4] is a method for separating the macroblocks of slices into different units which are called partitions. Although slices are very beneficial groups for creating independently decodable parts, they can be protected more efficiently if data partitioning is employed. These partitions are formed using the macroblocks which have common semantic features. Three types of partitions are created using this approach. They are called Type A Partition, Type B Partition and Type C Partition. Type A Partition carries the header information which contains MB types, quantization parameters and motion vectors. This is the most crucial partition because the other partitions are useless if this partition is not received correctly or lost. The Intra coefficients and Intra CBP's are transmitted in Type B partition which is also called Intra Partition. The information carried in this partition is crucial because it can provide resynchronization. Type C partition carries the inter coefficients and inter CBP's. If these partitions are ordered according to their importance Type A is the most important one because Type B and C can not be decoded without the information in Type A. Type B is more important then Type C because it can stop error propagation. Since different partitions have different

priorities UEP can be used to obtain better results at the decoder. Data Partitioning is performed in the source coder.

3.1.1.4 Redundant Slices (RSs)

Redundant Slice [4] is another representation of a coded slice which is also transmitted to the receiver. Although both of the coded slices are created using the same partition of the frame, they are different from each other since they are encoded using different parameters. If the decoder receives both of the slices, it ignores the redundant slice and uses the actual one. On the other hand, if only the packet which carries the redundant slice is received then this slice is used for decoding.

3.1.1.5 Flexible Macroblock Ordering (FMO)

During a regular encoding procedure, macroblocks are allocated using a raster scan pattern. FMO is a method which is developed for flexible macroblock allocation. Hence, appropriate allocation pattern is selected according to error characteristics of channel.

There are 7 different FMO types which are Interleave, Dispersed, Foreground with Leftover, Box-Out, Raster Scan, Wipe and Explicit types [4]. Detailed information about FMO is given in the next chapter.

3.1.2 Decoder Error Resilient Methods

These methods are also known as decoder error concealment methods. They are used for predicting corrupted sections of the frames by exploiting the information in the other sections of the same frame or the other frames.

There are two popular methods for applying error concealment in the decoder. They are Weighted Pixel Value Averaging and Boundary Matching Based Motion Vector Recovery [9].

3.1.2.1 Weighted Pixel Value Averaging

This method is employed for intra pictures. The pixels of lost macroblocks are estimated by extrapolating the boundary pixel values of the adjacent macroblocks. The weights are inversely proportional to the distance between the pixel of the lost macroblock and the boundary pixels of adjacent macroblocks. Detailed information can be found in [10].

3.1.2.2 Boundary Matching Based Motion Vector Recovery

This approach is used for inter macroblocks [4]. It consists of two steps. Firstly the length of correctly received motion vectors is compared with a threshold value. This comparison is done in order to decide whether using current frame or reference frame for prediction. If the length is smaller the macroblocks of the reference frame which are at the same positions are employed. Otherwise, motion compensated error concealment is performed using a column wise approach from outside to the center of the image. Concealment is performed for each 8x8 block. The adjacent blocks and the block which is at the same position of the reference frame can be used as candidate frames to conceal the lost block. The boundary matching errors of the candidate blocks are compared. The candidate, which has the smallest error, is used in the place of lost block. Detailed information can be found in [10].

3.1.3 Encoder and Decoder Interactive Error Resilient Methods

In this approach the source coder and decoder work together to decrease the effect of errors. There is a feedback channel between decoder and encoder which is used for sending messages from decoder to encoder to state which packets are lost. Then encoder decides what to do according to this information. Reference Picture Selection, which is an error resilience tool of H.264, is a member of this group.

3.1.3.1 Reference Picture Selection

Reference Picture Selection can be used in both systems which include feedback or not [4]. If there is feedback channel it decides the reference frame which is used for predicting the current frame according to the feedback information. Hence error propagation can be prevented.

CHAPTER 4

FLEXIBLE MACROBLOCK ORDERING

4.1 Introduction

Like parameter sets and redundant slices, FMO is a novel tool of H.264 [4]. It is developed in order to increase the robustness of encoded video data in highly error prone environments by allocating macroblocks in a different manner than the normal allocation. During a regular encoding procedure macroblocks are allocated to the slices in raster scan order which can be seen in Figure 4.1.

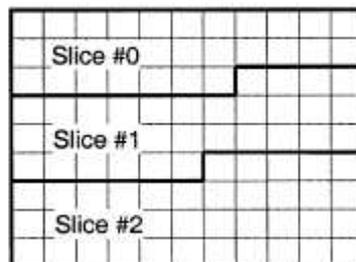


Figure 4.1: Slice structure without using FMO [6]

This allocation increases the coding efficiency because most of the neighboring macroblocks are carried in the same slice. Therefore, intra picture prediction can be performed efficiently. However, when one of the slices is lost, the macroblocks, which belong to that slice, cannot be

estimated successfully because adjacent macroblocks are also lost. FMO is a tool that provides flexible allocation of macroblocks to slices. It is firstly proposed by Wenger and Horowitz in the 3rd meeting of JVT in 2002 [11]. Moreover similar methods were also tried using former codec standards. For instance, it is used as Scattered Slices method in H.26L. More details can be found in [12]. It is developed for baseline profile which supports basic real time video communication applications.

FMO decreases the coding efficiency because prediction can not be performed on the slice boundaries. On the other hand, it does not affect the motion compensation and deblocking filter procedures. The first two patterns, which were used as FMO pattern, are Scattered Slices and Line Interleaving. Their test results are explained more elaborately in [13] and [14]. Although FMO increases necessary bandwidth because of coding efficiency and MBAmapping information, it also decreases bitrate by omitting slice data from macroblocks. Since FMO separates the macroblocks to the slices in a different order than the previously used pattern, this ordering has to be known both by encoder and decoder. Therefore it is kept in a structure which is called macroblock to slice allocation map (MBAmapping) [11]. MBAmapping is carried to the decoder in PPS. Each macroblock of the MBAmapping is a member of a slice group. The macroblocks are represented using slice group ID (SGid) in the MBAmapping. Because there can be no more than 8 slices (0-7), 3 bit is enough to represent a SGid. The decoding procedure for a slice, which is encoded using FMO, is as follows. Firstly the SGid of the first macroblock of the slice is determined. Then the macroblocks of the slice are decoded according to order of their indexes. Using FMO can cause coding inefficiency and delay but their amount can be adjusted by applying an appropriate MBAmapping. One of the important differences between regular scanning and FMO is determining the end of the frame. The end of the frame is easily found if raster scanning is applied because the macroblock which has the highest index number is decoded lastly. On the other hand the last macroblock do not have to be the highest index number of the frame for some FMO types. The end of the frame can be determined by finding the macroblock which has the highest index of the last slice group.

4.2 Types of Flexible Macroblock Ordering

There are 7 types of Flexible Macroblock Ordering. They are [7] Interleave, Dispersed, Foreground with Leftover, Box-Out, Raster Scan Wipe and Explicit Types. The first 6 methods have a definite MBAmapping pattern; however the explicit type does not have a constant pattern. Moreover, Box-Out, Raster Scan and Wipe Types are developed for Isolated Regions.

4.2.1 Interleave Type

This method was firstly developed in 1998 at British Columbia University. It is also known as Macroblock Line Interleaving [13]. This method allocates the macroblocks according to the following procedure. Firstly number of slices and their runlengths are determined. Then allocation is started from the first macroblock which is located at top right of the frame. Assigning slice group ID (SGid) follows a raster scan pattern. When the number of macroblocks is equal to the runlength of the slice, the following macroblocks are assigned to next slice. The allocation procedure goes on until all the macroblocks are distributed to slices. Since sum of the runlengths can be smaller than total macroblock number of the frame, a slice group can be used to represent more than one region. The MBAmapping structure of Interleave Type for 3 slices is illustrated in Figure 4.2. Runlength of each slice is 11.

0
1
2
0
1
2
0
1
2

Figure 4.2: Interleave FMO Type (3 Slices Groups)[7]

Since each line is transmitted in different packets, if one of the packets is lost then the macroblocks of the corresponding line can be estimated using the lines which are located below and above. Furthermore intra prediction can also be performed for slice which is coded using this type because the macroblocks are allocated such that they can be predicted either using the adjacent macroblocks.

The SGid's of macroblocks are found according to the following equation

$$M[i][j] = k \quad \text{where} \quad \sum_{n=0}^{n=k-1} R_n < \text{mod} \left(\text{ind}, \sum_{n=0}^{n=N-1} R_n \right) \leq \sum_{n=0}^{n=k} R_n \quad (4.1)$$

In this equation, M denotes the 2D MBAMap structure, i and j denote the row and column numbers MBAMap respectively and N represents the number of slice groups. R_0, R_1, \dots, R_{N-1} represent the runlength number of each slice group.

4.2.2 Dispersed Type

This type is firstly tried in H.26L as Scattered Slices [12]. However it is not added in H.26L standard. Since this method gave successful results in these trials it is prepared to use in H.264 standard. The main aim is distributing macroblocks to the slices such that no 2 adjacent macroblocks are placed in the same slice. The checkerboard pattern is a special kind of Dispersed FMO where number of slice is 2. Figure 4.3 demonstrates the structure of MBAMap of Dispersed Type when slice number is 4. Among different FMO types Dispersed Type is the most appropriate one for error resilience since all the macroblocks can be used if one of the slices is lost. On the other hand because slice boundaries are macroblock boundaries in picture prediction can not be used which decreases the coding efficiency substantially.

0	1	2	3	0	1	2	3	0	1	2
2	3	0	1	2	3	0	1	2	3	0
0	1	2	3	0	1	2	3	0	1	2
2	3	0	1	2	3	0	1	2	3	0
0	1	2	3	0	1	2	3	0	1	2
2	3	0	1	2	3	0	1	2	3	0
0	1	2	3	0	1	2	3	0	1	2
2	3	0	1	2	3	0	1	2	3	0
0	1	2	3	0	1	2	3	0	1	2

Figure 4.3: Dispersed FMO Type (4 Slice Groups) [7]

Equation (4.2) is used for creating Dispersed MBAmapping pattern

$$M[i][j] = \text{mod}(j + (i * N / 2), N) \quad (4.2)$$

In the formula, M denotes the 2D MBAmapping structure, i and j denote the row and column numbers MBAmapping respectively and N represents the number of slice groups.

4.2.3 Foreground with Leftover Type

This type is used for allocating macroblocks such that some regions of the frame that are determined according to a ROI algorithm and the remaining part of the frame are put into a different slice [13]. This method can be used to perform Unequal Error Protection (UEP) like in the Scalable Video Coding (SVC) case. Figure 4.4 illustrates a typical pattern of Foreground with Leftover Type. If this method is evaluated from the coding efficiency point of view, it can be observed that it is better than both Interleave and dispersed types since in-picture prediction can be performed using more than 2 neighborhood macroblocks.

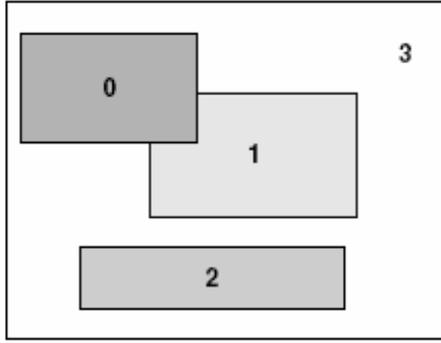


Figure 4.4: Foreground with Leftover FMO Type (4 Slice Groups) [7]

On the other hand this method is not a very efficient error resilience tool because the macroblocks of a slice are mostly adjacent blocks.

Allocation of macroblocks is performed according to the following formula

$$M[i][j] = \begin{cases} p, & MB[i][j] \in A_p \text{ and } 0 \leq p \leq k \\ k+1 & \end{cases} \quad (4.3)$$

In the equation above, There are k regions where $k \leq 6$. A_1, A_2, \dots, A_k represent the interested rectangular regions which do not overlap. These regions are determined by their top left and bottom right coordinates. $MB[i][j]$ denotes the macroblock that is located i^{th} row and j^{th} column of MBAmapping. M denotes the 2D MBAmapping structure, i and j denote the row and column numbers MBAmapping respectively.

4.2.4 Box-Out Type

Box-Out Type is one of the algorithms which are based on shape evolution [16]. Unlike the other types where first macroblock of the first slice is the one which has the lowest index number, it is the one that is located at the center of the frame. Then a spirally scanning pattern is used to choose the next macroblock. The scanning can be performed in the clockwise or counter clockwise direction. After coding a predefined number of macroblocks which determine the size of the packet, the next slice is formed following the same

pattern. This process goes on until the last macroblock of the frame. The following figure illustrates the pattern of Box-Out type when the number of slice groups is 2.

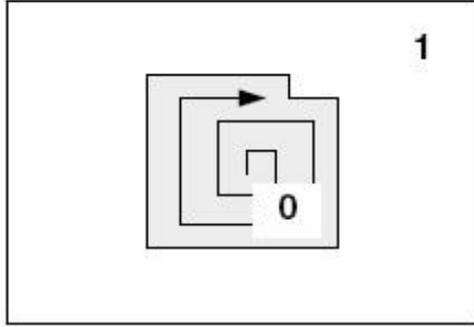


Figure 4.5: Box-Out FMO Type (2 Slice Groups) [7]

This type can also be used for UEP because the macroblocks which are close to the center are generally more important than the others. Although in-picture prediction is limited because of the pattern structure, this type is more appropriate for applying error concealment than Foreground with Leftover type. When the number of encoded macroblocks increase from center to the outside the circumference of the scanning pattern increases. Hence slices start to have more adjacent macroblocks from other slices.

In order to find the pattern structure firstly the macroblock order has to be calculated. The pattern is determined according to the following equations.

$$\mathbf{MB}_N[i][j] = \begin{cases} \mathbf{MB}_C[i][j-1], & (2r+1)^2 \leq ind \leq (2r+1)^2 + 2r - 1 \\ \mathbf{MB}_C[i+1][j], & (2r+1)^2 + 2r \leq ind \leq (2r+1)^2 + 4r - 1 \\ \mathbf{MB}_C[i][j+1], & (2r+1)^2 + 4r \leq ind \leq (2r+1)^2 + 6r - 1 \\ \mathbf{MB}_C[i-1][j], & (2r+1)^2 + 6r \leq ind \leq (2r+1)^2 + 8r \end{cases} \quad (4.4)$$

$$M[i][j] = ind_{MB[i][j]} * N / T \quad (4.5)$$

In these equations the first macroblock is located at the center which can be denoted by $MB[I_c][J_c]$. $MB_C[i][j]$ and $MB_N[i][j]$ represent the current and the next macroblocks respectively. r denotes the pixel distance of circumference from the center. Each square spiral starts with $MB[I_c - r][J_c + r - 1]$. T denotes the total number of macroblocks. M denotes the 2D MBAmapping structure, i and j denote the row and column numbers of MBAmapping respectively and N represents the number of slice groups. Lastly, ind represents the order of macroblock. Figure 4.6 illustrates Box-Out pattern structure

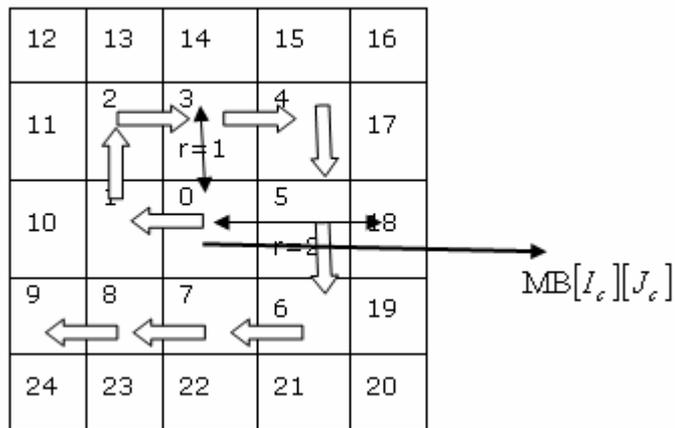


Figure 4.6: Box-Out Coding Sequence

4.2.5 Raster Scan Type

Actually this is the default type of scanning if no FMO is applied. This is the second shape evolution method where pattern progress is carried out in the

horizontal direction. [16]. Scanning starts from the macroblock which is placed at the top left of the frame. The second macroblock of the slice is located at the second column of the first row. Allocating macroblocks goes on until the end of the row. Then the first macroblock of the next line is encoded. This process continues until the end of the slice which is determined by maximum number of bytes or maximum number of macroblocks that a slice can have. The MBAmap for 2 slice is illustrated in Figure 4.7 Raster Scanning can also be performed in the reverse direction where scanning starts from the macroblock which is located at the bottom right of the frame. This method has a very good coding efficiency since in-picture prediction can be applied easily. On the other hand it is not a very appropriate error resilience tool neighboring macroblocks are mostly in the same slice. Therefore if one of the slices is lost during transmission, it is hard to predict the lost macroblocks spatially.

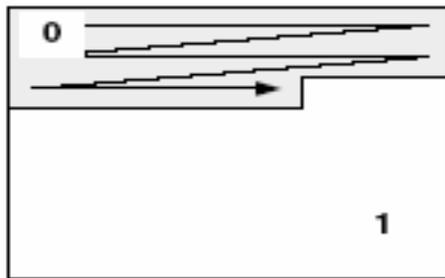


Figure 4.7: Raster Scan FMO Type (2 Slice Groups) [7]

The allocation is carried out using equation (4.6)

$$M[i][j] = (j + ((i-1) * W)) * N / T \quad (4.6)$$

In equation (4.6), M denotes the 2D MBAmap structure, i and j denote the row and column numbers MBAmap respectively and N represents the the number of slice groups. W denotes the number of macroblocks in a row.

4.2.6 Wipe Type

The fifth FMO type is Wipe Type which is also a space evolution type. This type is very similar to the Raster Scan Type. The main difference is scanning direction. Raster scanning performs row by row scanning. On the other hand wipe type performs column by column scanning [16]. Figure 4.8 illustrates the MBAMap pattern of Wipe Right FMO which consists of 2 slice groups. Because of these differences videos encoded using these types have different bitrate values for same quantization parameter. There is also a Wipe Left pattern where the first macroblock is located at the bottom right position.

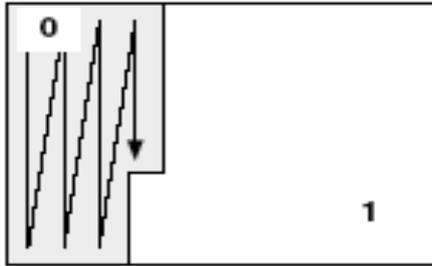


Figure 4.8: Wipe FMO Type (2 slices)

Error resilience and coding efficiency feature of Raster Scan are also valid for Wipe type. The MBAMap pattern can be found using the following equation

$$M[i][j] = (i + ((j-1) * H)) * N / T \quad (4.7)$$

In this equation, M denotes the 2D MBAMap structure, i and j denote the row and column numbers MBAMap respectively and N represents the number of slice groups. H represents the number of macroblocks in a column.

4.2.7 Explicit Type

Explicit type can be seen as the most flexible one among others. As mentioned before, unlike other FMO types there is not a known pattern for explicit type. Hence, the MBAMap of this type has to be known before starting encoding. Moreover MBAMap of Explicit Type can change dynamically according to the algorithm which determines the allocation of macroblocks. Since there is not a standard, most of the studies about FMO are carried out using explicit type. The aim of these studies is to find the most appropriate approach for determining the MBAMap. Some of these studies are explained briefly in the following chapter.

4.3 Related Work on Explicit FMO

They are divided into three groups. The first group tries to find methods which detect important macroblocks and apply unequal error protection to transmit these important macroblocks more reliably. These works can be seen as Layered Coding algorithms. The second group uses FMO for increasing the efficiency of practical applications. The last group deals with developing new patterns for FMO.

The study which is performed by Benierbah and Khamadja [15] explains a technique Block Ordering Scalability. The importance of the blocks is determined according to the motion characteristics of them. The macroblocks which have high motion complexity are more important than others. These blocks are collected in a region which is called New Coded Picture (NCP). The other region, which contains the remaining macroblocks, is called discarded region (DR). Indeed, NCP is coded as base layer of SVC and DRs compose the enhancement layer. Another approach is developed by Im and Pearmain [16] in 2006. The authors suggest a rate distortion based method to obtain the maximum resilience while keeping bitrate smaller than a predefined level. This aim is accomplished by determining the macroblocks which increase the objective quality most. According to [16] the macroblocks are determined by their sole effects to the total PSNR and their contribution to the error concealment. Furthermore, since most of the consecutive frames are similar PPS is not always sent for every frame. The third approach [17] is a rate distortion based approach like the previous one. However, the algorithm chooses whether using FMO or not during encoding procedure instead of

determining the importance of macroblocks. FMO has significant advantages like providing more robust streams. On the other hand, it also has a very serious disadvantage which is increasing bitrate. Wu and Boyce suggest a method to code a frame either using FMO or not based distortion and rate constraints. Although this method gives better PSNR results than regular encoding and transmission scenario encoding complexity increases substantially because this process has to be performed for all the frames of a GOP. Moreover buffer requirements at the decoder side may cause a memory problem.

The fourth method is another rate distortion approach which is proposed by Dhondt et al [18]. This algorithm like second algorithm is based on choosing crucial macroblocks. However in this case these critical macroblocks are determined according to the importance of their pixel values. Moreover importance is not only dependent to current frame but also succeeding frames. The importance of a pixel is calculated by multiplying the distortion caused by the pixel with how many times it is repeated. The distortion is calculated by subtracting the concealed value of the pixel from the actual value. Therefore the type of error concealment is also an important parameter. Since this procedure is repeated for each pixel in the frame and the number of repetitions is found by searching all the frames in the GOP, this algorithm increases the encoder complexity substantially. Furthermore the amount of delay also goes up. The fifth study [19] is carried out by Thomos et al. This work consists of two parts which are implemented iteratively. The first part is grouping macroblocks adaptively. The second part is applying FEC based on a rate constraint. The aim is to create the optimum source and channel coded stream using FMO. Baccichet et al [20] proposed a method to adaptively detect the important regions. Then the frame is coded twice. The first one is encoded such that the important region is quantized with a better parameter and the remaining part is quantized with a coarser parameter. The second coding procedure is carried out using an offset quantization parameter. The aim is to adjust the rate of redundant and primary descriptions such that the distortion is minimized. The next study [21] is performed using Discrete Wavelet Transform to obtain the slices. Since wavelet transform is used to create slices, they are not equally important. Although this method does not give the best objective quality, its psychovisual features are impressive. The eighth study of this group is done by

Wang, Li, Schimiza, Ikenaga and Goto [22]. The goal of this work is developing a method to handle two disadvantages of FMO which are bitrate overhead and misuse of UEP. According to the authors these two disadvantages can be eliminated by grouping macroblocks based on prediction mode and size information. The grouping is performed as follows: All the intra macroblocks and small sized inter blocks are collected in important slice group and the remaining macroblocks are collected in less significant one. Then LDPC coding is applied according to the importance of slices. The last study in this group is performed by Araichi et al [23]. They use two FMO types and UEP to provide more reliable transmission and efficient decoding for a selected region. The algorithm is composed of three steps. The first step is determining the region which will be protected more. This region is found according to the motion complexity of the macroblocks and set as foreground. The remaining part is selected as background of the frame. After determining the regions the foreground is separated into partitions using a dispersed pattern. In the last step UEP is applied.

In addition to the efficiency of FMO as an error resilience tool it can also be used for other purposes. In the work [24] of Tan et al FMO is used during transcoding to change the allocation of macroblocks to the packets. FMO is also used for ROI scalable coding with XML based ROI extraction method [25]. The region of interest is determined using FMO Type 2 pattern. Lastly FMO is used for Multiple Description Coding purpose in the study of Parameswaran et al [26].

The last group consists of new FMO patterns. Two studies are performed for this aim. The first one is performed by Dhondt et al [27]. In their study, they firstly define the features mapping of best scatter pattern. First of all each slice contains approximately same number of macroblocks. Secondly adjacent macroblocks should be put into different slices. Thirdly neighboring macroblocks should be members of as many slice group as possible. Lastly the SGid's are assigned to macroblocks such that the macroblocks in the same slice are located as far as possible. Based on these requirements authors suggest a novel scattering pattern which is at most 1.5 dB better than others. The second pattern which is found by Ogunfunmi and Huang has 3D MBAMap instead of 2D MBAMap [28]. According to the algorithm of this mapping, the macroblocks of the frame are allocated into the slices such that no adjacent macroblocks in the same frame and no two macroblocks which

are at the same index of the consecutive frames are allocated into the same slice. In order to perform this ordering procedure encoding and decoding processes has to be done using three consecutive frames which can increase delay and buffer requirements.

In [29], the important macroblocks are equally distributed to the slices. Hence random errors can not damage the slices which carry all the important macroblocks. This work is selected because it has low computational complexity with respect to others which is very important for real time communication. It will be explained elaborately in the next subsection.

4.4 Explicit Type 1 Algorithm

The algorithm is developed by Hanatanong and Aramwith [29]. The method is based on allocating important macroblocks to the slices equally. Hence, slices have similar importance. In order to achieve this, macroblocks are distributed to slices according to their bit count. Macroblock bit count information illustrates whether this macroblock is predicted easily. The macroblocks which are predicted easily are similar to the reference blocks. Therefore if the predicted block is lost, the error can be concealed successfully using the reference block. On the other hand macroblocks that have higher bit count generally can not be predicted well. As a result, if an error occurs during transmission, using reference block to conceal this error can cause drifting and this effect propagates to the succeeding frames. In order to decrease the damage, the hardly predicted blocks are put into different slices. This approach consists of three steps. In the first step, the bit numbers of encoded macroblocks are found. These numbers are obtained as a result of encoding procedure. However, encoding has to be done twice to use macroblock bit count information properly. Because this method increases the encoder complexity substantially, another method is also proposed in the paper. It is shown that if the bit count information of the previous frame is used for the current frame it also gives satisfactory results. This method works very efficiently for frames which have low and moderate motion complexity. However it does not have a good performance when scene changes. After obtaining the bit count information, the indexes of macroblocks are sorted according to that information. In the last step macroblocks are allocated into the slices in a periodic manner.

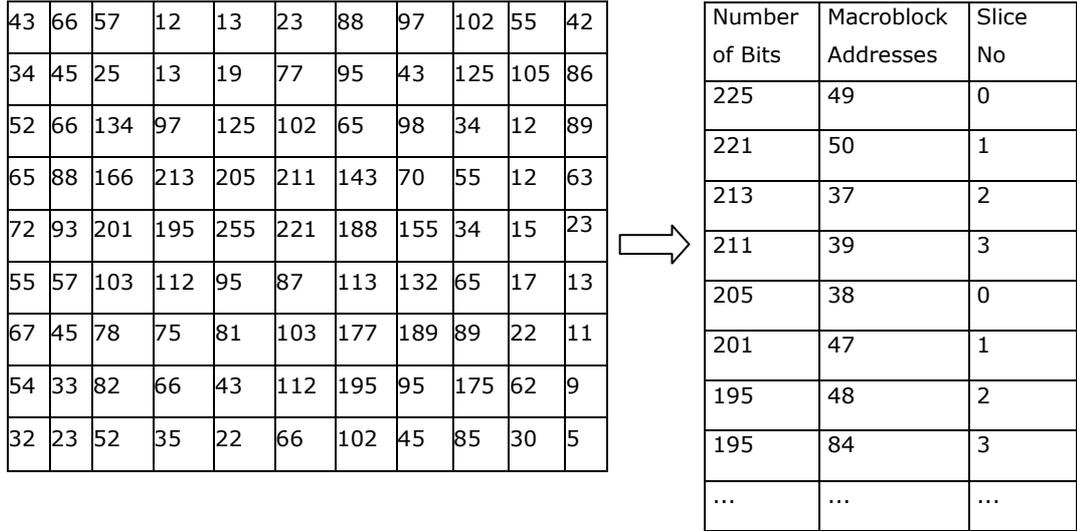


Figure 4.9: Determining bit count of macroblocks and sorting their addresses

For instance if the number of slices is 4 the indexes of macroblocks which have the 4 highest number of bits are assigned to slice groups 0,1,2,3 respectively. Then the macroblock, which has the next highest bit count, is assigned to slice group 0 and this process goes on until the end of the frame. These steps are illustrated in Figures 4.9.

The MBAMap can be demonstrated mathematically as follows:

$$M[i][j] = \text{mod}(ind_{MB[i][j]}, N) \quad (4.8)$$

In this equation, M denotes the 2D MBAMap structure, i and j denote the row and column numbers MBAMap respectively and N represents the the number of slice groups. $MB [i][j]$ denotes the macroblock that is located i^{th} row and j^{th} column of MBAMap. Lastly, ind represents the order of sorted macroblocks.

In [30], the results are enhanced by employing FEC, different types of FMO are not compared in this study. Efficiency of joint FEC-FMO with FMO, FEC and normally encoded video are compared.

4.5 Proposed Algorithm

The proposed algorithm is developed using the explicit algorithms mentioned in the previous sections. The aim of the algorithm is protecting the important macroblocks using dispersed allocation and UEP. Hence, both error correction and error concealment can be used efficiently in order to protect the interested region. The approach is composed of 2 parts. The first part is determining the region of interest. Unlike [23] which uses motion complexity to find the region, this approach finds important macroblocks according to the bit count of macroblocks. After detecting the important macroblocks and sorting them based on their bit count, they are separated into two groups as in Figure 4.10. The MBAmap can be demonstrated mathematically as follows:

$$M[i][j] = \begin{cases} 0, & ind_{MB[i][j]} \leq T/2 \\ 2, & otherwise \end{cases} \quad (4.9)$$

M denotes the 2D MBAmap structure, i and j denote the row and column numbers MBAmap respectively. $MB [i][j]$ denotes the macroblock that is located i^{th} row and j^{th} column of MBAmap. Lastly, ind represents the order of sorted macroblocks.

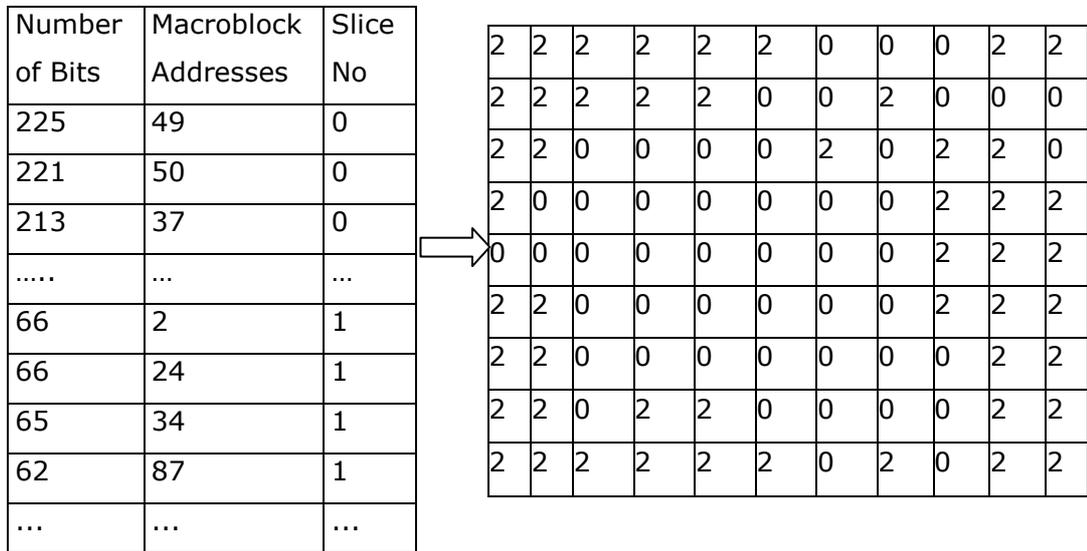


Figure 4.10: Step1, macroblocks are splitted into 2 groups according to their bit count numbers

The remaining macroblocks are allocated such that the macroblocks which have the higher bit count number are collected in one slice and the remaining macroblocks are collected in the last slice. Hence intra picture prediction can be employed. Equations (4.10) and (4.11) are used to determine the final MBAmap pattern. The pattern of final MBAmap is demonstrated in Figure 4.12.

$$M[i][j] = \begin{cases} 0, & W * i + j \text{ even} \\ 1, & W * i + j \text{ odd} \end{cases} \text{ where } ind_{MB[i][j]} \leq T/2 \quad (4.10)$$

$$M[i][j] = \begin{cases} 2, & ind_{MB[i][j]} \leq 3T/2 \\ 3, & \text{otherwise} \end{cases} \text{ where } ind_{MB[i][j]} > T/2 \quad (4.11)$$

W denotes the number of macroblocks in a row. T denotes the number of macroblocks in a frame. Lastly, ind represents the order of sorted macroblocks.

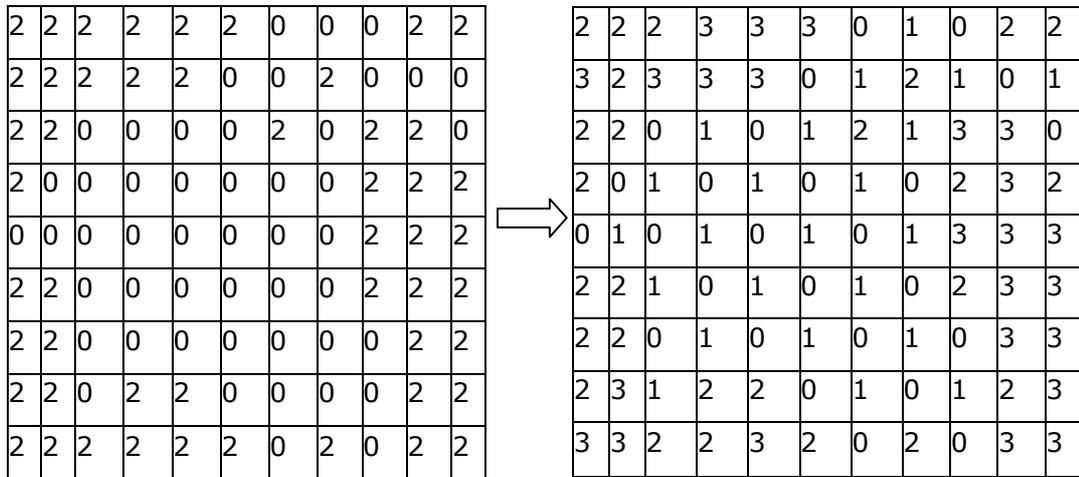


Figure 4.11: Splitting the foreground and background regions using checkerboard pattern and bit count information, respectively

This procedure does not always split the macroblocks of foreground region equally into the slices. In order to solve this problem, the following approach

can be followed. Firstly, macroblocks of foreground region which do not have any adjacent macroblocks from this region are detected. Secondly, their SGid's are adjusted according to the macroblock numbers of these groups. For instance in Figure 4.13 there are 5 macroblocks from slice 0 and 3 macroblocks from slice 1. Besides that, one of the slice0's macroblocks does not have any neighboring macroblock whose SGid is 0 or 1. Hence, SGid of the first macroblock alters from 0 to 1.

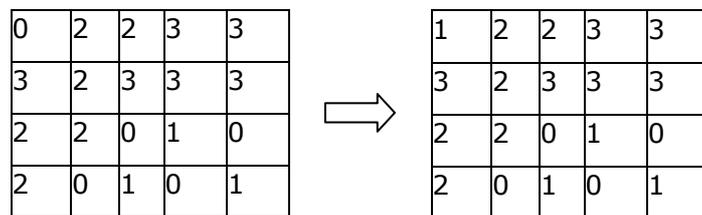


Figure 4.12: Adjusting slice group is of an isolated macroblock according to macroblock numbers of the groups

CHAPTER 5

SIMULATION ENVIRONMENT

5.1 Simulator

During simulations, loss simulator of JVT is used [31]. This simulator can be used both for AVC coding. The simulator uses byte stream oriented NALU's which have only start code prefix. Program uses 3 parameters for AVC coded videos. They are input, output and error pattern file. The error pattern files, which are used for wired network simulations, are explained in the following subsection. The error pattern files of wireless channel simulation are prepared according to the method which is described in subsection 5.3.

5.2 Wired Network Simulation Model

In this section, the characteristics of wired network simulation platform are explained [32]. First of all, the videos are coded such that no slice exceeds 1400 bytes including RTP/UDP/IP header. Moreover, PPS and SPS are assumed to be successfully transmitted. The network consists of 1 sender and one receiver which communicate over the Internet Backbone. Average delay is 150 ms. Bitrates of the videos are adjusted according to support the requirements of [32]. Frame rate of the videos is 15 fps. %3, %5, %10 and %20 [33] packet error files are used for all the stages. Besides that, a hypothetical %30 packet loss rate file is used for third stage of the simulation. Since the worst packet loss rate is %30 according to [2], it is not necessary to examine when the packet loss rate is higher than %30. Congestion control is not employed.

5.3 Wireless Channel Simulation Model

Wireless channels are very challenging platforms to model because many features can alter the performance of the channel. Hence, the simulation has

to be performed according to some assumptions. The aim is to create a point to point Rayleigh Fading Wireless Channel for Slow and Fast fading cases. Packet loss sequences are obtained based on [34], [35]. The simulations are only carried out for indoor environment. The wireless channel is modeled according to [34]. An SNR sequence is obtained at the output which illustrates the characteristics of channel at definite intervals. This SNR sequence is used to calculate the estimated packet success rate from the probability of bit error rate using equation (1) in [35]. These steps are repeated both for slow and fast fading channels.

5.3.1 Channel Modeling

The main aim of this step is preparing a model which simulates real environment characteristics as accurate as possible. A simple 2 state Gilbert Elliot is one of the most applied methods among all of them. The states are good and bad. The transition probabilities are used to determine the next state. A figure of this model is illustrated in Figure 5.1.

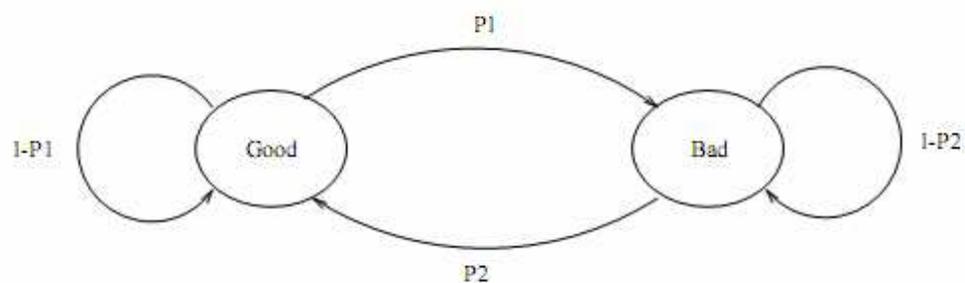


Figure 5.1: 2 state Gilbert Elliot Model [36]

The transition probabilities are generally determined using randomly generated values. Although this model is very successful for basic simulations the number of states can be insufficient to model more complex channels. Hence another model is proposed by [34]. According to this model the

current SNR interval of the signal is found instead of finding whether the transmission of packet is performed. The number of these intervals and their ranges are affected from the type of fading. Therefore these intervals are calculated for slow and fast fading channels. The channel is formed using Finite State Markov Model where states determine the SNR intervals. Packet oriented streaming is used. In order to use the same packet error loss rate for all the packets, state transitions are done after each packet transmission. The fading characteristic is assumed to be Rayleigh Fading. Hence, the probability of received SNR is calculated as follows

$$P(y) = \frac{1}{y_0} \exp\left(-\frac{y}{y_0}\right) \quad (5.1)$$

Where y represents the SNR value of received signal and y_0 is the average SNR value of the received signal. According to the [34] the number of transitions is determined using the following formula.

$$N(\Gamma) = \sqrt{\frac{2\pi\Gamma}{y_0}} f_m \exp\left(-\frac{\Gamma}{y_0}\right) \quad (5.2)$$

In this formula f_m represents the Doppler frequency and Γ represents the crossing level. Therefore the number of transitions in the fast fading case is higher than slow fading case. This number is determined for a unit time interval. The Γ levels are calculated according to an SNR partitioning procedure. If there are k states than there are $k+1$ boundary values. Two of these values are known. Γ_1 and Γ_{k+1} are zero and infinity respectively.

Determining the range and the boundary values of SNR intervals is a crucial task. They should be adjusted such that the SNR level of the received signal remains in the same state during a packet transmission and transition only occurs for the next packet transmission. Moreover this interval should not be very large because the difference between upper and lower boundaries of a state interval can be big. Hence BER probability of a state can take very high or very low values which make the model unstable.

If τ_k is the average duration of the received signal between Γ_k and Γ_{k+1}

then

$$\tau_k = \frac{\pi_k}{N(\Gamma_k) + N(\Gamma_{k+1})} \quad (5.3)$$

Where π_k is equal to $P(\Gamma_k \leq y < \Gamma_{k+1})$ which is found using (5.1)

$$\pi_k = \int_{\Gamma_k}^{\Gamma_{k+1}} P(y) dy = \exp\left(-\frac{\Gamma_k}{y_0}\right) - \exp\left(-\frac{\Gamma_{k+1}}{y_0}\right) \quad (5.4)$$

Using (5.2) and (5.4) the equation (5.3) is written as follows

$$\tau_k = \frac{\exp\left(-\frac{\Gamma_k}{y_0}\right) - \exp\left(-\frac{\Gamma_{k+1}}{y_0}\right)}{\sqrt{\frac{2\pi\Gamma_k}{y_0}} \exp\left(-\frac{\Gamma_k}{y_0}\right) + \sqrt{\frac{2\pi\Gamma_{k+1}}{y_0}} \exp\left(-\frac{\Gamma_{k+1}}{y_0}\right)} \cdot \frac{1}{f_m} \quad (5.5)$$

In order to satisfy the SNR range requirements of τ_k which represents coherence channel duration has to be bigger than or equal to τ_p which represents packet transmission time. That is, the state remains same during a packet transfer. The authors use a coefficient to demonstrate the relation between τ_k and τ_p . $c_k = \tau_k / \tau_p$. In order to fulfill this requirement, c_k has to be bigger than or equal to 1 [34]. Based on this information, the formula (5.5) can be represented in the following form.

$$c_k = \frac{\exp\left(-\frac{\Gamma_k}{y_0}\right) - \exp\left(-\frac{\Gamma_{k+1}}{y_0}\right)}{\sqrt{\frac{2\pi\Gamma_k}{y_0}} \exp\left(-\frac{\Gamma_k}{y_0}\right) + \sqrt{\frac{2\pi\Gamma_{k+1}}{y_0}} \exp\left(-\frac{\Gamma_{k+1}}{y_0}\right)} \cdot \frac{1}{f_m \tau_p} \quad (5.6)$$

From the equation it can be observed explicitly that if Doppler frequency f_m , packet transmission time τ_p , average channel SNR y_0 are known than all Γ values where $i \in 1, 2, \dots, k+1$ can be found using this formula. Since the first and the last Γ values are known if there are k intervals then there are $k-1$ unknown state boundary values. Figure 5.2 illustrates this state chain.

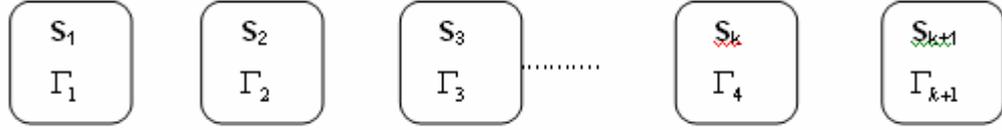


Figure 5.2: SNR states of the wireless channel

S_i denotes the i^{th} Markov Chain State. The number of states changes according to the c_k and $f_m \tau_p$ coefficients. Because τ_p is a constant value, $f_m \tau_p$ can be used to represent fading characteristic. Large $f_m \tau_p$ value means a fast fading channel and small $f_m \tau_p$ value means a slow fading channel.

When the channel is fast fading, the average number of transitions increases which decreases the average duration of remaining in the same state. Therefore when f_m is high, the received SNR value changes more rapidly. On the other hand SNR intervals of a slow fading channel are smaller than the fast fading channel. Moreover slow fading channels have more states than fast fading channels. Hence the consecutive SNR values do not change abruptly. Equal probability method and Minimum Mean Square Error can also be used for determining SNR transition values. In Tables 5.1 and 5.2 SNR threshold values of slow and fast fading channels are illustrated.

The parameters which are used to calculate these values are as follows. τ_p is 2.75 msec, c_k is approximately 3 for both channels. f_m values are 1 Hz and 10 Hz for Slow and Fast Fading channels respectively.

Table 5.1: SNR state boundary values of Fast Fading Channel

State	Γ_1	Γ_2	Γ_3	Γ_4	Γ_5	Γ_6	Γ_7	Γ_8	Γ_{10}
dB	-Inf	-6.412	-1,281	1.796	4,025	5,79	7,27	8,563	Inf

Table 5.2: SNR state boundary values of Slow Fading Channel

State	Γ_1	Γ_2	Γ_3	Γ_{30}	Γ_{31}	Γ_{32}	Γ_{59}	Γ_{60}
dB	-Inf	-17,45	-12,29	9,149	9,428	9,698	15,02	Inf

After determining SNR boundaries the transition probabilities are calculated according to these threshold values. Figure 5.3 demonstrates the Finite State Markov Model of Wireless channel. Transition probabilities from state s_k to s_{k+1} and from s_k to s_{k-1} are represented by $P_{k,k+1}$ and $P_{k,k-1}$ respectively. Based on some approximations [34] the transition probabilities are calculated using the equations (5.7) and (5.8). The probability of remaining in s_k is represented by π_k it can be easily found by subtracting the sum of these values from 1.

$$P_{k,k+1} = \frac{N(\Gamma_{k,k+1})\tau_p}{\pi_k} \quad (5.7)$$

$$P_{k,k-1} = \frac{N(\Gamma_{k,k-1})\tau_p}{\pi_k} \quad (5.8)$$

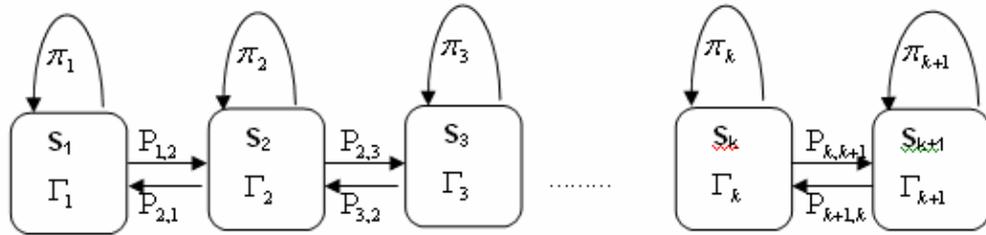


Figure 5.3: FSMC representation of the wireless channel

The transition probabilities of slow and fast fading channels are illustrated in Table 5.3 and Table 5.4 respectively. Obtaining these probabilities complete the process of preparing an accurate channel model for different fading cases.

Table 5.3: State transition probabilities of fast fading channel

Transition	Value	Transition	Value	Transition	Value
$P_{1,2}$	0.547			π_1	0,453
$P_{2,3}$	0.375	$P_{2,1}$	0.211	π_2	0,413
$P_{3,4}$	0.313	$P_{3,2}$	0.287	π_3	0,4
$P_{4,5}$	0.266	$P_{4,3}$	0.344	π_4	0,39
$P_{5,6}$	0.224	$P_{5,4}$	0.394	π_5	0,381
$P_{6,7}$	0,185	$P_{6,5}$	0.442	π_6	0,373
$P_{7,8}$	0,147	$P_{7,6}$	0.488	π_7	0,365
$P_{8,9}$	0,108	$P_{8,7}$	0.535	π_8	0,356
		$P_{9,8}$	0.547	π_9	0,453

Table 5.4: State transition probabilities of slow fading channel

Transition	Value	Transition	Value	Transition	Value
$P_{1,2}$	0.323			π_1	0,676
$P_{2,3}$	0.303	$P_{2,1}$	0.149	π_2	0,556
$P_{3,4}$	0.285	$P_{3,2}$	0.181	π_3	0,533
....
$P_{30,31}$	0.213	$P_{30,29}$	0.304	π_{30}	0,483
$P_{31,32}$	0.211	$P_{31,30}$	0.307	π_{31}	0,482
$P_{32,33}$	0,209	$P_{32,31}$	0.309	π_{32}	0,482
....
$P_{58,59}$	0,166	$P_{58,57}$	0.383	π_{58}	0,451
$P_{59,60}$	0,165	$P_{59,58}$	0.385	π_{59}	0,45
		$P_{60,59}$	0.386	π_{60}	0,614

The SNR sequences of different fading channels are illustrated in Figures 5.4 and Figure 5.5. Change rate of these levels can be easily observed from these figures. In the last part of this step, a random SNR sequence is generated based on the transition probabilities and average SNR. The first value of this sequence is average SNR. Then using a random generator and transition probabilities the next state is found. It can be seen from the tables that the transition probabilities have values such that they tend to increase in the direction of average SNR value. The average signal to noise ratio of the received signal is approximately 8 dB for both fading cases. Carrier frequency of the transmitted signal is 915 MHz. Bitrate is 32 kbit/s. Lastly the simulations are performed for indoor environment.

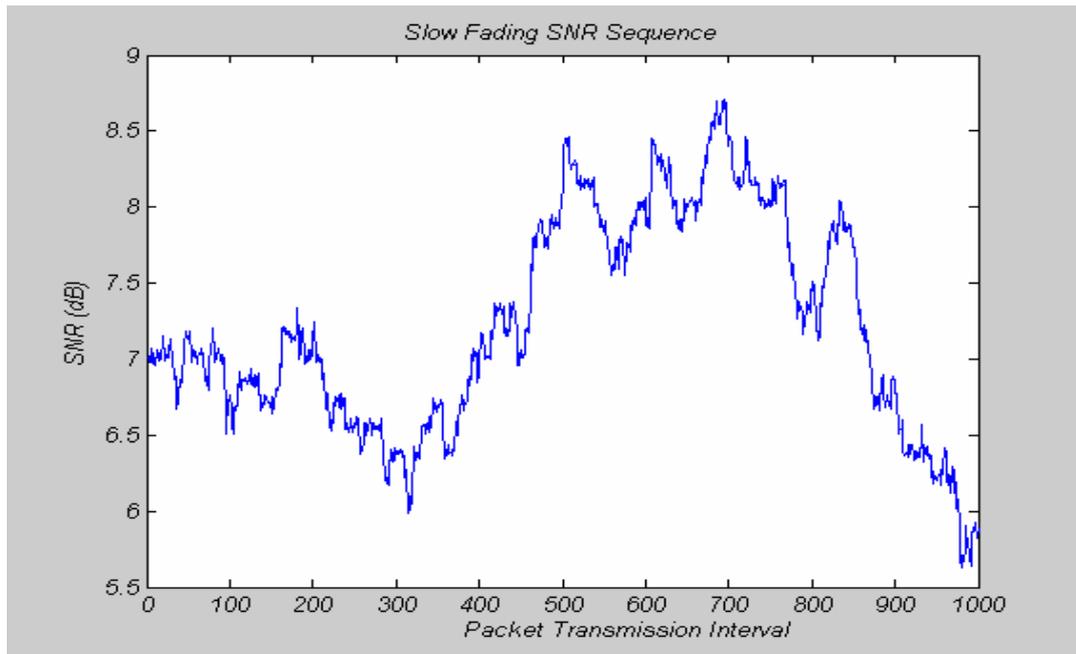


Figure 5.4: SNR sequence of the slow fading channel

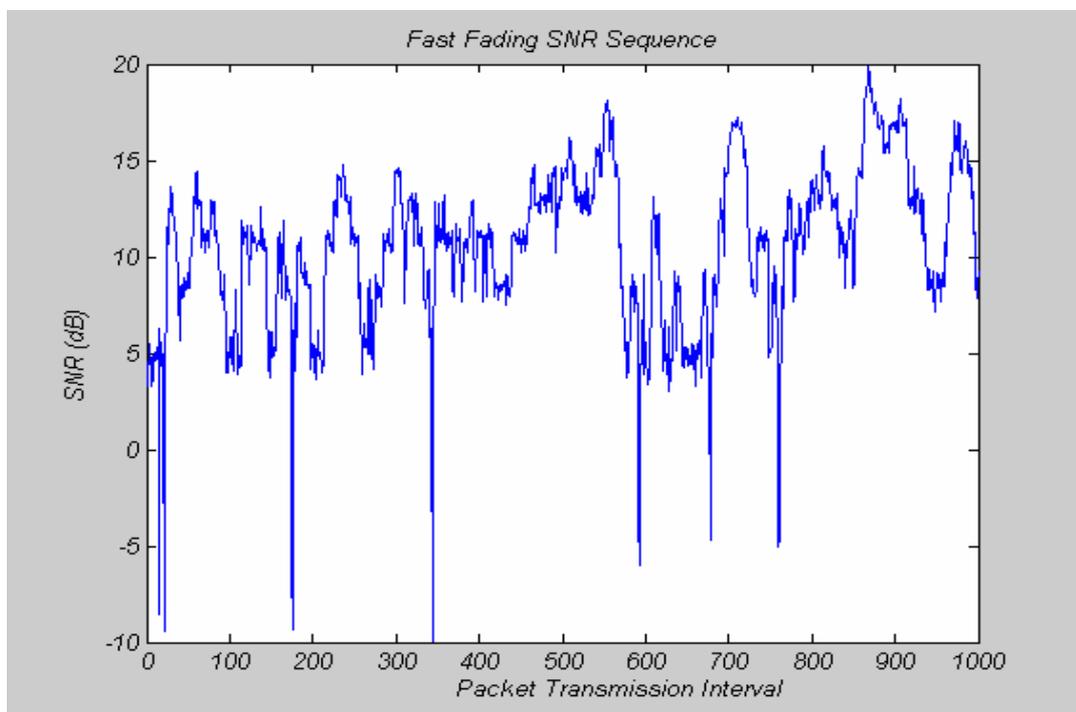


Figure 5.5: SNR sequence of the fast fading channel

After generating channel SNR sequence, packet loss pattern is found according to this sequence in the second step. This pattern is found using the packet error rate approximation in [35]. The estimated packet success rate is calculated using the following formula.

$$E(PSR) = \left((1 - P_e)^8 \right)^{P_B} \left(\sum_{i=0}^{P_b} \binom{8}{i} P_e^i (1 - P_e)^{8-i} \right)^{L - P_B} \quad (5.9)$$

L represents the number of bytes in the packet. P_B is the number of header bytes that are assumed to be transmitted correctly. P_b represents the maximum number of erroneous bits for a byte except header byte. Lastly P_e is the Bit Error Rate (BER).

Since the modulation is BFSK (Binary Frequency Shift Keying) P_e is calculated using the formula.

$$P_e = \frac{1}{2} \exp\left(-\frac{\sigma}{2}\right) \quad (5.10)$$

σ symbol represents SNR value of the received signal. If (5.10) is substituted in equation (5.9), then estimated packet success rate value of the j^{th} packet can be written as follows

$$E(PSR_j) = \left(\left(1 - \frac{1}{2} \exp\left(-\frac{\sigma_j}{2}\right) \right)^8 \right)^{P_B} \left(\sum_{i=0}^{P_b} \binom{8}{i} \left(\frac{1}{2} \exp\left(-\frac{\sigma_j}{2}\right) \right)^i \left(1 - \frac{1}{2} \exp\left(-\frac{\sigma_j}{2}\right) \right)^{8-i} \right)^{L - P_B} \quad (5.11)$$

If these parameters are set according to the wireless channel characteristics of the simulator which is used in [29], L is set to 11 bytes (1 header). P_B and P_b values are set to 1.

$$E(PSR_j) = \left(\left(1 - \frac{1}{2} \exp\left(-\frac{\sigma_j}{2}\right) \right)^8 \right) \left(\sum_{i=0}^1 \binom{8}{i} \left(\frac{1}{2} \exp\left(-\frac{\sigma_j}{2}\right) \right)^i \left(1 - \frac{1}{2} \exp\left(-\frac{\sigma_j}{2}\right) \right)^{8-i} \right)^{10} \quad (5.12)$$

Since SNR values of the received signal are found in the first step, the success rate of each packet can be determined by exploiting this information in the formula. In order to find packet success rate sequence the estimated packet success probabilities are used for each packet transfer. A random

number generator is used which has a uniform distribution between 0 and 1 to determine whether the packet is successfully transmitted. This number is compared with the estimated packet success rate. If the generated number is smaller than the estimated value then the packet is assumed to be transmitted accurately otherwise error is occurred.

CHAPTER 6

EXPERIMENTAL RESULTS

In order to evaluate the performance of the various FMO types from different point of views, 3 popular test sequences are used which are Foreman, Akiyo and Carphone test sequences. Basic characteristics of these video sequences are explained in the following table

Table 6.1: Main characteristics of the video test sequences

Test Sequences	Foreman	Akiyo	Carphone
Total Number of Frames	300	300	300
Spatial Resolution	QCIF (176x144)	QCIF	QCIF
GOP Size	15	15	15
Motion Activity	Moderate	Low	Moderate
Prediction Type	P	P	P

Since Foreman and Carphone have similar characteristics. Only the test results of Akiyo and Foreman are explained in this chapter. The motion activity of Foreman sequence is caused by the movement of the body, head, gestures of the speaker and camera movement. On the other hand Akiyo sequence has a low motion complexity. Slow movement of head and gestures of the speaker are the sources of the motion. Different FMO types are evaluated under 4 different test conditions. In every condition the videos are

coded using 8 different FMO types. These FMO types are Interleave, Dispersed, Foreground with Leftover, Box-out, Raster Scan, Wipe, Explicit Type 1 (Spreading important macroblocks) and Proposed FMO (Collecting important macroblocks). The last two types can also be referred as Explicit Types in the remaining sections. During simulations clockwise direction is used for Box-Out Type. Moreover scanning patterns of Wipe and Raster Types start from the top right macroblock.

In the first part, Foreman sequence is encoded at approximately 300 kbps and Akiyo sequence is encoded approximately 90 kbps. These rates are not exactly the values mentioned, since quantization parameters can not be adjusted properly. The frames are segmented into the 4 slices. In order to adjust the size of the slices, fragmentation can be used to divide slices into several parts which contain less byte than MTU. Moreover, the temporal resolution is 15 fps for this case. The transmission of the encoded videos is simulated using the loss simulator for error files which are explained in Section 5.3. The received data is decoded offline.

In the second part, the simulations are carried out for slow and fast fading wireless channels. All the videos are encoded at 32 kbps as specified in [28]. In this part, 8 slices are used for each frame like in the [28]. Besides that, temporal resolution is adjusted to 10 fps. While creating the simulator, the first byte is assumed to be received correctly for header information. The error file sequences are prepared according to the model described in the Section 5.4. The average signal to noise ratio of the received signal is approximately 8 dB for both fading cases. Lastly, carrier frequency of the transmitted signal is 915 MHz.

In the third part, the first part is repeated using Forward Error Correction. This part is done especially to see the performance of Proposed Algorithm. Since Forward Error Correction increases the payload, the videos are encoded at a rate to fulfill the requirements of the simulator 1. Besides that, it is necessary to create packets which have same number of byte to use block coding like Reed Solomon. Unfortunately, most of the coded slices do not include equal number of bytes. Zero Padding can be used to solve this problem, but it decreases the efficiency crucially. Therefore a different approach is exploited. All the frames which formed the GOP are collected in a buffer. Then slices of the consecutive frames are appended until the size is smaller than a definite number. The remaining part is filled using padding.

The slices of the last frames generally don't contain sufficient number of bytes. These bytes are sent in the next GOP. This method increases the delay but the amount of delay is approximately 4 frames and it only affects the beginning of the streaming. This method is also used for Test 1 and Test 2 in order to create equal sized packets. Reed Solomon is used for FEC

In the last part, wireless simulations are repeated using Forward Error Correction. The videos are encoded at a lower rate which is approximately 20 kbps to adjust the channel conditions. The method, which is applied in Test 3, is also exploited to overcome the size problem. However, while carrying out these methods the intra frames and inter frames are collected separately, because the size difference between the slices of these frame types is huge. The test results are evaluated according to their encoder performance, decoder performance and objective quality of the transmitted sequence.

PSNR values of the test results are luminance PSNR values and they are calculated using the following formula:

$$PSNR = 10 \times \log \left(255^2 / \left(\sum_{i=1}^W \sum_{j=1}^H (AF_{i,j} - RF_{i,j})^2 \right) \right) \quad (6.1)$$

In this formula, W denotes the width of the frame, H denotes the height of the frame. $AF_{i,j}$ and $RF_{i,j}$ denote the pixel value of the actual frame and reconstructed frame at position i, j respectively. During simulations, Weighted Pixel Averaging is used for error concealment of intra macroblocks, and Boundary Matching Based Motion Vector Recovery is used for error concealment of inter macroblocks. If a frame is totally corrupted the previous frame is copied. Test results are evaluated based on 2 performance criteria. which are Encoder Performance and Objective Quality. Encoder Performance measurements illustrate PSNR value of the encoded sequence and total necessary bitrate. Encoding procedure is performed offline. Average PSNR values and number of undecoded macroblocks are used in order to compare objective qualities.

6.1 Test 1

6.1.1 Encoder Performance

Motion estimation and intra picture prediction are very important for quality of the coded sequences which are transmitted at constant rate. Patterns, which include neighboring macroblocks in the same slice, have higher PSNR values because they compress the video more, so they can use better quantization parameters for constant channel rate. Therefore, coded sequences, which are created using Raster Scanning, has higher PSNR values. On the other hand, the patterns, which have checkerboard characteristics, do not have high PSNR values because intra picture prediction can not be done at the slice borders. Hence, sequences which are coded using dispersed type usually have lower PSNR values. The PSNR values of the encoded Foreman bitstreams can be seen at Table 6.2. Encoded Foreman sequence, which is coded using Wipe FMO Type, has the highest PSNR values. On the other hand, video that is coded using dispersed type has a PSNR value smaller than others. Streams, which are encoded using Box-out and Interleaving, have the best PSNR values for Akiyo and Foreman sequences respectively.

Table 6.2: Encoder performance measurements of coded Foreman sequences using different FMO types for Test1

Foreman	Interleave	Dispersed	Foreground With Leftover	Box-out	Raster Scan	Wipe	Explicit Type 1	Proposed
PSNRY(dB)	43,71	43,45	43,8	43,8	43,81	43,86	43,61	43,62
Bitrate(kbps)	300,07	300,1	300,07	300,09	300,06	300,04	304,28	304,31

Moreover there is also an interesting situation. Although Akiyo sequence is coded at a rate smaller than other sequences, PSNR values of the coded Akiyo sequences are higher than other sequences. This is caused by the low motion activity feature of Akiyo sequence. Table 6.3 illustrates the encoder performance measurements of Akiyo sequence. Although same bitrate is used for all FMO types PPS increases the bitrates of Explicit Types.

Table 6.3: Encoder performance measurements of coded Akiyo sequences using different FMO types for Test1

Akiyo	Interleave	Dispersed	Foreground With Leftover	Box-out	Raster Scan	Wipe	Explicit Type 1	Proposed
PSNRY(dB)	45,49	45,11	45,91	45,92	45,59	45,7	45,34	45,44
Bitrate(kbps)	90,19	90,14	90,15	90,12	90,19	90,17	94,49	94,43

Therefore required rate of the two explicit algorithms is more than the definite types. When PPS is sent for every frame the bit rate of PPS is 4 kbps. Therefore percentage of PPS payload to the total bitrate is $(4/300 * 100) = \%1.33$.

6.1.2 Objective Quality

In order to compare the objective quality level, PSNR values of the decoded videos are used. The simulations are carried out using the %3, %5, %10 and %20 packet loss patterns. The average PSNR values and number of undecoded macroblocks are illustrated in tables from Table 6.4 to Table 6.7.

6.1.2.1 %3 Packet Loss Results

In the first case, the performance of different sequences is evaluated when the packet loss rate is %3. Since error percentage is small, PSNR values of

the streams are close. Moreover the number of undecoded macroblocks is around 915 for both sequences. Table 6.4 illustrates the quality measurements of Foreman and Akiyo sequences when the packet loss rate is %3.

Table 6.4: Objective Quality Measurements of Foreman and Akiyo sequences for different FMO types when packet loss rate is %3

%3		Interleave	Dispersed	Foreground With Leftover	Box-out	Raster Scan	Wipe	Explicit Type 1	Proposed
Foreman	Number of Undecoded MB	917	908	916	914	914	914	913	915
	Average PSNR(dB)	36,22	36,55	36,38	36,46	36,31	35,3	36,16	35,63
Akiyo	Number of Undecoded MB	917	908	916	914	914	914	913	913
	Average PSNR(dB)	42,98	43,14	43,85	43,54	43,15	43,15	43,19	43,95

Video coded with Dispersed Type has the highest average PSNR values for Foreman sequence. According to this result, it can be concluded that, spatial error concealment plays an important role when the channel is error prone. The patterns, which separate adjacent macroblocks, have higher PSNR values than others. Figure 6.1 illustrate the PSNR values of different FMO types for Foreman sequence. However results of Akiyo sequence contradict with the results of Foreman. According to Table 6.5 Foreground with Leftover and Proposed FMO types have 2 highest PSNR values. On the other hand, videos that are coded with Interleave and Dispersed types have the worst PSNR characteristics. Since error ratio is low and motion characteristic of Akiyo

sequence is low, spatial concealment may not be necessary in this case.

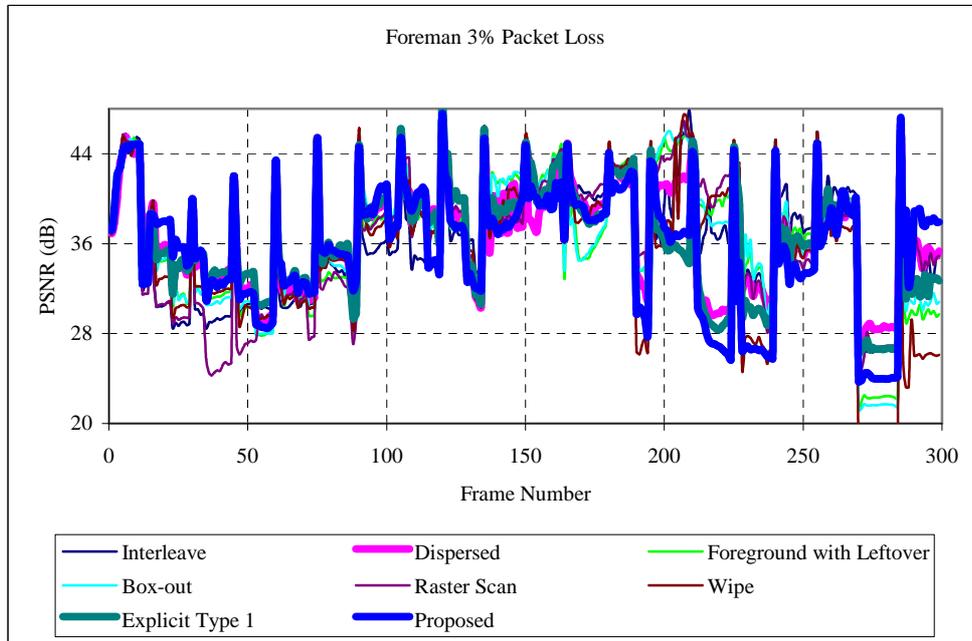


Figure 6.1: PSNR values of different FMO types for Foreman sequence when packet loss rate is %3

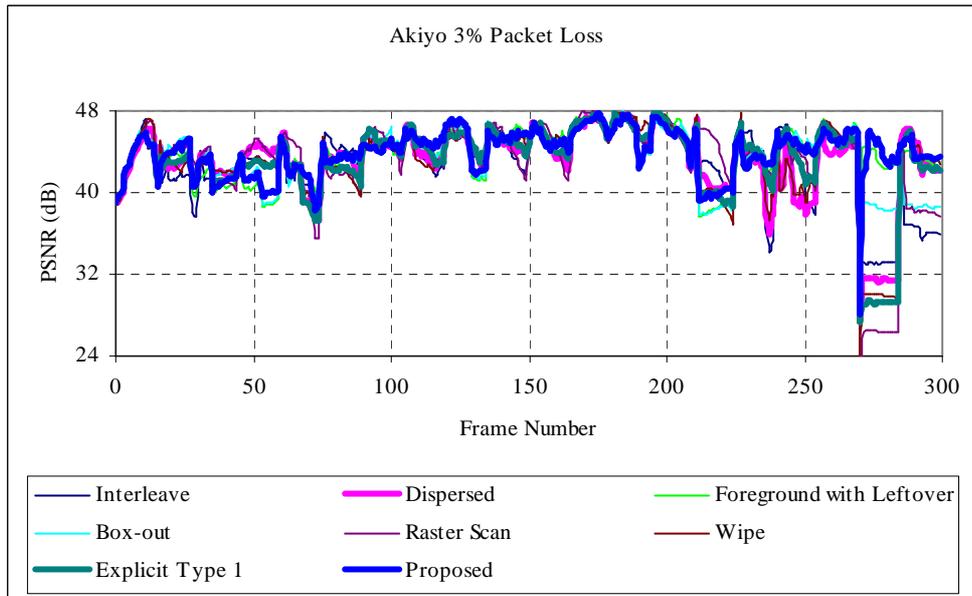


Figure 6.2: PSNR values of different FMO types for Akiyo sequence when packet loss rate is %3

6.1.2.2 %5 Packet Loss Results

In this step, transmission error percentage increases from %3 to %5. Table 6.5 illustrates the quality measurements of Foreman and Akiyo sequences when the packet loss rate is %5. According to Table 6.5, when the Foreman sequence results are examined, Dispersed coded video has the highest PSNR value and the video coded with Foreground with Leftover FMO type has the worst average PSNR value. The difference between these two types is 3,5 dB. Interleave type follows Dispersed type. Figure 6.3 illustrate the PSNR values of different FMO types for Foreman Sequence

Table 6.5: Objective Quality Measurements of Foreman and Akiyo sequences for different FMO types when packet loss rate is %5

%5		Interleave	Dispersed	Foreground With Leftover	Box-out	Raster Scan	Wipe	Explicit Type 1	Proposed
Foreman	Number of Undecoded MB	1811	1800	1801	1810	1800	1810	1836	1835
	Average PSNR(dB)	28,85	30,04	26,49	26,96	27,26	26,53	28,03	28,14
Akiyo	Number of Undecoded MB	1811	1800	1801	1810	1810	1810	1836	1825
	Average PSNR(dB)	31,03	32,48	29,27	27,98	26,71	26,2	31,3	30,69

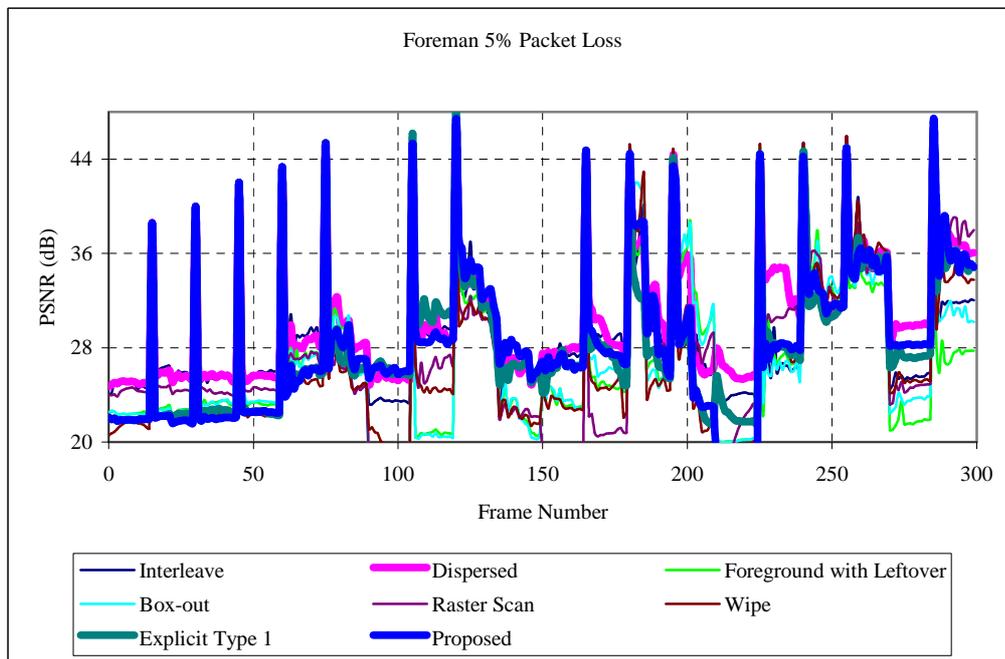


Figure 6.3: PSNR values of different FMO types for Foreman sequence when packet loss rate is %5

When the results of Akiyo sequence which are found in Table 6.5 are evaluated, it can be seen that the Dispersed and Wipe types have the best and worst PSNR characteristics respectively. Moreover, Explicit Type 1 is second type instead of Interleave Type. The difference between Dispersed and Wipe Types is 5 dB which is approximately 1 dB when the packet loss rate is %3. This increase is higher than the one in Foreman case.

In both cases, Proposed FMO comes after Interleave and Explicit FMO types. PSNR values of Akiyo sequences which are coded using different FMO types are shown in Figure 6.4. According to the Table 6.6 different FMO types of each sequence has approximately 1800 number undecoded macroblocks

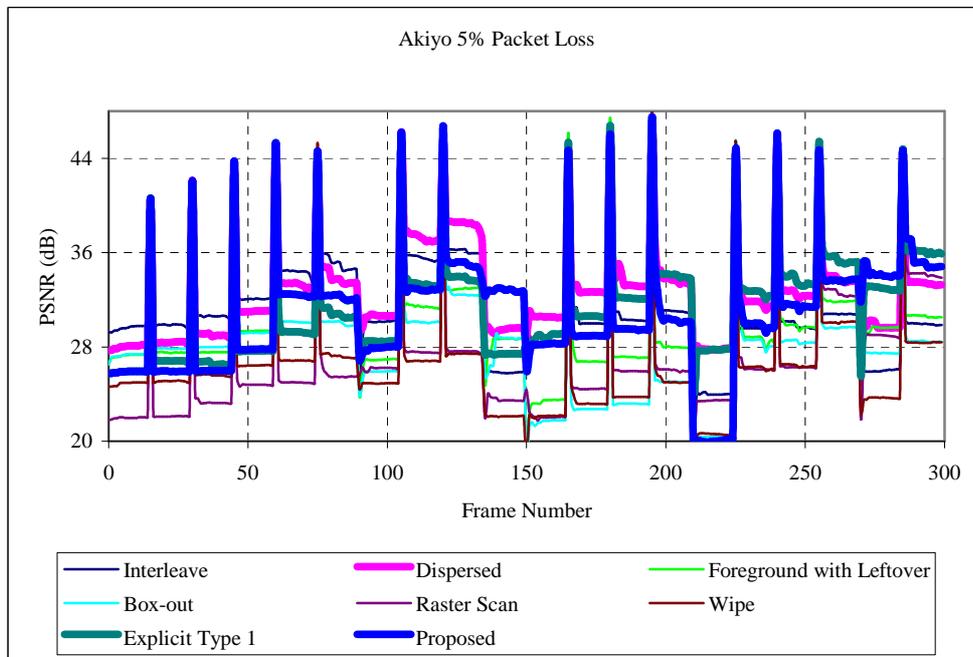


Figure 6.4: PSNR values of different FMO types for Akiyo sequence when packet loss rate is %5

6.1.2.3 %10 Packet Loss Results

As the error increases from %5 to %10, the difference among the FMO types can be observed more easily. According to Table 6.6, Dispersed Type has the best performance for both sequences. Interleave, Explicit 1 and Proposed types follow the Dispersed type like in the previous case. The number of undecoded macroblocks is nearly 3400 for all FMO types of each sequence. Results of Foreman sequence illustrates that Interleave Type has the second best PSNR value and Foreground with Leftover has the worst PSNR value. The difference between Dispersed and Foreground with Leftover Type is 3 dB. Figure 6.5 illustrates the results of Foreman sequence.

Table 6.6: Objective Quality Measurements of Foreman and Akiyo sequences for different FMO types when packet loss rate is %10

%10		Interleave	Dispersed	Foreground With Leftover	Box-out	Raster Scan	Wipe	Explicit Type 1	Proposed
Foreman	Number of Undecoded MB	3393	3384	3415	3390	3390	3390	3391	3387
	Average PSNR(dB)	26,78	27,86	24,82	24,98	25,38	24,93	26,06	25,96
Akiyo	Number of Undecoded MB	3393	3384	3415	3390	3390	3390	3391	3395
	Average PSNR(dB)	27,9	31,69	24,55	25,55	26,38	23,66	28,71	27,13

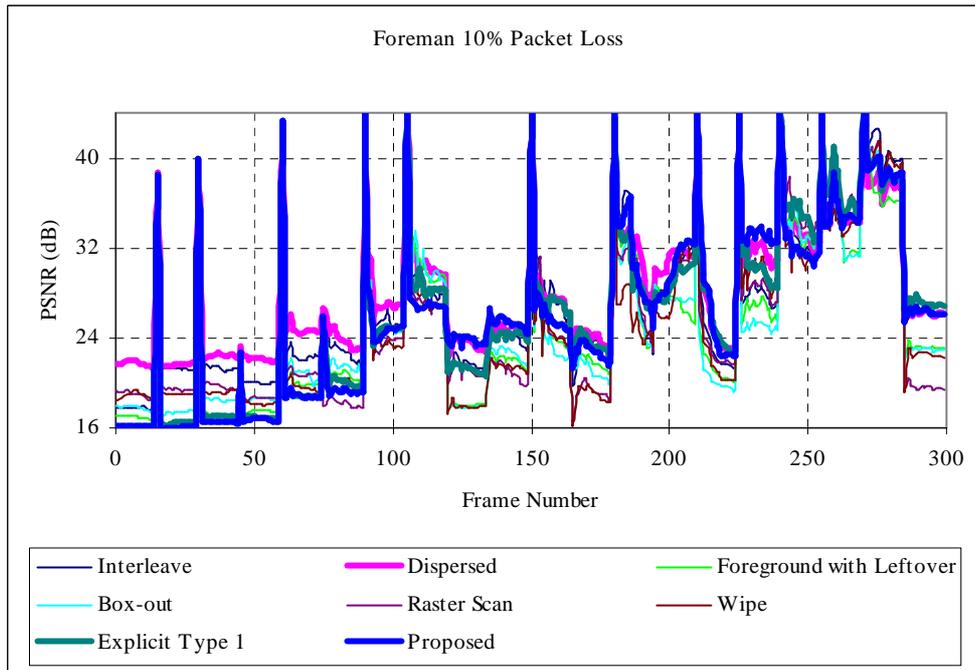


Figure 6.5: PSNR values of different FMO types for Foreman sequence when packet loss rate is %10

When the results of Akiyo sequence are evaluated, Explicit Type 1 has the second best average PSNR value. Moreover, Wipe type has the smallest PSNR values among all the types. The difference between Dispersed and Wipe Types is 8 dB which is more than the difference in the previous case. Figure 6.6 illustrates the results of Akiyo sequence.

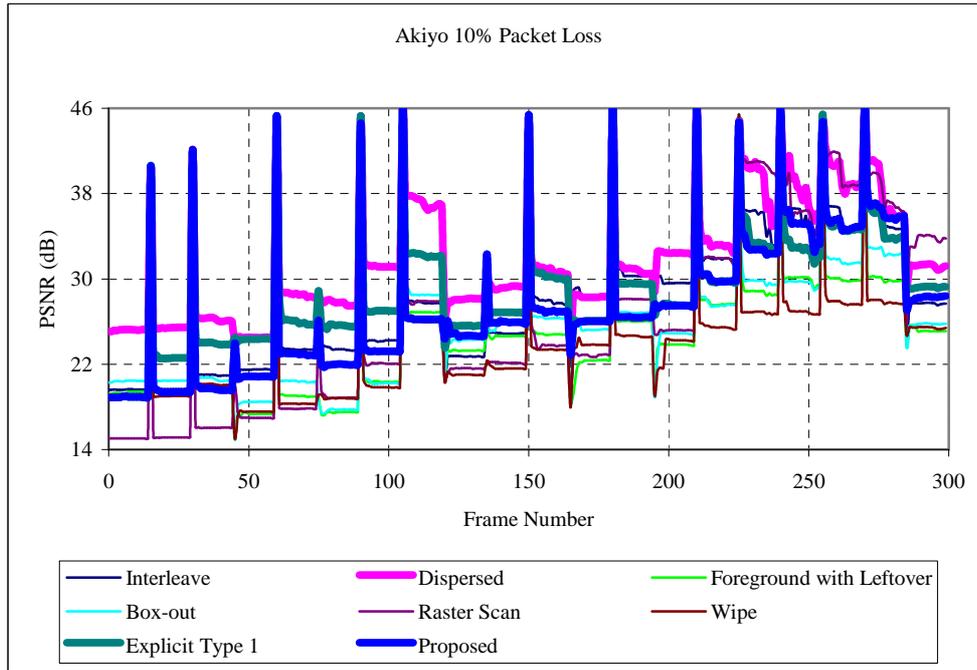


Figure 6.6: PSNR values of different FMO types for Akiyo sequence when packet loss rate is %10

6.1.2.4 %20 Packet Loss Results

The error ratio is increased to %20 in the last case of Test 1. The best 4 types are Dispersed, Interleave, Explicit Type 1 and Proposed again. According to the results of Foreman sequence in Table 6.7, Dispersed Type has the highest PSNR value which is 25.77 dB. Explicit Type 1, Interleave and Proposed Types

Table 6.7: Objective Quality Measurements of Foreman and Akiyo sequences for different FMO types when packet loss rate is %20

%20		Interleave	Dispersed	Foreground With Leftover	Box-out	Raster Scan	Wipe	Explicit Type 1	Proposed
Foreman	Number of Undecoded MB	6040	6137	6034	6023	6023	6023	6025	6026
	Average PSNR(dB)	25,05	25,77	23,46	23,59	22,9	22,32	25,11	25,02
Akiyo	Number of Undecoded MB	6040	6037	6034	6023	6023	6023	6025	6021
	Average PSNR(dB)	27,13	30,55	24,14	24,01	23,96	22,47	28,25	28,03

follow it. On the other hand Wipe Type has the worst PSNR characteristic among all FMO types. Its PSNR value is 22.32 dB. Hence, it can be concluded that spatial concealment enhances the PSNR approximately 3 dB for this video sequence when the packet error rate is between %5 and %20. Test results of Foreman sequence are illustrated in Figure 6.7.

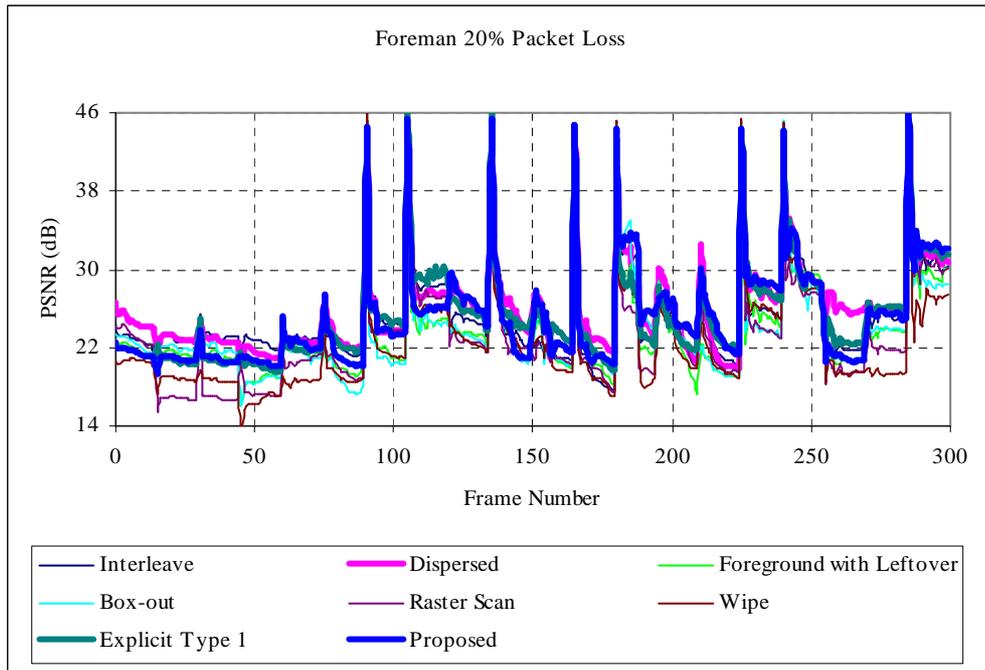


Figure 6.7: PSNR values of different FMO types for Foreman sequence when packet loss rate is %20

Average PSNR values of Akiyo sequence illustrates that, the order of FMO types is similar to the Foreman case. However, the PSNR difference among different FMO types is higher. PSNR values of Akiyo sequences which are coded using different FMO types are demonstrated in Figure 6.8

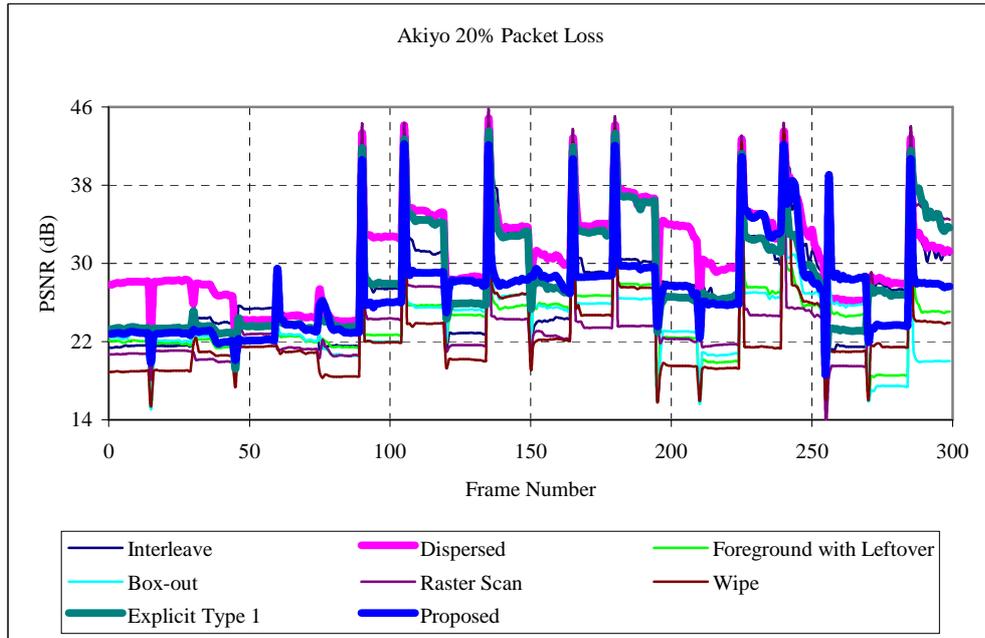


Figure 6.8: PSNR values of different FMO types for Akiyo sequence when packet loss rate is %20

In Figure 6.9 and Figure 6.10 the average PSNR values of different FMO types are illustrated for Foreman and Akiyo sequences. As can be seen from the figures when the packet loss rate is low, FMO patterns, which allocate adjacent macroblocks into the same slice, have higher PSNR values. As the packet loss rate increases Dispersed, Interleave, Explicit Type 1 and Proposed FMO types perform better because they are appropriate for applying error concealment.

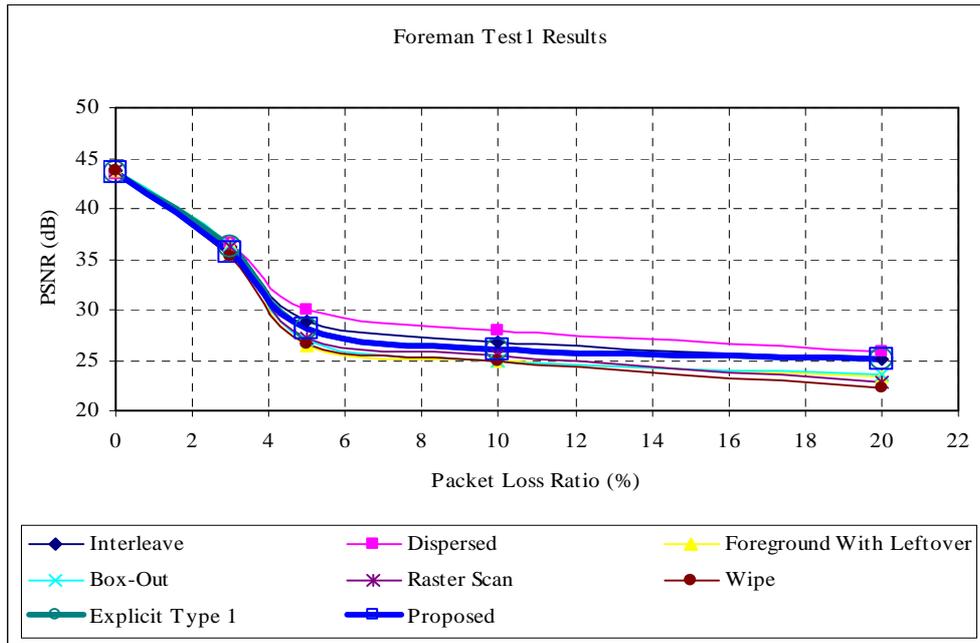


Figure 6.9: Average PSNR values of FMO types for Test 1 Foreman sequence

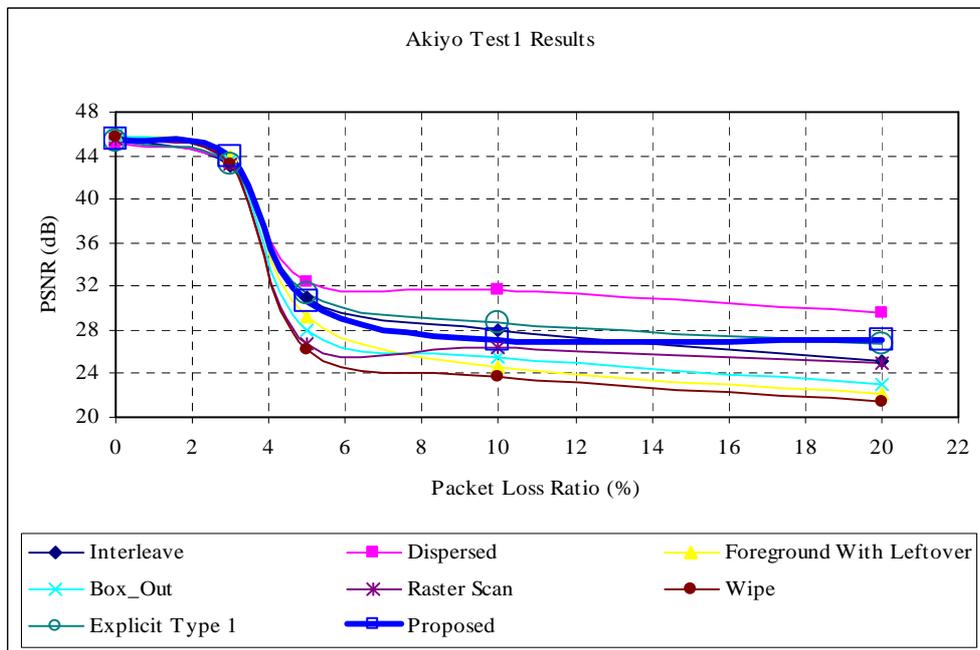


Figure 6.10: Average PSNR values of FMO types for Test 1 Akiyo sequence

6.2 Test 2

6.2.1 Encoder Performance

The first two encoder performance measurements of Test 2 are different from the results of Test 1 because the channel characteristic is changed and encoder parameters are adjusted for this medium. Firstly, PSNR values are compared. According to Table 6.8, Wipe Type has the highest PSNR value and Explicit Type 1 has the lowest PSNR value. This consequence shows that, FMO types, which form slices using neighboring macroblocks, have better quality.

Table 6.8: Encoder performance measurements of coded Foreman sequence using different FMO types for Test2

Foreman	Interleave	Dispersed	Foreground With Leftover	Box-out	Raster Scan	Wipe	Explicit Type 1	Proposed
PSNRY(dB)	30,55	30,42	31,07	31,02	31,39	31,92	30,1	30,28
Bitrate (kbps)	32,25	32,28	32,21	32,25	32,26	32,33	36,53	36,5

Table 6.9 illustrates the encoder performance measurements of Akiyo sequence. Since Akiyo sequence has the lowest motion activity, it is encoded using better quantization parameters for all FMO types. Hence, PSNR values of encoded Akiyo videos are higher than other types.

Table 6.9: Encoder performance measurements of coded Akiyo sequence using different FMO types for Test2

Akiyo	Interleave	Dispersed	Foreground With Leftover	Box-out	Raster Scan	Wipe	Explicit Type 1	Proposed
PSNRY(dB)	39,56	39,47	40,1	40,07	40	40,17	39,31	39,56
Bitrate (kbps)	32,25	32,42	32,35	32,28	32,47	32,39	36,65	36,41

Rates are almost equal to 32 kbps in all cases. However, rate of Explicit types are 4 kbps higher than other types. Periodic retransmission of PPS causes this difference. However sending this information out-of-band can prevent this problem.

6.2.2 Objective Quality

In this section, the quality comparison is done for encoded videos that are transmitted over error prone fading channels. Both of these channels have an average 8 dB SNR. Although both Slow and Fast Fading Channels have the same average channel SNR, results show that videos, which are transmitted over slow fading channels, have better quality than the streams that are transmitted over fast fading channels. Moreover, Tables 6.10 and 6.11 show that Explicit Type 1 has the fewest number of undecoded number of macroblocks. Allocation pattern of important macroblocks can cause this result.

6.2.2.1 Slow Fading Results

The average PSNR values obtained from the slow fading simulations with the Foreman sequence can be seen in Figure 6.11 and Table 6.10.

Table 6.10: Objective Quality Measurements of Foreman sequence for different FMO types when wireless channel is slow fading

Foreman Slow Fading	Interleave	Dispersed	Foreground With Leftover	Box-out	Raster Scan	Wipe	Explicit Type 1	Proposed
Number of Undecoded MB	789	679	669	706	762	750	634	781
Average PSNR (dB)	24,00	27,75	26,64	29,35	24,93	25,89	26,14	28,15

According to Table 6.10, Box-Out FMO type has higher PSNR values than others. Indeed, this result is unexpected, because Explicit Type 1 has the fewest number of undecoded macroblocks, but it can not have the best performance. It is stated in [29] that, error concealment affects the results substantially. Hence, this consequence is possible. The main reason behind the PSNR characteristics of other types is their number of undecoded macroblocks.

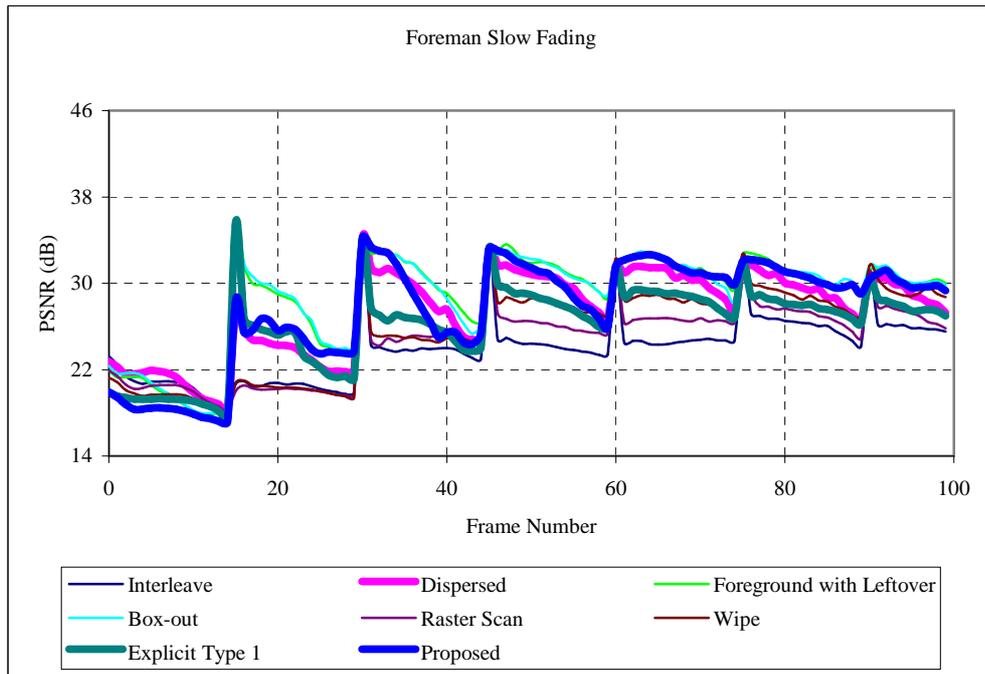


Figure 6.11: PSNR values of different FMO types for Foreman sequence when the channel is slow fading wireless channel

PSNR values of coded Akiyo sequences are illustrated in Table 6.11 and Figure 6.12. Explicit Type 1 has the highest average PSNR value. Two reasons lead this conclusion. First reason is the correlation between consecutive frames. Since important blocks are determined from the order of previous frames' macroblocks, this correlation increases the accuracy of allocation of the macroblocks to the slices. Second reason is the number of undecoded macroblocks. Explicit Type 1 has the fewest number of undecoded macroblocks.

Table 6.11: Objective Quality Measurements of Akiyo sequence for different FMO types when wireless channel is slow fading

Akiyo Slow Fading	Interleave	Dispersed	Foreground With Leftover	Box-out	Raster Scan	Wipe	Explicit Type 1	Proposed
Number of Undecoded MB	913	962	938	983	910	963	799	886
Average PSNR (dB)	33,00	33,32	33,35	31,06	32,16	33,02	34,45	32,70

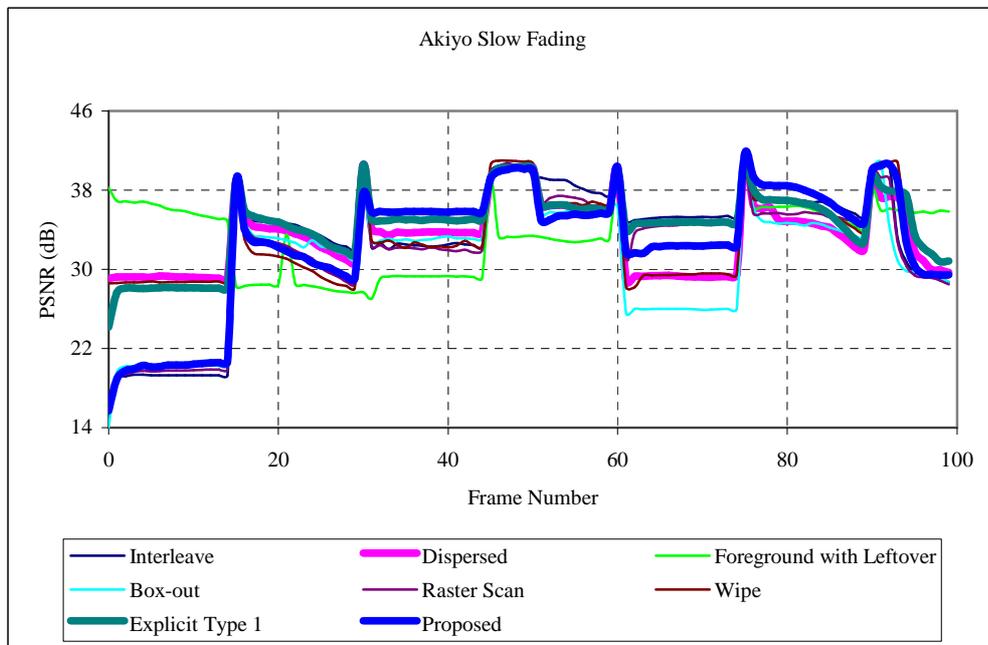


Figure 6.12: PSNR values of different FMO types for Akiyo sequence when the channel is slow fading wireless channel

6.2.2.2 Fast Fading Results

SNR characteristic of Slow Fading channels vary slowly. It causes several numbers of consecutive frames to be corrupted. Hence, decoding can be stopped because these corrupted frames can be used for reconstruction of the following frames. Frame Copy may be a solution for this problem, but it causes drift in the video when the motion activity of the sequence is high. On the other hand, the error characteristic of Fast Fading channels is random. Hence, the risk of burst packet loss drops. Fast fading simulation results of Foreman sequence are illustrated in Table 6.12 and Figure 6.13. According to these illustrations, there are not great differences among the FMO types except Box-Out and Foreground with Leftover types. These two types have poor performance with respect to others.

Table 6.12: Objective Quality Measurements of Foreman sequence for different FMO types when wireless channel is fast fading

Foreman Fast Fading	Interleave	Dispersed	Foreground With Leftover	Box-out	Raster Scan	Wipe	Explicit Type 1	Proposed
Number of Undecoded MB	1287	1277	1247	1219	1397	1321	1201	1383
Average PSNR (dB)	23,09	21,79	20,44	20,19	22,76	22,07	22,79	22,18

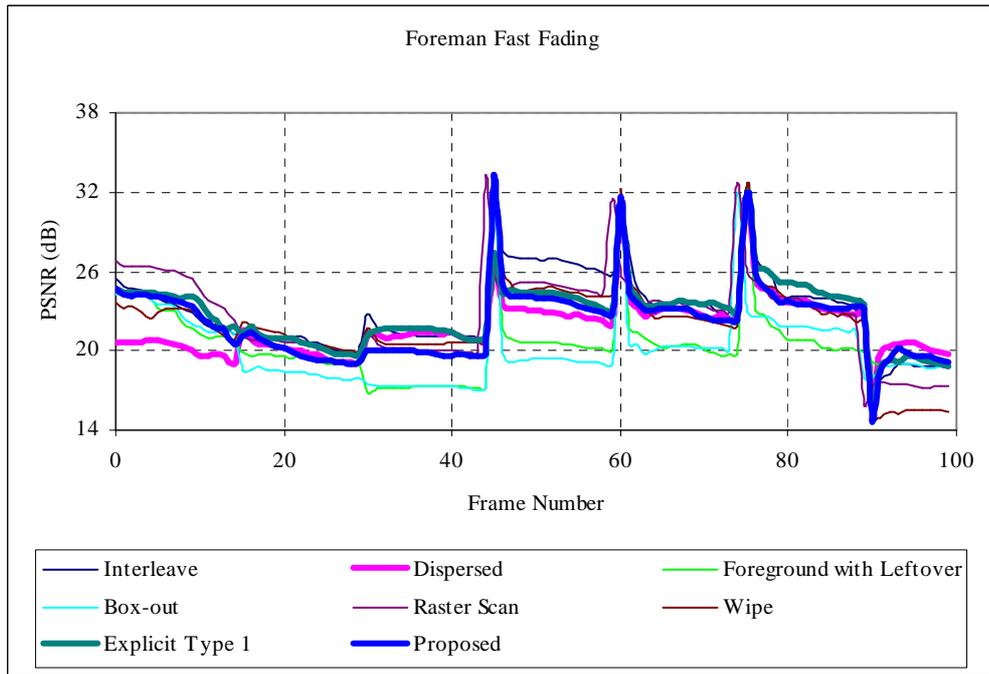


Figure 6.13: PSNR values of different FMO types for Foreman sequence when the channel is fast fading wireless channel

Fast fading simulation results of Akiyo sequence are illustrated in Table 6.13 and Figure 6.14. According to these measurements, Proposed, Explicit Type 1, Dispersed, Interleave perform better than others

Table 6.13: Objective Quality Measurements of Akiyo sequence for different FMO types when wireless channel is fast fading

Akiyo Fast Fading	Interleave	Dispersed	Foreground With Leftover	Box-out	Raster Scan	Wipe	Explicit Type 1	Proposed
Number of Undecoded MB	1452	1406	1651	1431	1415	1515	1298	1512
Average PSNR (dB)	20,98	20,53	17,11	20,06	18,23	18,06	22,05	21,60

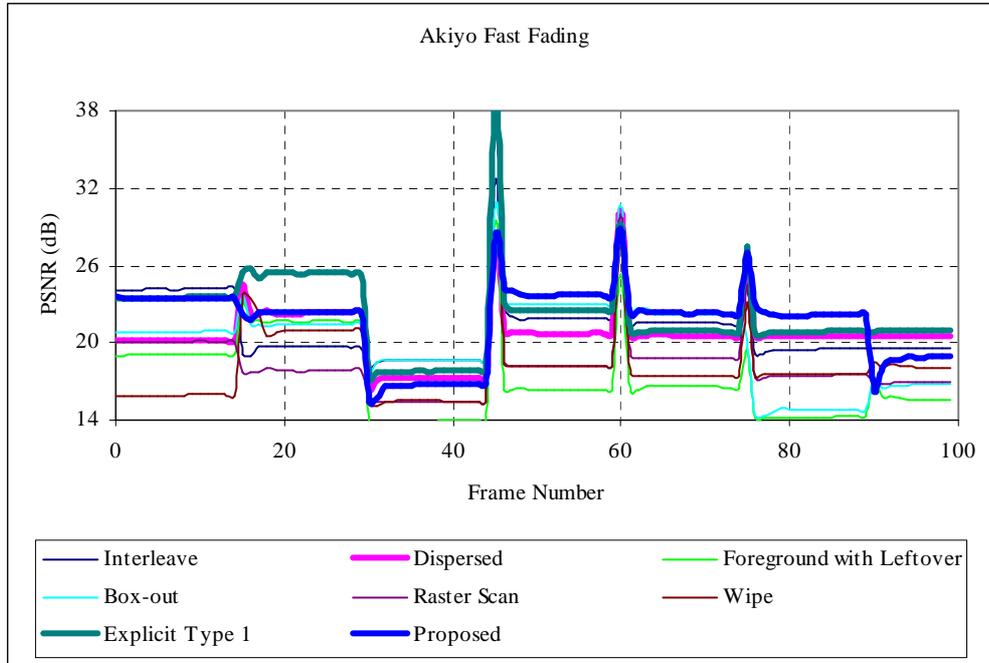


Figure 6.14: PSNR values of different FMO types for Akiyo sequence when the channel is fast fading wireless channel

Macroblock allocation pattern leads this consequence. Since fast fading causes random packet losses and the duration of these losses is generally short, FMO types which separate neighboring macroblocks appropriately have better performance than the types collect these blocks in one slice.

In Figure 6.15 and 6.16 the average PSNR values of different FMO types are illustrated for Foreman and Akiyo sequences. According to the figures Explicit Type 1 performs better than others in most of the cases. Number of undecoded macroblocks causes this consequence.

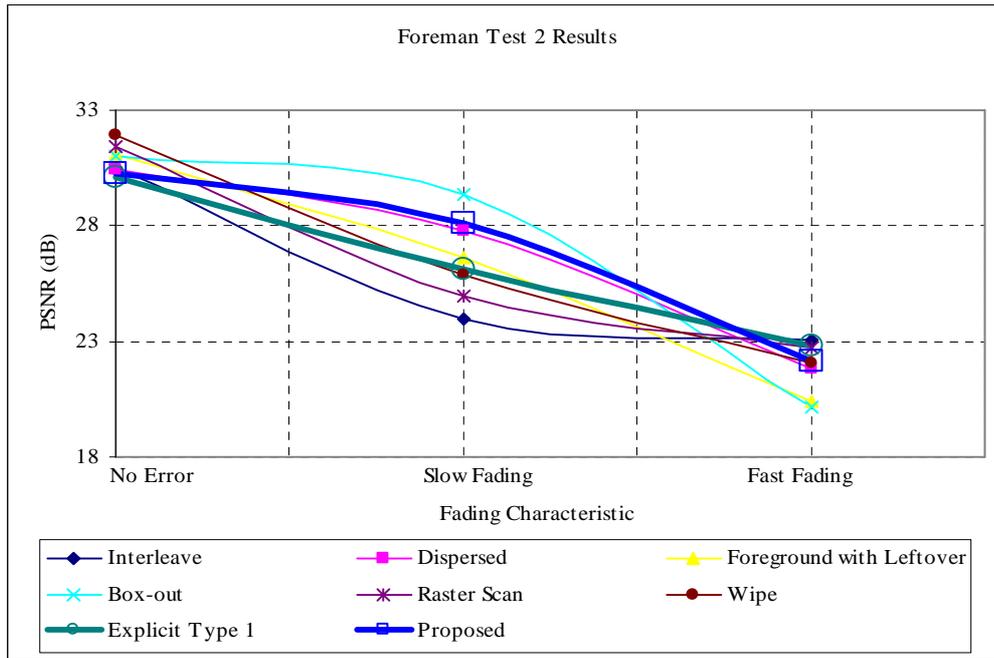


Figure 6.15: Average PSNR values of FMO types for Test 2 Foreman sequence

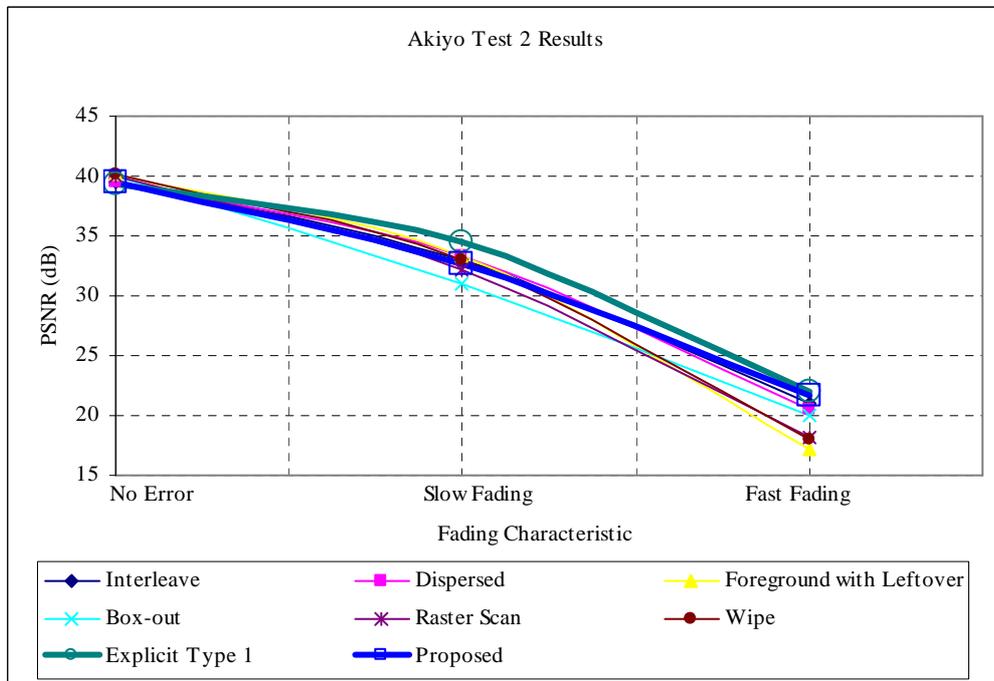


Figure 6.16: Average PSNR values of FMO types for Test 2 Akiyo sequence

6.3 Test 3

6.3.1 Encoder Performance

First of all, the types, which place neighboring macroblocks into the same slice, have the best PSNR values. On the other hand, these values are lower than the PSNR values of Test1, because they are encoded at a smaller bitrate to fulfill the bandwidth requirements. Tables 6.14 illustrate the encoder performance measurements of Foreman sequence.

Table 6.14: Encoder performance measurements of coded Foreman sequence using different FMO types for Test3

Foreman	Interleave	Dispersed	Foreground With Leftover	Box-out	Raster Scan	Wipe	Explicit Type 1	Proposed
PSNRY(dB)	41,17	40,84	41,3	41,31	41,3	41,37	41,04	40,59
Bitrate(kbps)	200,17	200,13	200,09	200,12	200,11	200,1	206,7	191,89

According to Table 6.14, the highest PSNR value belongs to Wipe Type and Proposed Type has the lowest PSNR value. The PSNR results of Akiyo sequence show similar characteristic to the Foreman, but their values are higher. Encoder performance measurements of Akiyo sequence is illustrated in Table 6.15. Secondly, the bitrate values are approximately equal except the bitrate of Proposed Type. EEP is used for other types which is RS (6, 4). On the other hand, UEP is employed for Proposed Algorithm. RS (8, 4) and RS (5, 4) are used for protecting important slices and the other slices respectively. Since important slices contain more bytes than others, the total number of bytes after adding FEC may require higher bitrate. Therefore, it is encoded with quantization parameter that satisfies bandwidth constraints.

Table 6.15: Encoder performance measurements of coded Akiyo sequence using different FMO types for Test3

Akiyo	Interleave	Dispersed	Foreground With Leftover	Box-out	Raster Scan	Wipe	Explicit Type 1	Proposed
PSNRY(dB)	42,92	42,31	43,31	43,21	43,05	43,09	42,5	41,94
Bitrate (kbps)	60,15	60,08	60,08	60,11	60,17	60,17	66,39	59,38

6.3.2 Objective Quality

The objective quality tests are carried out for the packet loss rates of %0, %3, %5, %10, %20 and % 30. Simulations show that there is not any corruption, if error percentage is %3. Hence average PSNR values of the reconstructed videos which are simulated using %0 and %3 error patterns are same.

6.3.2.1 %3 Packet Loss Results

When packet loss rate of the channel is %3, all the packets are successfully transmitted according to results of Foreman and Akiyo sequences except Proposed Type. According to Table 6.16, Wipe Type and Foreground with Left over have the highest PSNR values for Foreman and Akiyo respectively. On the other hand, Proposed Type has the lowest PSNR value in both cases. This is caused as a result of UEP and bandwidth constraints

Table 6.16: Objective Quality Measurements of Foreman and Akiyo sequences for different FMO Types when packet loss rate is %3 and FEC is employed

%3		Interleave	Dispersed	Foreground With Leftover	Box-out	Raster Scan	Wipe	Explicit Type 1	Proposed
Foreman	Number of Undecoded MB	0	0	0	0	0	0	0	925
	Average PSNR(dB)	41,17	40,84	41,3	41,31	41,3	41,37	41,04	38,35
Akiyo	Number of Undecoded MB	0	0	0	0	0	0	0	0
	Average PSNR(dB)	42,92	42,31	43,31	43,21	43,05	43,09	42,5	41,94

Figure 6.17 shows the PSNR values Foreman sequences that are coded using different FMO types. The error of the proposed type can be easily detected from the figure. The PSNR values of the Akiyo sequence is illustrated in Figure 6.18. The types which have the best and worst PSNR values can be clearly seen from the figure.

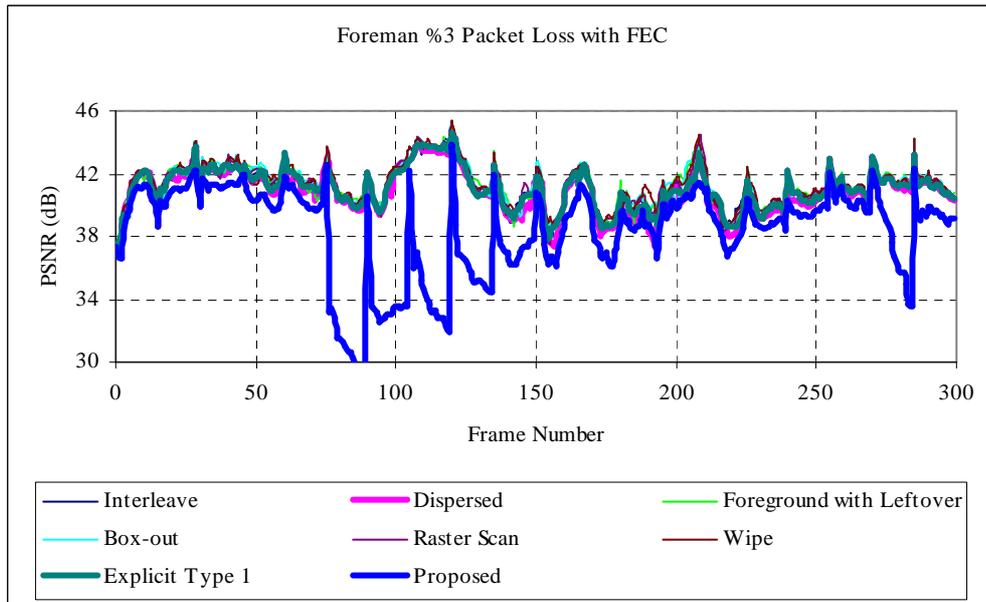


Figure 6.17: PSNR values of different FMO types for Foreman sequence when packet loss rate is %3 and FEC is employed

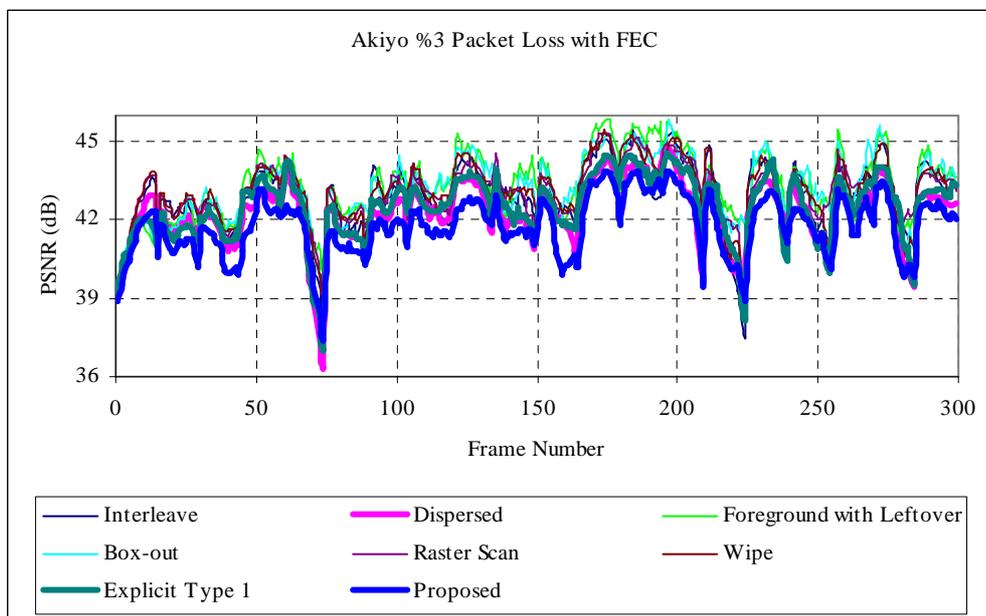


Figure 6.18: PSNR values of different FMO types for Akiyo sequence when packet loss rate is %3 and FEC is employed

6.3.2.2 %5 Packet Loss Results

When the packet loss rate increases to %5, it can be observed that from Table 6.17 that, Dispersed Type has the best PSNR characteristics for Foreman sequence, whereas Foreground with Leftover has the highest PSNR value for Akiyo sequence. Error concealment is the main reason behind the success of Dispersed type in the first case. On the other hand, number of undecoded macroblocks can explain why Foreground with Leftover has such a high average PSNR value. Raster Scan and Wipe types have the lowest PSNR values. Moreover Interleave and Raster Scan types have the fewest number of undecoded macroblocks, whereas Proposed Type has the highest number. UEP causes this result because the macroblocks of the background region is protected less than others

Table 6.17 Objective Quality Measurements of Foreman and Akiyo sequences for different FMO Types when packet loss rate is %5 and FEC is employed

%5		Interleave	Dispersed	Foreground With Leftover	Box-out	Raster Scan	Wipe	Explicit Type 1	Proposed
Foreman	Number of Undecoded MB	325	485	425	450	420	500	400	1125
	Average PSNR(dB)	39,17	39,11	39,1	37,89	37,49	37,5	38,6	38,16
Akiyo	Number of Undecoded MB	586	527	471	500	450	513	525	825
	Average PSNR(dB)	32,93	37,13	39,35	36,33	32,73	30,95	34,92	39,17

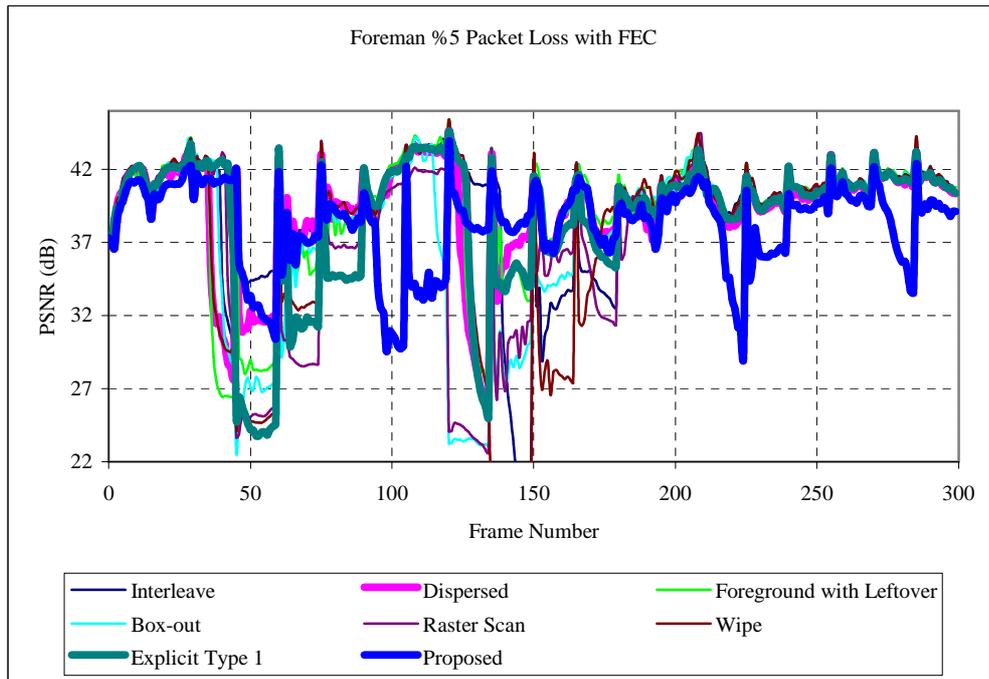


Figure 6.19: PSNR values of different FMO types for Foreman sequence when packet loss rate is %5 and FEC is employed

Figure 6.19 illustrates the PSNR values of Foreman sequences which are coded using different FMO types. The difference between Dispersed Type and Wipe Type can be seen from the figure. According to Figure 6.20, Proposed Type has better PSNR than Dispersed and Explicit Type 1 algorithms. This consequence can be caused by UEP. That is, the macroblocks which are lost during transmission are mostly unimportant macroblocks.

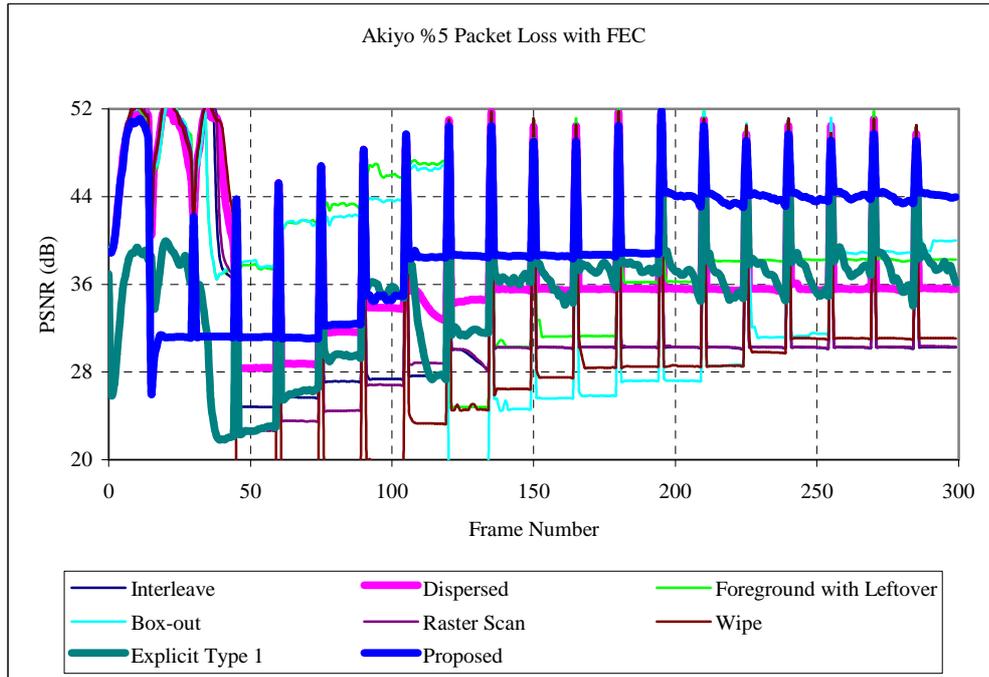


Figure 6.20: PSNR values of different FMO types for Akiyo sequence when packet loss rate is %5 and FEC is employed

6.3.2.3 %10 Packet Loss Results

As can be seen from Table 6.18, Dispersed Type shows better performance than other types when the packet loss rate is %10. The type which shows worst performance changes according to video sample. Since dispersed type has a structure that is appropriate for using neighboring macroblocks, it is more robust than others.

Table 6.18 Objective Quality Measurements of Foreman and Akiyo sequences for different FMO Types when packet loss rate is %10 and FEC is employed

%10		Interleave	Dispersed	Foreground With Leftover	Box-out	Raster Scan	Wipe	Explicit Type 1	Proposed
Foreman	Number of Undecoded MB	834	863	850	865	1003	752	834	1570
	Average PSNR(dB)	33,42	35,06	34,42	34,67	32,92	34,29	34,93	34,62
Akiyo	Number of Undecoded MB	776	514?	414	635	800	569	736	875
	Average PSNR(dB)	29,93	36,92	31,59	35,73	29,76	31,30	31,30	33,39

According to Table 6.18, Explicit Type 1 follows Dispersed Type when the video sample is Foreman. Since Explicit Type 1 has a structure that interleaves macroblocks that has high motion activity or used for prediction, the received packets always contain important information. On the other hand, Raster Scan does not have a high PSNR values because it can not use the error concealment efficiently and it can not separate macroblocks according to their contribution to the quality of reconstructed video. The PSNR values of Foreman and Akiyo sequences that are coded using different FMO types are illustrated in Figures 6.21 and 6.22 respectively.

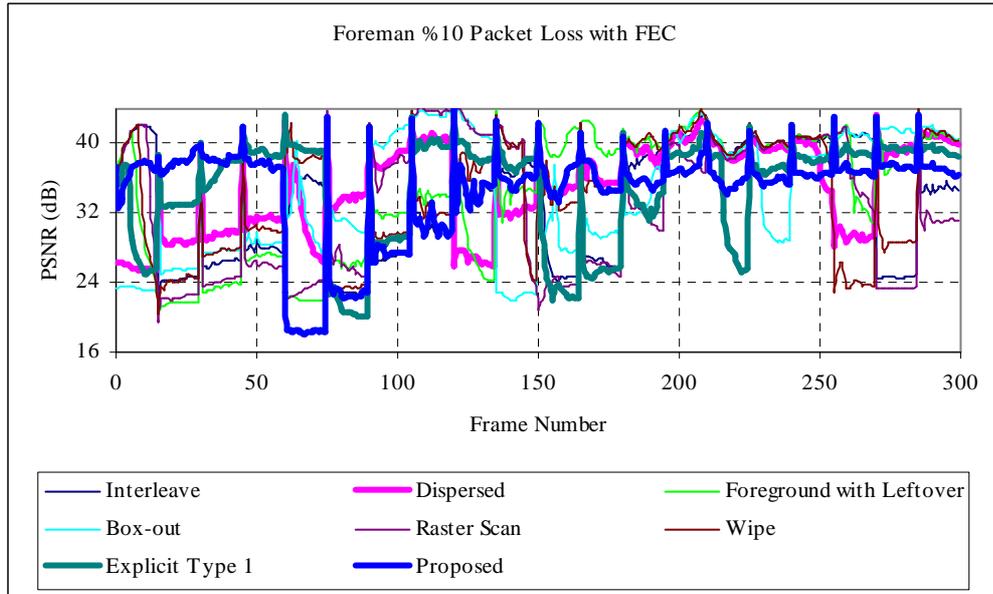


Figure 6.21: PSNR values of different FMO types for Foreman sequence when packet loss rate is %10 and FEC is employed

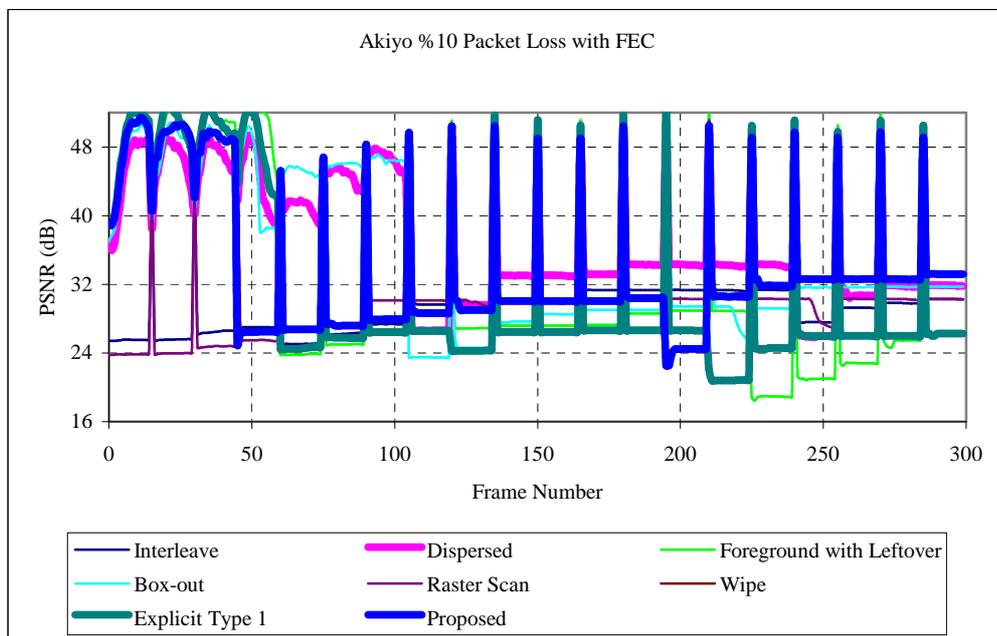


Figure 6.22: PSNR values of different FMO types for Akiyo sequence when packet loss rate is %10 and FEC is employed

6.3.2.4 %20 Packet Loss Results

When the performances of types are compared for %20 packet loss rate, it can be observed that Dispersed, Explicit Type 1 and Proposed types have better PSNR values than others. According to Table 6.19, Dispersed type and Explicit Type 1 have the best performance for Foreman and Akiyo sequences respectively.

Table 6.19 Objective Quality Measurements of Foreman and Akiyo sequences for different FMO Types when packet loss rate is %20 and FEC is employed

%20		Interleave	Dispersed	Foreground With Leftover	Box-out	Raster Scan	Wipe	Explicit Type 1	Proposed
Foreman	Number of Undecoded MB	2318	2621	2425	2593	2553	2938	2647	4499
	Average PSNR(dB)	30,22	31,94	27,71	27,63	30,02	27,55	28,46	30,42
Akiyo	Number of Undecoded MB	2520	2447	2661	2309	2703	2457	2943	3224
	Average PSNR(dB)	26,28	30,64	25,27	25,41	24,28	20,97	30,8	26,1

On the other hand, Wipe Types has the lowest PSNR values in both cases Allocation of adjacent macroblocks causes this consequence. Besides that, as the packet error rate increases the probability of losing consecutive slices also increases. The success of Proposed Type is affected from number of undecoded macroblocks

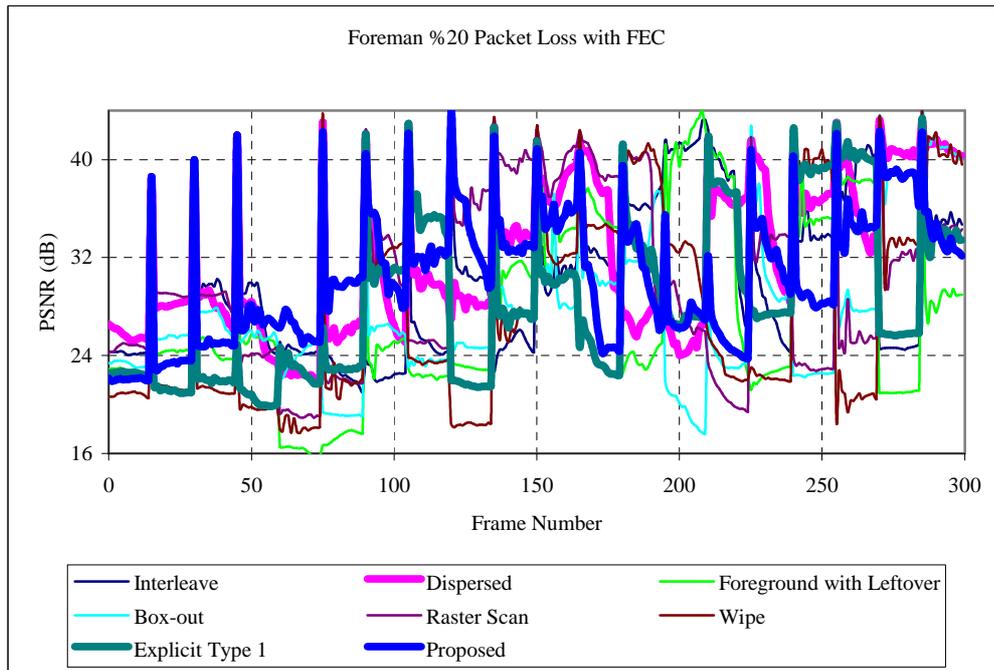


Figure 6.23: PSNR values of different FMO types for Foreman sequence when packet loss rate is %20 and FEC is employed

The success of Dispersed type can also be seen from Figure 6.23 which illustrates the PSNR values of Foreman sequences which are encoded using different FMO types. According to Figure 6.24, Explicit Type 1 and Dispersed Type have better PSNR characteristics than Proposed Algorithm.

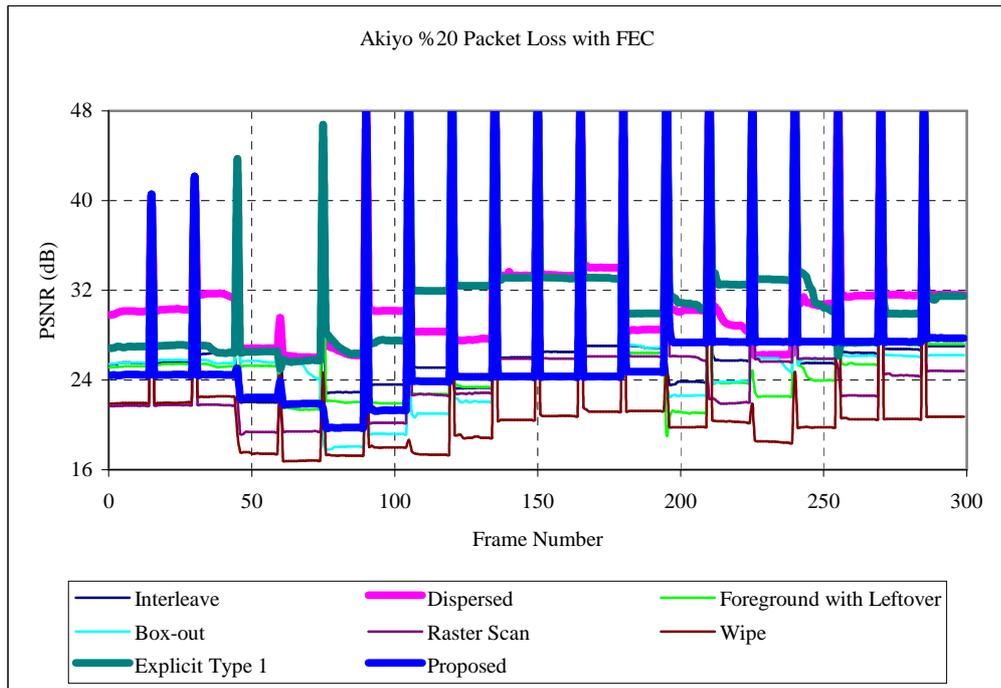


Figure 6.24: PSNR values of different FMO types for Akiyo sequence when packet loss rate is %20 and FEC is employed

6.3.2.5 %30 Packet Loss Results

The results in Table 6.20 show that Proposed Type outperforms the others for both sequences. Dispersed Type and Proposed Type follow the proposed algorithm. Two important factors affect the result. The first reason is protecting important macroblocks with a higher error protection. Proposed Type has the least number of undecoded macroblocks. Secondly, important macroblocks are separated according to a checkerboard pattern. On the other hand, Wipe and Foreground with Leftover Types do not have sufficient quality.

Table 6.20: Objective Quality Measurements of Foreman and Akiyo sequences for different FMO Types when packet loss rate is %20 and FEC is employed

%30		Interleave	Dispersed	Foreground With Leftover	Box-out	Raster Scan	Wipe	Explicit Type 1	Proposed
Foreman	Number of Undecoded MB	9725	10681	max	10138	10570	10930	10505	8852
	Average PSNR(dB)	20,91	23,21	18,58	18,76	19,46	18,86	21,27	24,87
Akiyo	Number of Undecoded MB	10267	9415	10659	9831	9270	9867	10521	5877
	Average PSNR(dB)	21,53	23,84	18,33	17,55	19,27	17,42	24,48	24,73

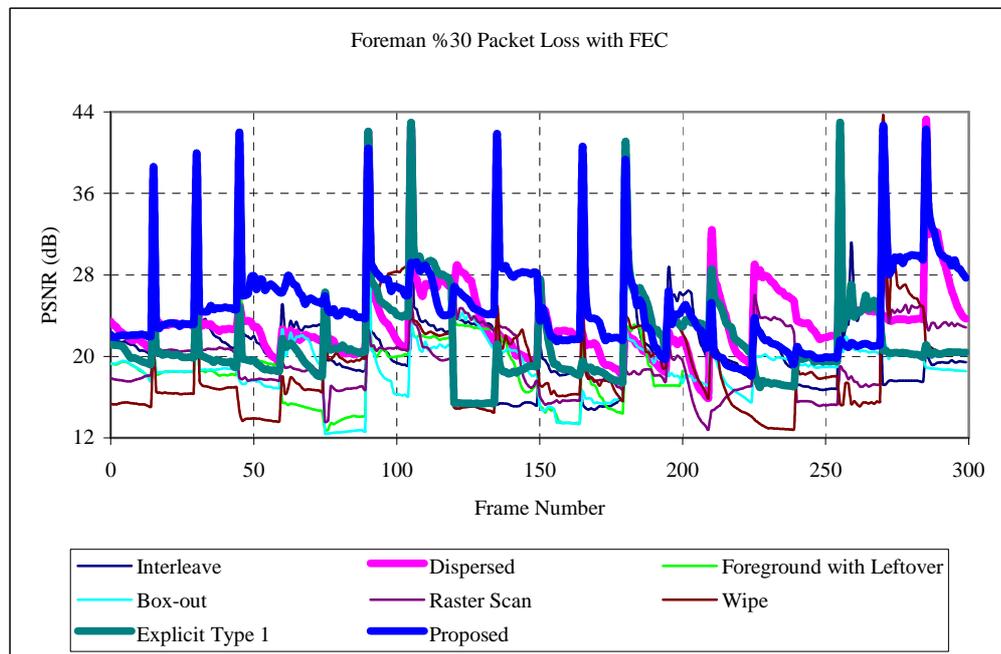


Figure 6.25: PSNR values of different FMO types for Foreman sequence when packet loss rate is %30 and FEC is employed

The success of Proposed Type can be explicitly in both Figures 6.25 and 6.26.

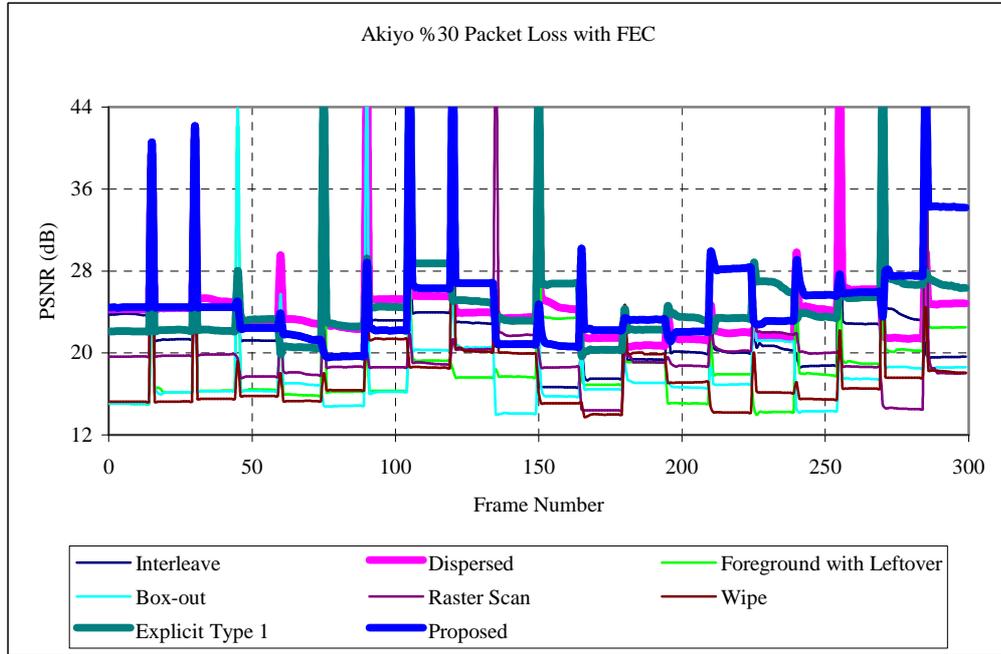


Figure 6.26: PSNR values of different FMO types for Akiyo sequence when packet loss rate is %30 and FEC is employed

In Figure 6.27 and 6.28 the average PSNR values of different FMO types are illustrated for Foreman and Akiyo sequences. The results of Test 3 are similar to the results of Test 1. When the packet loss rate is low Wipe and Foreground with Leftover have the highest PSNR values for Foreman and Akiyo sequences respectively. As the packet loss rate increases Dispersed and Explicit Type 1 perform better than others. Lastly, when the packet loss rate is %30 Proposed type outperforms others.

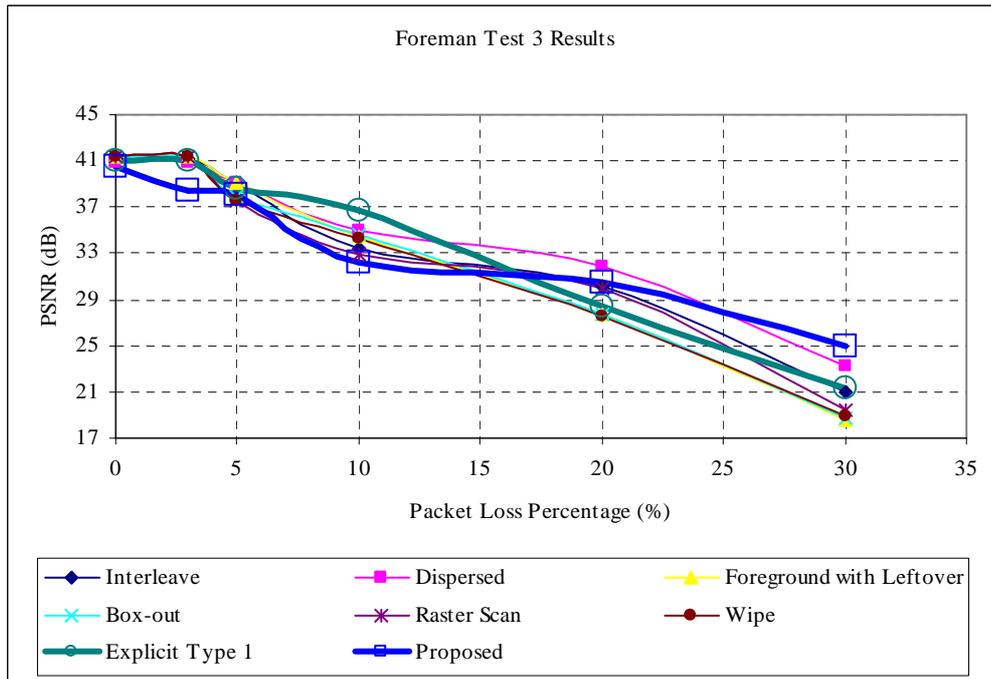


Figure 6.27: Average PSNR values of FMO types for Test 3 Foreman sequence

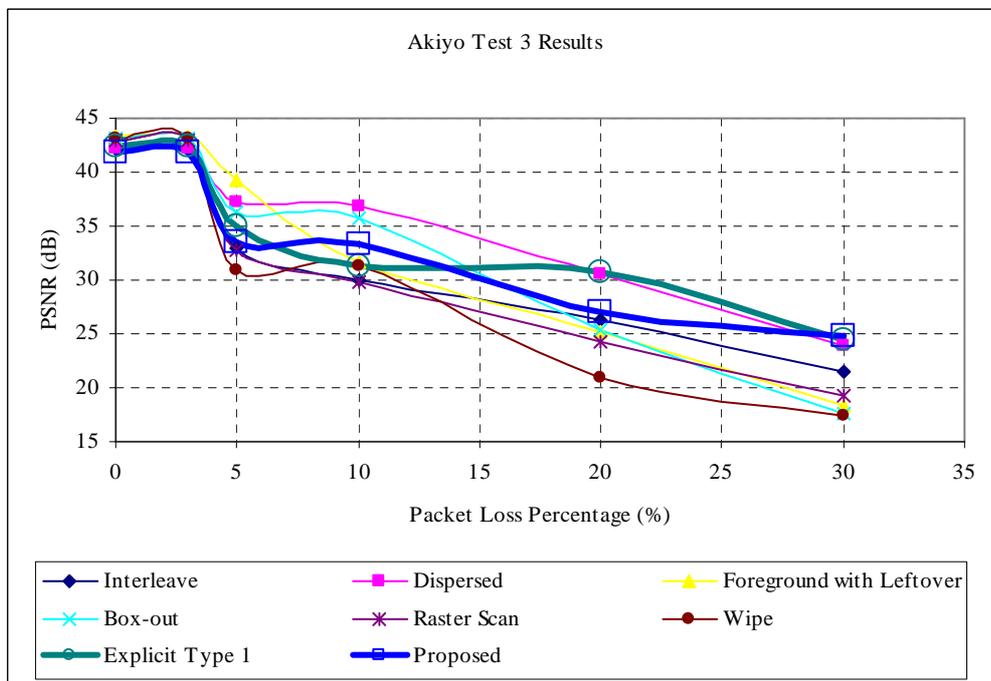


Figure 6.28: Average PSNR values of FMO types for Test 3 Akiyo sequence

6.4 Test4

6.4.1 Encoder Performance

When the PSNR values of the coded videos are compared, it can be observed that slices that contain adjacent macroblocks have better PSNR performances like in the previous cases. Slices, which contain neighboring macroblocks, are coded more efficiently. Therefore, they can be quantized with a better quantization parameter. According to Table 6.21 and 6.22, Foreground with Leftover has the highest PSNR value. Moreover, Proposed type has the worst PSNR value. Applying UEP causes this consequence. As can be seen from Tables 6.21 to 6.22, necessary bitrates of videos that are coded using Explicit Types are higher than others. This is due to the periodic retransmission of PPS packets which carry dynamically changing MBAMap patterns.

Table 6.21: Encoder performance measurements of coded Foreman sequences using different FMO types for Test4

Foreman	Interleave	Dispersed	Foreground With Leftover	Box-out	Raster Scan	Wipe	Explicit Type 1	Proposed
PSNRY(dB)	27,89	27,88	28,2	28,16	27,94	27,86	28,06	27,48
Bitrate (kbps)	23,04	23,55	24,35	24,57	23,24	22,24	27,37	26,5

Table 6.22: Encoder performance measurements of coded Akiyo sequences using different FMO types for Test4

Akiyo	Interleave	Dispersed	Foreground With Leftover	Box-out	Raster Scan	Wipe	Explicit Type 1	Proposed
PSNRY(dB)	35,89	35,27	36,57	36,57	36,21	36,21	35,35	34,79
Bitrate (kbps)	20,55	21,07	21,25	21,45	21,54	21,03	24,5	23,69

6.4.2 Objective Quality

The simulations are done for Slow and Fast Fading Channels like the simulations of Test 2.

6.4.2.1 Slow Fading Results

Tables 6.23 and 6.24 show that, Proposed Type has the highest PSNR value among all FMO types. For instance, the PSNR value of decoded Foreman sequence, which is coded using Proposed Type, is at least 1.3 dB better than the others. The primary reason of this consequence is using UEP for Proposed Type. Although high protection payload affects the quantization parameter and the PSNR value of the encoded video substantially, this error protection structure protects more packets than others when the packet error loss rate is high.

Table 6.23: Objective Quality Measurements of Foreman sequence for different FMO types when wireless channel is slow fading and FEC is employed

Foreman Slow Fading With FEC	Interleave	Dispersed	Foreground With Leftover	Box-out	Raster Scan	Wipe	Explicit Type 1	Proposed
Number of Undecoded MB	1299	1447	1220	1138	1224	879	1129	127
Average PSNR (dB)	21,00	23,59	24,44	22,64	24,51	20,77	23,00	25,88

Moreover, using dispersed pattern for important macroblocks also enhances the PSNR since error concealment can be used more efficiently. Lastly, important macroblocks are generally the macroblocks which can not be predicted easily. Hence probability of error propagation decreases.

Table 6.24: Objective Quality Measurements of Akiyo sequence for different FMO types when wireless channel is slow fading and FEC is employed

Akiyo Slow Fading With FEC	Interleave	Dispersed	Foreground With Leftover	Box-out	Raster Scan	Wipe	Explicit Type 1	Proposed
Number of Undecoded MB	1335	843	863	848	1070	1342	1095	151
Average PSNR (dB)	29,44	29,82	27,96	29,34	30,44	32,47	33,05	34,13

When the results are examined, it can also be seen that, Proposed Type has

the lowest number of undecoded of macroblocks.

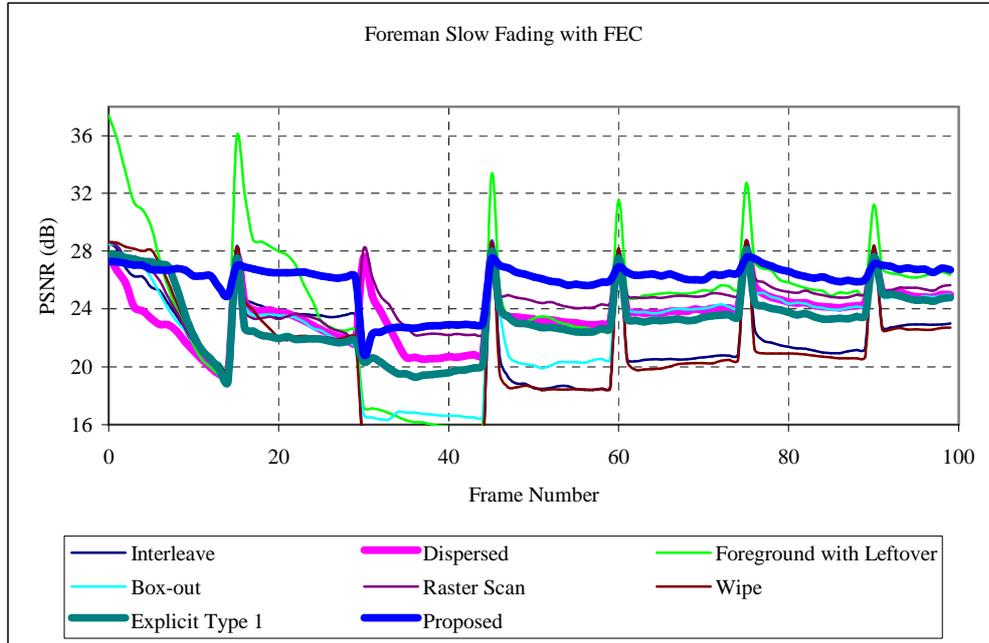


Figure 6.29: PSNR values of different FMO types for Foreman sequence when wireless channel is slow fading and FEC is employed

The types, which have the lowest PSNR values, are Wipe Type for Foreman sequence and Foreground with Leftover Type for Akiyo. Figures 6.29 and 6.30 illustrate the PSNR values of different FMO types when the sequences are Foreman and Akiyo respectively. Since types like Wipe and Foreground with Leftover distribute neighboring macroblocks to same slices, concealment can not be applied efficiently.

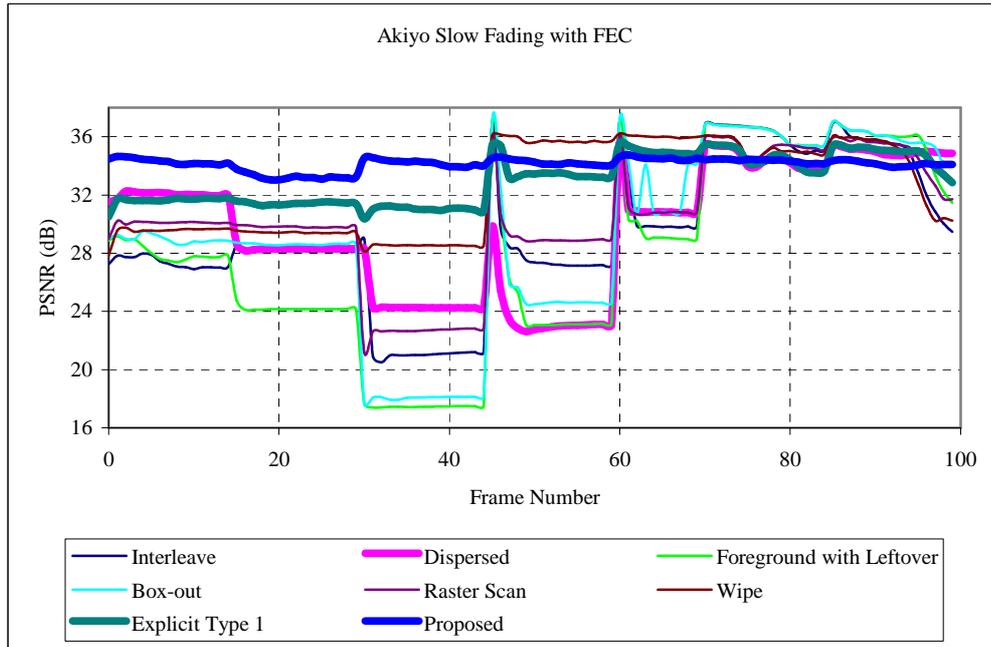


Figure 6.30: PSNR values of different FMO types for Akiyo sequence when wireless channel is slow fading and FEC is employed

6.4.2.2 Fast Fading Results

When the results of simulations which are performed for Fast Fading case are observed, it can be found that, number of undecoded macroblocks and the motion complexity of the video samples affect the results. Table 6.25 illustrates that, Proposed Type has the best PSNR value for the Foreman sequence. This is mainly because of undecoded macroblocks and protection of important macroblocks.

Table 6.25: Objective Quality Measurements of Foreman sequence for different FMO types when wireless channel is fast fading and FEC is employed

Foreman Fast Fading With FEC	Interleave	Dispersed	Foreground With Leftover	Box-out	Raster Scan	Wipe	Explicit Type 1	Proposed
Number of Undecoded MB	912	579	805	654	616	716	632	231
Average PSNR (dB)	23,01	24,28	25,40	23,58	24,18	23,54	24,80	26,05

Figure 6.31 illustrates the PSNR values different FMO Types when Foreman sequence is used. It can be explicitly seen from the figure that, Proposed Type keeps its PSNR value almost same.

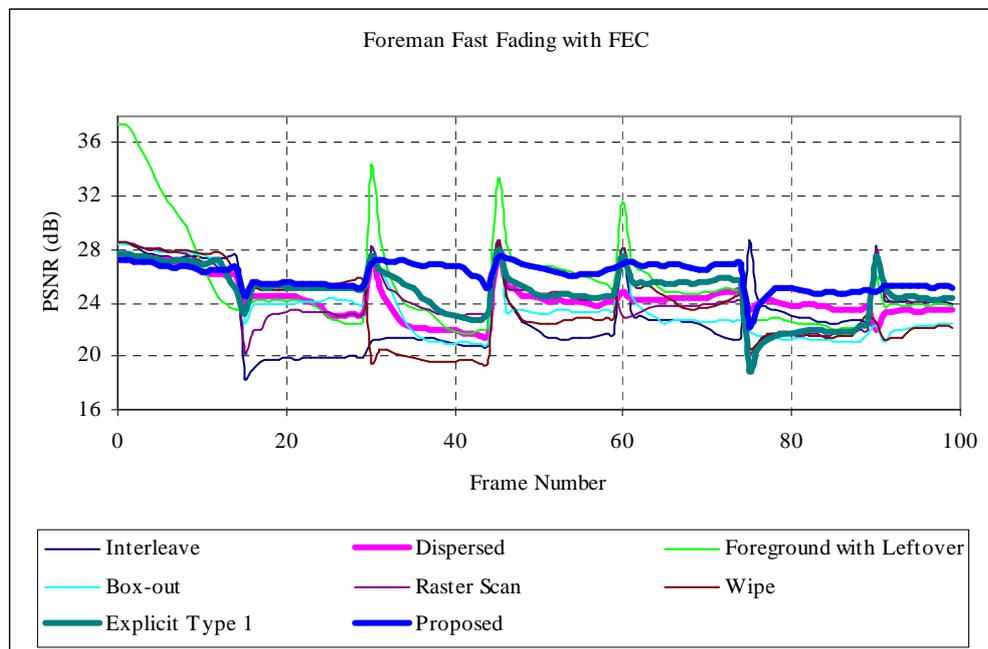


Figure 6.31: PSNR values of different FMO types for Foreman sequence when wireless channel is fast fading and FEC is employed

Table 6.26: Objective Quality Measurements of Akiyo sequence for different FMO types when wireless channel is fast fading and FEC is employed

Akiyo Fast Fading With FEC	Interleave	Dispersed	Foreground With Leftover	Box-out	Raster Scan	Wipe	Explicit Type 1	Proposed
Number of Undecoded MB	819	572	425	411	648	715	595	180
Average PSNR (dB)	34,08	33,90	33,27	33,83	34,24	32,12	33,61	33,99

Measurements of Akiyo sequence, which are found in Table 6.26, show that Raster Scan has the highest PSNR value. Interleave, Dispersed and Proposed Types follow Raster Scan.

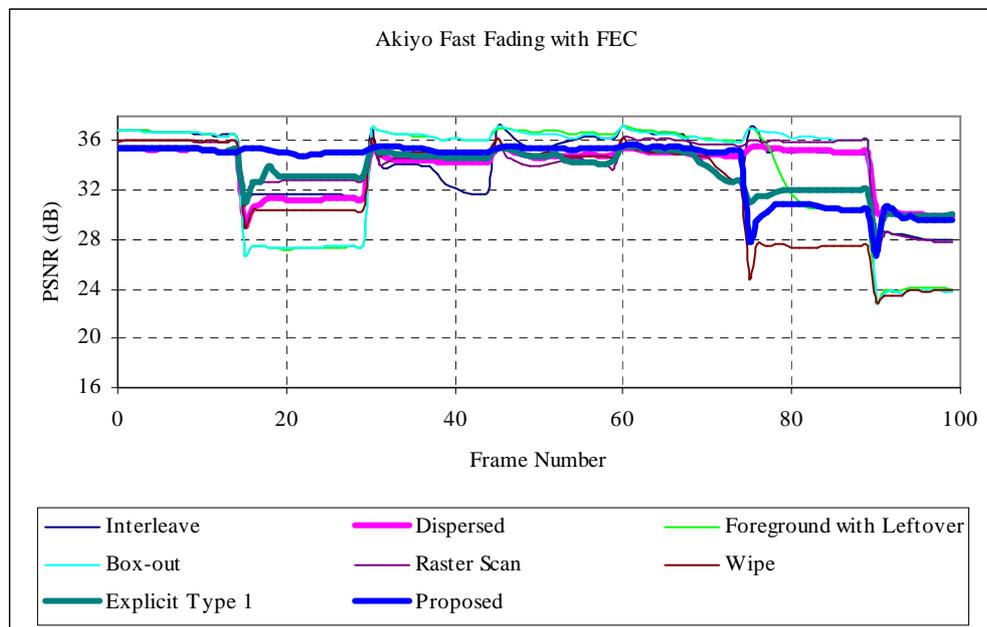


Figure 6.32: PSNR values of different FMO types for Akiyo sequence when wireless channel is fast fading and FEC is employed

Figure 6.32 illustrates the PSNR values of different FMO types when Akiyo sequence is used. Hence, main parameters, which determine the quality in this simulation, are the quantization parameter and the spatial concealment characteristics of the FMO type.

In Figure 6.33 and Figure 6.34 the average PSNR values of different FMO types are illustrated for Foreman and Akiyo sequences. As can be seen from the figures Proposed algorithm outperforms other FMO types for both sequences. Applying Unequal error protection and allocates important macroblocks using a checkerboard pattern cause this consequence

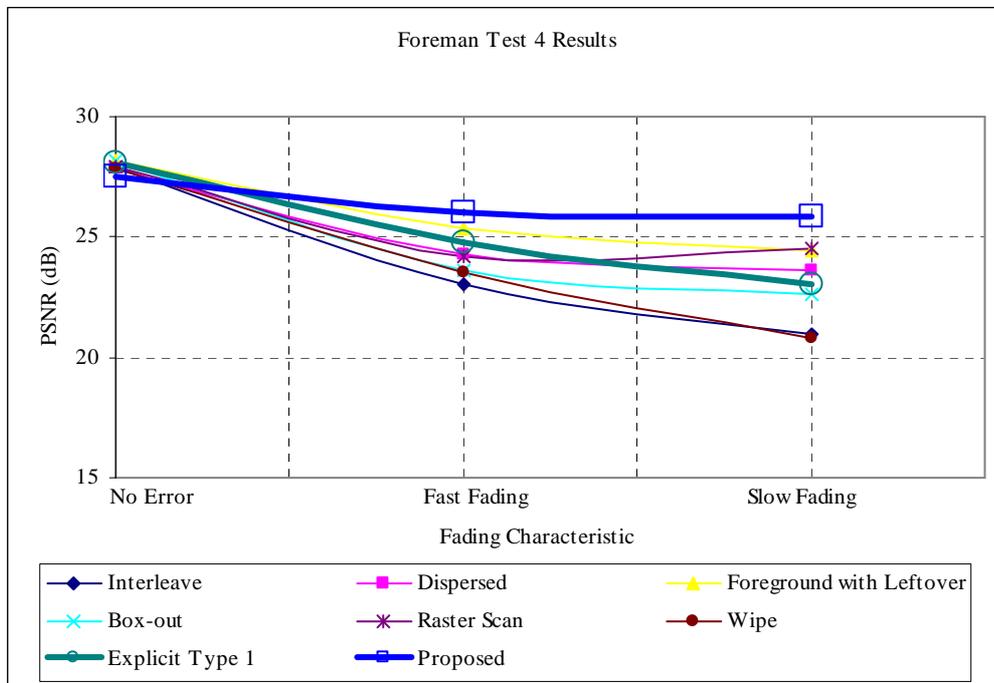


Figure 6.33: Average PSNR values of FMO types for Test 4 Foreman sequence

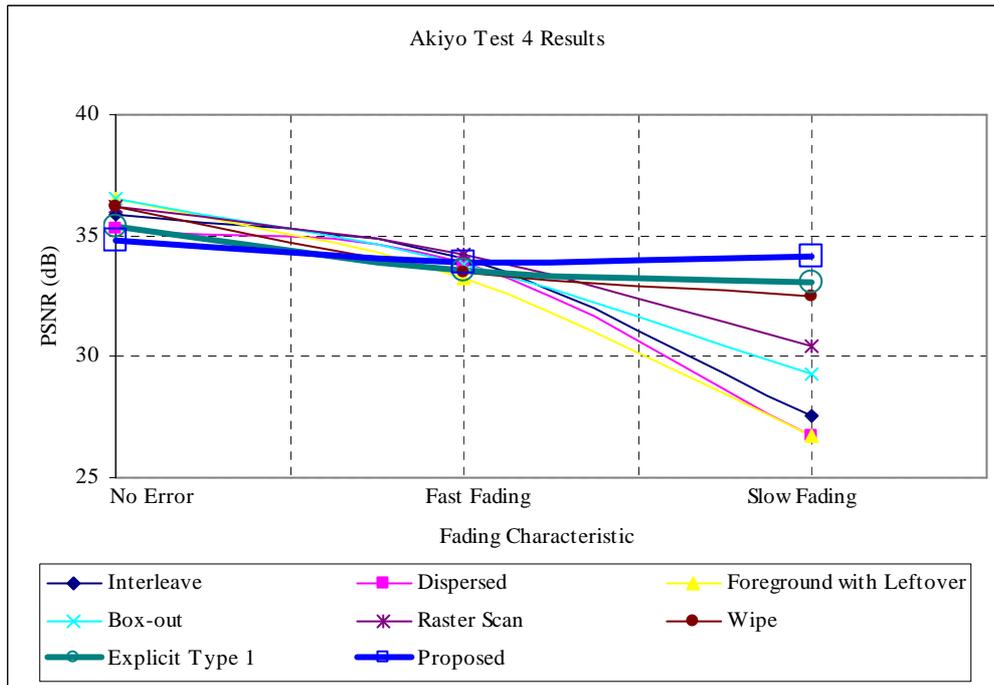


Figure 6.34: Average PSNR values of FMO types for Test 4 Akiyo sequence

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

In this thesis a new FMO algorithm is proposed and this algorithm is compared with the other FMO algorithms. The results of the first stage show that when the error percentage is low, Wipe and Foreground with Leftover algorithms have the best PSNR values. However, as the packet loss increases Dispersed Type becomes better than others. Interleave, Explicit Type 1 and Proposed types follow Dispersed type. Proposed Algorithm is not as successful as Dispersed algorithm because spatial error concealment can not be employed efficiently in the background region. Explicit Type 1 has higher PSNR values than others in the second simulation stage. This is mainly caused by the number of undecoded macroblocks. Moreover distributing important macroblocks to slices equally also increases robustness. The consequences of the third simulation stage illustrates that the best FMO algorithm changes according to the packet loss rate. When the rate is low, Wipe and Raster Scan have better quality than others. As the packet loss rate increases from %5 to %20 Interleave, Dispersed and Explicit Type 1 have higher PSNR values than others. Lastly, Proposed algorithm outperforms other types, when the packet loss rate is %30. Its PSNR value is at least 1,3 dB higher than others for Foreman sequence. In the last simulation stage, Proposed type has the best PSNR values when the wireless channel is slow fading. Using scattered pattern to allocate important macroblocks and protecting them with higher rate lead this consequence. In the fast fading case, Proposed and Raster Scan have the best objective quality values for Foreman and Akiyo sequences respectively. Since errors occur at random intervals in fast fading wireless channels, EEP saves more bits. Moreover, proposed type is encoded using a coarse quantization parameter. Hence the

success of Proposed type is mostly caused by allocation of macroblocks. According to these results, number of undecoded macroblocks, importance of these macroblocks, quantization parameter of the encoded stream and error concealment characteristic of the FMO type affect the PSNR value of the video which is transmitted over an error prone communication channel. When the necessary bitrates are compared, it can be easily realized that Explicit Types require more than others. The number of transmitted PPS causes this consequence. Since MBAMap changes dynamically, it has to be sent to the receiver periodically in PPS. Hence, the necessary bandwidth increases substantially. In order to decrease required bitrate, PPS, which carries similar MBAMap structure with the former PPSs, may not be transmitted. PPS can also be transmitted out of band using a reliable channel. These observations show that a cross layer approach can be employed in the following studies. The FMO type changes according to the error characteristics of the channel. When error rate is low, types like Wipe and Raster Scan are used. At very high error rates Proposed algorithm can be employed with UEP. Moreover, the efficiency of the Proposed algorithm can be analyzed in a wireless network.

REFERENCES

- [1] ITU-T Recommendation H.264, "Advanced Video Coding for generic audiovisual services", November 2007.
- [2] Y. Wang, S. Wenger, Jiangtao Wen and Aggelos K. Katsaggelos, "Error Resilient Video Coding Techniques", IEEE Signal Processing Magazine, July 2000.
- [3] Y. Wang, J. Ostermann and Y. Zhang, "Video Processing and Communications" Prentice Hall, 2002.
- [4] S. Wenger, "H.264/AVC Over IP", IEEE Trans. On Circuits and Systems for Video Technology, Vol. 13, No. 7, pp. 645-656, July 2003.
- [5] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Recommendation H.264, ISO/IEC 14496-10)", JVT G-050, March 2003.
- [6] T. Wiegand, G. Sullivan, G. Bjøntegaard and A. Luthra, "Overview of the H.264/AVC Video Coding Standard", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 7, pp. 560-576, July 2003.
- [7] Iain E.G. Richardson, "H.264 and MPEG-4 Video Compression", UK: Wiley & Sons, 2003
- [8] G. Côté and F. Kossentini "Optimal intra coding of blocks for robust video communication over the internet," EUROSIP J. Image Commun.-Special Issue on Real-time Video Over the Internet, pp. 25-34, Sept. 1999.
- [9] T. Stockhammer, M. Hannuksela and T. Wiegand, "H.264/AVC in wireless environments," IEEE Trans. Circuits and Syst. Video Technol., Vol. 13, pp. 657-673, July 2003.
- [10] V. Varsa, M. M. Hannuksela, Y.-K. Wang, "Non-Normative Error Concealment Algorithms," ITU-T VCEG-N62, 2001.
- [11] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, "FMO: Flexible Macroblock Ordering", JVT-C089, May 2002
- [12] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, "Scattered Slices: A New Error Resilience Tool for H.26L", JVT-B027, January 2002
- [13] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, "Scattered Slices: Simulation Results", JVT-C090, May 2002

- [14]** Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, "Slice Interleaving: Simulation Results", JVT-C091, May 2002
- [15]** Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, "Enhancements to FMO", JVT-D095, July 2002
- [16]** Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, "On Random Access", JVT-D097, July 2002
- [15]** S. Benierbah and M. Khamadja, "A new technique for Quality Scalable Video Coding with H.264", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 15, No 11, pp. 1332-1339, November 2005
- [16]** S. K. Im and A. J. Pearmain, "An optimized mapping algorithm for classified video transmission with the H.264 flexible macroblock ordering", IEEE ICME 2006, pp. 1445-1448
- [17]** Z. Wu and J. M. Boyce, "Optimal frame selection for H.264/AVC FMO coding", IEEE ICIP 2006
- [18]** Y. Dhondt, P. Lambert and R. V. Walle, "A flexible macroblock scheme for unequal error protection", IEEE ICIP 2006
- [19]** N. Thomos, S. Argyropoulos, N. V. Boulgouris and M. G. Strintzis, "Error resilient transmission of H.264/AVC streams using flexible macroblock ordering" partially approved FP6-511568 3DTV
- [20]** P. Baccichet, S. Rane, A. Chimienti and B. Girod, "Robust low-delay video transmission using H.264/AVC redundant slices and flexible macroblock ordering", ICIP, pp. 93-96, 2007.
- [21]** Z. Y. Lin and C. C. Ho, "Novel H.264 /AVC flexible macroblock ordering based on wavelet-domain partition", ISCIT, pp 451-454 October 2007
- [22]** J. Wang, S. Li, K. Shimizu, T. Ikenaga and S. Goto, "Unequal error protected transmission with dynamic classification in H.264/AVC", ASIC, pp 798-801, October 2007
- [23]** H. K. Arachchi, W.A.C. Fernando, S. Panchadcharam and W. A. R. J. Weerakkody, "Unequal error protection technique for ROI based H.264 video coding", IEEE CCECE/CCGEI, pp. 2033-2036, Ottawa, May 2006
- [24]** W. Tan, E. Setton and J. Apostolopoulos, "Lossless FMO and slice structure modification for compressed H.264 video", ICIP, Vol. 4 pp. 285-288, 2007
- [25]** D. D. Schrijver, W. D. Neve, D. V. Deursen, Y. Dhondt and R. V. Walle, "XML-based exploitation of region of interest scalability in scalable video coding", WIAMIS'07, pp. 56-58, June 2007
- [26]** V. Parameswaran, S. Murthy, A. Sen and B. Li, "An adaptive slice group multiple description coding technique for real-time video transmission over wireless networks", MILCOM, pp. 1-7, October 2007

- [27]** Y. Dhondt, S Mys, S. D. Zutter and R. F. Walle, "An alternative scattered pattern for flexible macroblock ordering in H.264/AVC", WIAMIS, pp. 57-60, June 2007
- [28]** T. Ogunfunmi and W. C. Huang, "A flexible macroblock ordering with 3D mbamap for H.264/AVC", ISCAS, Vol. 4, pp. 3475-3478 May 2005
- [29]** W. Hantanong and S. Aramvith, "Analysis of macroblock-to-slice group mapping for H.264 video transmission over packet-based wireless fading channel", Midwest Symposium on Circuits and Systems, pp. 1541-1544, August 2005.
- [30]** S. Aramvith and W. Hantanong, "Joint flexible macroblock ordering and FEC coding for H.264 Wireless video transmission", ISPACS, pp. 139-142, 2006
- [31]** H.264 Packet Loss simulator, http://ftp3.itu.int/av-arch/jvt-site/2005_10_Nice/JVT-Q069.zip, last updated 2005, visited 19 August 2008
- [32]** VCEG, "Common Conditions for wire-line, low delay IP/UDP/RTP packet loss resilient testing", VCEG-N79r1, Santa Barbara, CA, USA, 21-24 September, 2001
- [33]** Stephan Wenger, "Proposed Error Patterns for Internet Experiments", ITU-T VCEG document Q15-I-16r1, October, 1999.
- [34]** Q. Zhang and S. A. Kassam, "Finite state markov model for rayleigh fading channels", IEEE Transactions on Communications, Vol. 47, No. 11, November 1999
- [35]** D. Lal, A. Manjeshwar, F. Herrmann, E. U. Biyikoglu and A. Keshavarzian, "Measurement and characterization of link quality metrics in energy constrained wireless sensor networks", GLOBECOM, pp. 446-452, 2003
- [36]** J. P. Ebert and A. Willig, "A Gilbert-Elliot bit error model and efficient use in packet level transmission",TKN-99-002, Berlin, March 1999
- [37]** JM 12.1 H.264/AVC Reference Software, http://iphome.hhi.de/suehring/tml/download/old_jm/jm12.1.zip, last updated 2008, visited 19 August 2008