ERROR RESILIENT MULTIVIEW VIDEO CODING AND STREAMING

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

BY

ANIL AKSAY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

FEBRUARY 2010

Approval of the thesis:

ERROR RESILIENT MULTIVIEW VIDEO CODING AND STREAMING

submitted by **ANIL AKSAY** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy 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**

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

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 _____

Prof.Dr. Erdal Arıkan
Electrical and Electronics Engineering Dept., Bilkent University _____

Prof. Dr. Murat Tekalp
Electrical and Electronics Engineering Dept., Koç University _____

Asst. Prof. Dr. Cüneyt Bazlamaçcı
Electrical and Electronics Engineering Dept., METU _____

Date: _____

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: ANIL AKSAY

Signature :

ABSTRACT

ERROR RESILIENT MULTIVIEW VIDEO CODING AND STREAMING

Aksay, Anıl
Ph.D., Department of Electrical and Electronics Engineering
Supervisor : Prof. Dr. Gözde Bozdağı Akar

February 2010, 119 pages

In this thesis, a number of novel techniques for error resilient coding and streaming for multi-view video are presented. First of all, a novel coding technique for stereoscopic video is proposed where additional coding gain is achieved by downsampling one of the views spatially or temporally based on the well-known theory that the human visual system can perceive high frequencies in 3D from the higher quality view. Stereoscopic videos can be coded at a rate upto 1.2 times that of monoscopic videos with little visual quality degradation with the proposed coding technique. Next, a systematic method for design and optimization of multi-threaded multi-view video encoding/decoding algorithms using multi-core processors is proposed. The proposed multi-core decoding architectures are compliant with the current international standards, and enable multi-threaded processing with negligible loss of encoding efficiency and minimum processing overhead. End-to-end 3D Streaming system over Internet using current standards is implemented. A heuristic methodology for modeling the end-to-end rate-distortion characteristic of this system is suggested and the parameters of the system is optimally selected using this model. End-to-end 3D Broadcasting system over DVB-H using current standards is also implemented. Extensive testing is employed to show the importance and characteristics of several error resilient tools. Finally we modeled end-to-end

RD characteristics to optimize the encoding and protection parameters.

Keywords: 3D video, compression, streaming, transmission, error resilience

ÖZ

HATA DAYANIKLI ÇOK GÖRÜNTÜLÜ VİDEO KODLAMASI VE DURAKSIZ İLETİMİ

Aksay, Anıl
Doktora, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Gözde Bozdağı Akar

Şubat 2010, 119 sayfa

Bu tezde, çok görüntülü videonun hataya dayanıklı kodlanması ve iletimi için kullanılabilecek yeni teknikler anlatılmaktadır. İlk olarak, iki görüntülü video için yeni bir kodlama tekniği önerilmiştir. Bu teknik görüntülerden birinin uzaysal ya da zamansal olarak kalitesi azaltmasıyla yapılmaktadır. Bu teknik, insan görme sisteminin 3 boyutlu görüntülerdeki yüksek frekans bilgisini daha kaliteli olan görüntüden alma teorisine dayanmaktadır. Önerilen teknik sayesinde iki görüntülü videolar çok az kalite kaybıyla tek görüntülü videoların 1.2 katına kadar hızda kodlanabilmektedir. İkinci olarak, çok çekirdekli işlemciler kullanılarak çok kullanımlı çok görüntülü video kodlama/kodçözme algoritmalarının eniyilemesi ve tasarımı için sistematik bir metod önerilmiştir. Önerilen çok çekirdekli kodçözme yapısı standartlar ile uyumludur. İhmal edilebilir kodlama verimi kaybı ve en az işleme ek yükü ile çok kullanımlı işlemlerin yapılmasını mümkün kılmaktadır. Üçüncü olarak mevcut standartlar kullanılarak internet üzerinden uçtan uca 3 boyut video iletimi gerçekleştirilmiştir. Sistemin uçtan uca hız bozulma karakteristiğini çıkartabilmek için bir metod önerilmiştir, ve bu sistemin parametreleri bu model kullanılarak eniyilenmektedir. Dördüncü olarak, mevcut standartlar kullanılarak DVB-H üzerinden uçtan uca 3 boyut video yayıncılık sistemi gerçekleştirilmiştir.

Hataya dayanıklı kodlama metodlarının önemi ve karakteristiğini belirlemek için detaylı testler yapılmıştır. Son olarak, kodlama ve koruma parametrelerini eniyilemek için uçtan uca hız-bozulum karakteristiği modellenmiştir.

Anahtar Kelimeler: 3B video, kodlama, duraksız iletim, iletim, hata dayanıklı

To my family.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my supervisor Prof. Gözde Bozdağı Akar for her supervision, guidance and encouragements throughout the research. I would like to acknowledge the members of my thesis committee, Prof. Erdal Arıkan, Prof. Murat Tekalp, Assoc. Prof. Aydın Alatan and Asst. Prof. Cüneyt Bazlamaçcı for their support and suggestions which improved the quality of the thesis. I would also like to thank Prof. Erdal Arıkan and Assoc. Prof. Aydın Alatan for their valuable feedbacks in my thesis progress presentations during the last three years.

I would also like to gratefully acknowledge the researchers who directly contributed to this thesis. I would like to thank Dr. Tanır Özçelebi and Engin Kurutepe of Koç University for hosting me in their institute for several times and help with subjective testing of stereoscopic video encoding and content adaptation of stereoscopic video encoding. These are addressed in second chapter of this thesis. I would like to sincerely thank Göktuğ Gürler of Koç University for his collaboration on 3D decoding in third chapter of this thesis. I would like to thank Göktuğ Gürler and Selen Pehlivan of Koç University, Engin Kurutepe of Technical University of Berlin, and Dr. Serdar Tan of Bilkent University for their collaboration on streaming 3D video over internet and this subject is addressed in fourth chapter of this thesis. I would also like to acknowledge my friends Oğuz Bici, Murat Demirtaş and Döne Buğdaycı for their discussions and contributions on the broadcasting of 3D video over DVB-H. This is addressed in the fifth chapter of the thesis.

I would like to thank all of my friends in METU during my graduate years. I would like to thank all the present and previous members of METU Multimedia laboratory, Oğuz, Çağdaş, Murat, Döne, Erman, Özgü, Murat, Berkan, Özlem, Alper, Ahmet, Serdar, Cevahir, Burak, Evren, Engin, Yoldaş, Oytun, Elif and others, for their friendship and collaboration.

Finally and most importantly, I would like to thank my family for their intense support throughout this study. This work is dedicated to them.

This thesis is partially supported by European Commission within FP6 under Grant 511568

with the acronym 3DTV, within FP7 under Grant 216503 with the acronym MOBILE3DTV and The Scientific and Technological Research Council of Turkey under National Scholarship Programme for PhD Students.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

Introduction

3D video is a new area and getting very popular with advances in display technologies. Although 3D illusions had people's imagination since the 19th century, technologies recently reached a necessary level to allow 3D video applications. 3D video applications including 3DTV and 3D telepresence, are offering an impressive sense of depth and increased realism. 3D home display systems that are fed by multi-view video (MVV) are getting very popular and destined to become a serious alternative to classic 2D home-entertainment/communication systems.

A typical 3D transmission system is shown in Figure 1.1 [1]. Main blocks are capture, coding, transmission and display. 3D video requires more information than monoscopic video, and brings more challenge for each of the blocks of the transmission systems.

Capturing process depends on the representation type of input 3D video. There are several representations that can be used for 3D scenes such as multi-view video (MVV) [2], video plus depth (VpD) [3], [4], [5], geometry with texture (lightfield) [6], volumetric (voxels) [7] and holographic [8]. In this thesis, we are dealing with MVV and VpD, due to their popularity and compatibility with the existing systems.

MVV refers to a set of N temporal synchronized video streams coming from cameras that capture the same real world scene from different viewpoints. Stereo video is a special case of multi-view video where N = 2 and cameras are closely located (similar to the distance between two eyes). These two streams are referred as left and right views. The close location of cameras in these applications results in a high redundancy between the sequences from each camera. VpD representation consists of a view and depth information associated with

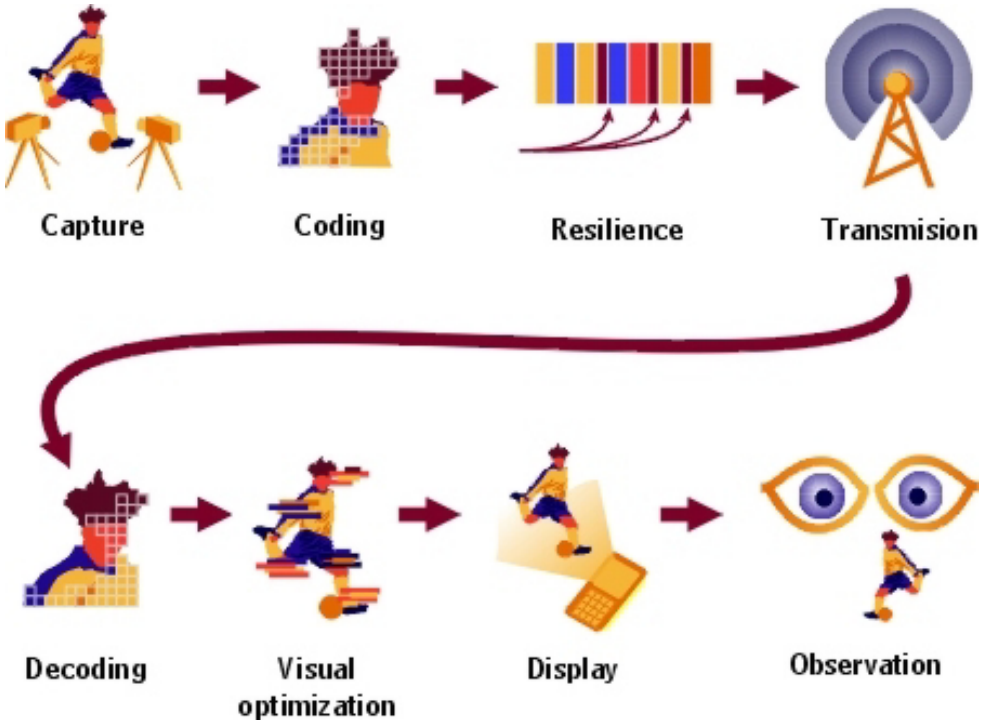


Figure 1.1: A typical 3D transmission system

this view. Depth can be represented as a synchronized grayscale video where each pixel takes an integer value between 0-255. Since depth values are smooth, it can be compressed very efficiently. Depth information may be either provided (either generated synthetically or measured [9]) or estimated from the left and right views of a stereo video [10]. Depending on the display type, rendering right view using left view and depth information may be necessary. Although there are available coding standards for MVV (MVC [11]) and VpD (MPEG-C Part 3 [12]), still there are investigations for improvement and current coding efficiencies are not enough for new displays using 45-views [13].

Error resilience and protection are required for transmission system over lossy networks. Since video encoding uses predictions, losses propagates. In 3D video, prediction structures are more complex and protection is required more to overcome loss propagation. There are two approaches for loss protection. One solution is to use a feedback channel for the lost packets and retransmit lost packets. Feedback channel might not be available in some solutions like broadcasting. Feedback solution also cause delays and not suitable for real-time communications. Forward error correction (FEC) is another solution where data is sent with redundant error correcting codes in order to overcome losses. Multiple description coding

(MDC) is also another alternative which uses multiple descriptions of the video, where each description can be decoded independently, but all of them can have a better decoding quality.

There are two transmission mediums used in this study. Internet and DVB-H. Internet is a lossy and best effort network, where losses are due to congestion. DVB-H is a broadcasting medium and can have severe loss conditions due to mobile fading. More details on the properties of DVB-H will be given in Section 5.

1.1 Major Contributions

Major contributions of this thesis to the existing body of knowledge can be summarized as follows:

- **Asymmetric Stereoscopic Video Coding [14, 15]:** A novel coding technique for stereoscopic video is proposed. Additional coding gain is achieved by downsampling one of the views spatially or temporally based on the well-known theory that the human visual system can perceive high frequencies in 3D from the higher quality view. Using this technique, stereoscopic videos can be coded at a rate upto 1.2 times that of monoscopic videos with little visual quality degradation.

- **Multi-threaded 3D Decoding [16]:** A systematic method for design and optimization of multi-threaded multi-view video encoding/decoding algorithms using multi-core processors is proposed. The proposed multi-core decoding architectures are compliant with the current MVC international standard, and enable multi-threaded processing with negligible loss of encoding efficiency and minimum processing overhead.

- **Optimization of 3D Internet Streaming [17]:** End-to-end 3D Streaming system over Internet using current standards is implemented. Encoding and decoding performances in a lossy network is modeled for to optimize the parameters of the system.

- **Optimization of 3D DVB-H Broadcast [18, 19]:** End-to-end 3D Broadcasting system over DVB-H using current standards is also implemented. Extensive testing is employed to show the importance and characteristics of several error resilient tools. Finally we modeled end-to-end RD characteristics to optimize the encoding and protection parameters.

1.2 Scope and Outline of the Thesis

In this thesis, we address error resilient multi-view video streaming systems. Our aim is to propose solutions for different parts of the system.

In Chapter 2, 3D coding methodologies and proposed algorithms are presented. Multi-view prediction structure is proposed, which is used in later standards.

In Chapter 3, complexity issue of 3D decoding is emphasized and a novel multi-core solution to 3D decoding is proposed.

In Chapters 4 and 5, transmission of 3D is investigated for two different transmission environment.

In Chapter 4, streaming over Internet is investigated. Standards based architectures for 3D streaming, adaptive rate and format control for 3D and error resilient streaming are the proposed algorithms in this section.

In Chapter 5, broadcasting of stereoscopic video over DVB-H with error resilient tools are proposed. Optimization of encoding and error protection parameters are also proposed using end-to-end distortion modeling.

Chapter 6 concludes the dissertation with a summary of the work done and a discussion of the results.

CHAPTER 2

3D Coding

In this chapter, we will review the current state of the art coding techniques for 3D video according to the most used 3D representations, namely MVV and VpD. Other representations are not addressed in this thesis. After reviewing the current technologies, we will explain our proposed asymmetric coding techniques used for stereoscopic video. Although this work is concentrated on stereoscopic video, it can be easily adapted for MVV as well. Final section will explain our extensive tests on coding of stereoscopic video for mobile devices.

2.1 Current state of the art

Depending on the representation of 3D, different approaches and standards are available for 3D coding. H.264/AVC [20] is the current state-of-the-art monoscopic video codec providing almost twice the coding efficiency with the same quality comparing the previous codecs [21]. 3D encoders are mostly based on H.264/AVC.

2.1.1 Multi-View Video

Compressing multi-view sequences independently is not efficient since the redundancy between the closer cameras is not exploited. MPEG and VCEG groups jointly created an ad-hoc group 3DAV [22] which received several contributions for Multi-View coding. A good review on the proposed algorithms can be found in [23, 24]. As an output of this work, Multi-View Video Coding (MVC)[11] is generated as an amendment to H.264/AVC, exploiting temporal and inter-view redundancy by interleaving camera views and coding in a hierarchical manner.

First draft is approved in October 2008 and second draft is approved in February 2009.

In [25], we proposed a multi-view video codec based on H.264/AVC exploiting the correlation between cameras in a backward compatible way. Several prediction structures are proposed with the signalling in the bitstream. Codec is based on baseline profile and using only P pictures. It showed superior performance for dense cameras. First version of MVC extension of H.264/AVC was released in 2006 and we compared the performance against this standard codec. Since MVC uses high profile with hierarchical B pictures, it performs better than our codec. However due to the speed and the less complex prediction structures of our encoder, it could be used for some applications such as real time video communication. In 2008, Nokia also released a baseline version of MVC which is very similar to our codec.

MVC extension of H.264/AVC is based on High Profile. Mainly it uses Hierarchical B-pictures [26], Context-Adaptive Binary Arithmetic Coding (CABAC) and disparity compensation between the frames of different cameras. Therefore, it requires more pictures in Decoded Picture Buffer (DPB) and also requires more buffering before the pictures can be given to display in actual display order. Although general MVC requires a complex prediction structure, in [27] a simplified prediction scheme is proposed without significant loss of coding efficiency. In simplified prediction scheme, inter-view prediction is only allowed for so called anchor pictures.

Recently, Blu-ray Disc Association (BDA) announced the ”Blu-ray 3DTM” specification which calls for encoding 3D video using the MVC codec to be supported by all Blu-ray Disc players. Usage of MVC will be more popular by upcoming new specifications and streaming and error resilient tools based on MVC will be required for transmission mediums.

2.1.2 Video Plus Depth

In ATTEST project [28], single view and depth map is compressed and broadcast. MPEG-C Part-3 [12] is the current standard to encode VpD data by individually compressing video and depth data using standard H.264/AVC [29]. Since depth data is much easier to encode rather than another view, compression efficiency of VpD is better than MVC coded stereoscopic videos, however image-based-rendered view have artifacts due to occlusion. Another advantage of VpD representation is to enable changing the distance between the cameras while

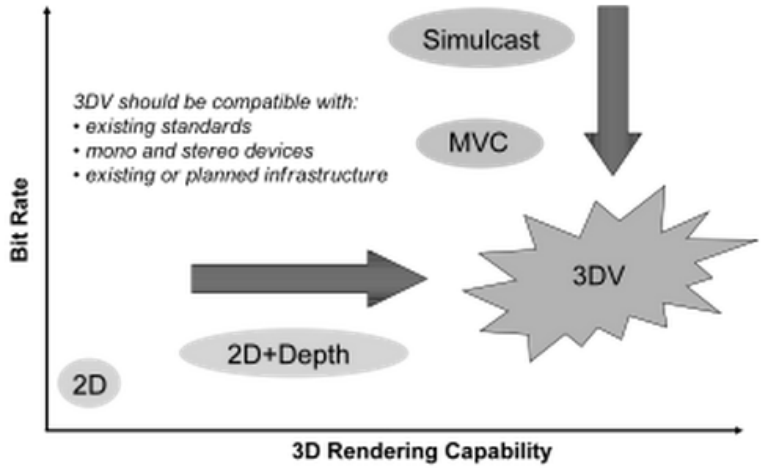


Figure 2.1: Illustration of 3D rendering capability versus bit rate for different formats

rendering other view.

2.1.3 Multi-View Video Plus Depth

In order to handle 45-view displays, MVV and VpD are not adequate. The solution to feed such systems is to decode a limited number of views using MVC and then generate artificial views using interpolation techniques such as [30, 31]. There are two important reasons for this approach. First practical reason is the problem of data acquisition. It is difficult to set up a mobile recording system that is composed of fifty high definition cameras that are calibrated. Second reason is the problem of data transmission. It is difficult to transmit even encoded data for fifty views over the Internet. MVC provides significant compression gain but the result is still linear with the number of views. Multi-view streaming systems such as [32] use video+depth format to drive displays with high number of views. Future display systems will use M video signal and N depth maps.

As shown in Figure 2.1 [33], MPEG group visons a new upcoming 3D Video format (3DV) that goes beyond the capabilities of existing standards to enable both advanced stereoscopic display processing and improved support for auto-stereoscopic N-view displays, while enabling interoperable 3D services. This is an ongoing task between MPEG working groups.

2.2 Asymmetric Stereoscopic Coding

In order to exploit the redundancies between MVV, H.264/AVC based multi-view codec [25] is proposed. In case of stereoscopic video, coding gain is sometimes below % 20. In order to improve the coding efficiency without degrading the visual quality, we used properties of Human Visual System (HVS).

There are two different theories about the effects of unequal bit allocation between left and right video sequences, namely *fusion theory* and *suppression theory* [34, 35, 36]. In fusion theory, it is believed that the stereo distribution must be equally made for the best human perception. On the other hand, in suppression theory, it is believed that the highest quality image in the stereo-pair determines the overall perception performance. Therefore, according to this theory, we can compress the one of the image from a stereo pair as much as possible to save bits for the other image from the stereo pair, so that the overall distortion is the lowest. If we assume that the overall distortion measure of a stereo-pair will be a weighted average of the individual images, we can define weighting coefficients between right and left image distortion values to take different amount of contributions from each picture into account.

In monoscopic video coding, chrominance values are usually subsampled, since HVS is less sensitive for chrominance values. Similar to this behavior and theories for stereo perception, it is reported in [37] that, HVS can use the high frequency information in one of the videos if the other video is low pass filtered. The authors proposed using spatial subsampling in one of the videos to reduce bandwidth requirements without any visual quality degradation. Authors also tried temporal scaling, but visual quality results show that spatial scaling gives more promising results.

Using these theories, we have implemented temporal and spatial downsampling in our H.264/AVC based multi-view codec to enable asymmetric coding [14]. We have experimented with several parameters and conducted a subjective quality test with the coded sequences. In the following subsections, we will explain modifications of the codec and the quality test setup and the results.

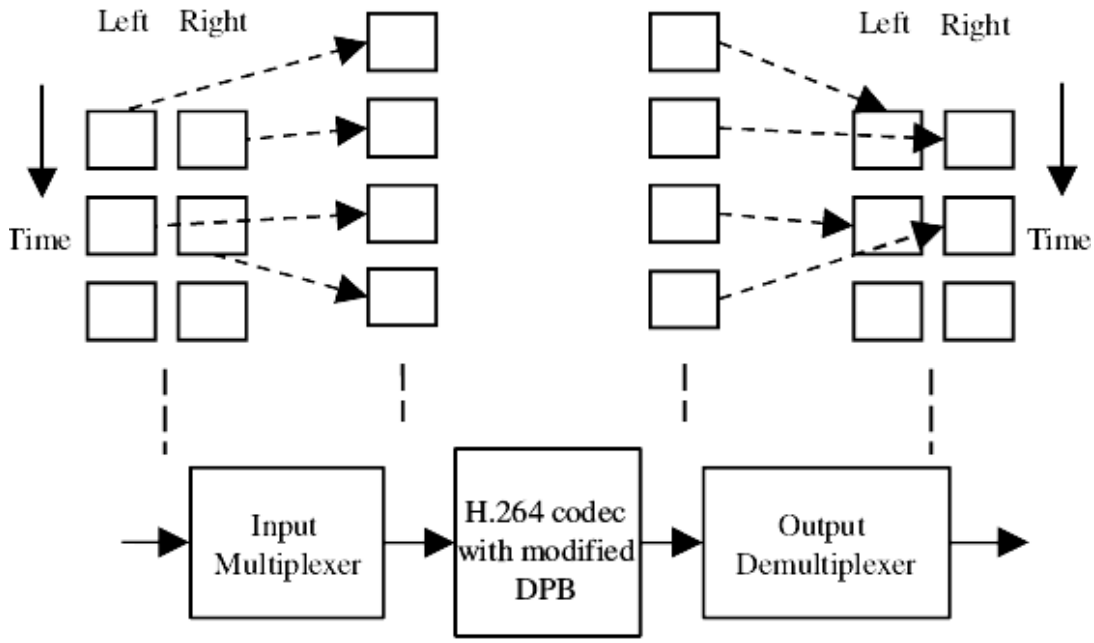


Figure 2.2: Stereoscopic Encoder

2.2.1 H.264/AVC based multi-view codec

The structure of the encoder and decoder is shown in Figure 2.2 and Figure 2.3 respectively based on the H.264/AVC based multi-view codec [25]. In order to improve the coding gain without any significant perceptual quality loss, we added two modes called spatial and temporal scaling.

Spatial Scaling The spatial scaling mode corresponds to downsampling the right video by a predefined scale prior to encoding in order to improve the coding gain. The implementation of downsampling the image consists of both decimation and low-pass filtering in order to prevent the aliasing. For spatial scaling following filters are used:

13-tap *downsampling* filter:
{0,2,0,-4,-3,5,19,26,19,5,-3,-4,0,2,0}/64
11-tap upsampling filter:
{1, 0,-5, 0, 20, 32, 20, 0,-5, 0, 1}/64

Filters are applied to all Y, U, and V channels and in both horizontal and vertical directions. The picture boundaries are padded by repeating the edge samples. These filters are currently

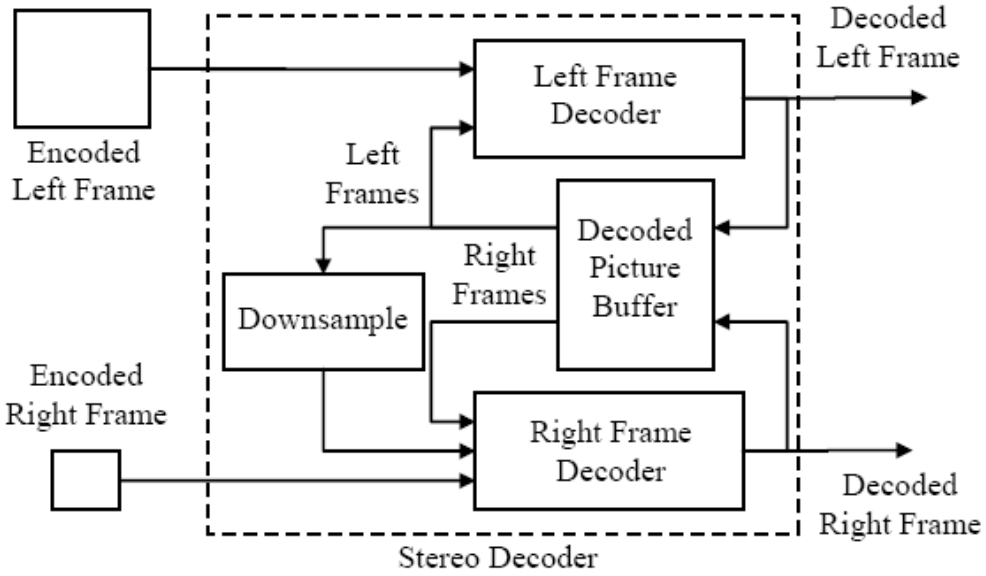


Figure 2.3: Stereo Decoder

used in Scalable Video Coding extension of H.264/AVC [38] and explained in [39] . In order to keep filtering process simpler in both encoder and decoder, we have implemented downscaling by factors of 2 (dyadic sampling) in both dimensions. Although the spatial scaling is applied to the right channel only, left frames are also temporarily scaled just for disparity estimation required for right frame coding.

Temporal Scaling Temporal scaling mode corresponds to the decimation of right video in time, i.e. frame dropping in the right sequence. The implementation of temporal down-sampling is done by sending all the macro blocks of dropped frame as skipped mode of the H.264/AVC standard. In our codec notation, temporal scaling of n denotes encoding 1 frame out of n frames and dropping the remaining n-1 frames.

Test Method We have adapted DSCQS Test [40] method where non-experts and inexperienced assessors are used. The two videos are evaluated by the assessor on a continuous scale ranging from 0 to 100 with help of two sliders.

2.2.2 Test Methodology and Display System

Multiple assessors are shown two conditions, A and B (two stereoscopic images), consecutively one of which is always the source and the other is the tested condition applied on the source. The identity of the images, whether it is the source or the test condition, should be known by the experimenter but not by the assessors. The next pair of conditions is shown after the assessors establish an opinion.

Analysis Method: For the analysis of the test results, each evaluation is graded between 0-100 and the difference between the scores of source image and the test condition is calculated to find the score of that test condition on that image by the assessor. After all these scores are calculated, the values are normalized to fit in 0-100. And as a final step, to find the scores of each algorithm (test condition) the average of all the scores over the assessors and images are taken. Scores of the algorithms can be compared with their closeness to the number to which zero score is mapped during the normalization process.

Display System: Subjective evaluation of the encoded stereo videos was conducted at Koç University using a pair of Sharp MB-70X projectors. More information of this stereoscopic display system can be found in [14].

2.2.3 Experiments and Results

In these experiments, we investigated effects of spatial and temporal scaling in stereoscopic videos. In order to meet time requirements of assessment test, we use only 4 video sets with 8 algorithms.

Assessors: 21 assessors (13 female, 8 male with average age 24) with ages ranging from 19 to 36, volunteered to participate in the experiment. The participants were non-experts in the area of picture quality and were screened for color vision, stereo depth perception and visual acuity.

Test Material: As the test material, four different stereoscopic video pairs are used: Balloons (720x480, 25 fps, 10 seconds), Botanical (960x540, 15 fps, 5 seconds), Flowerpot (720x480, 25 fps, 10 seconds), Train (720x576, 25 fps, 10 seconds). 8 different algorithms are applied on these videos as shown in Table 2.1. Sample frames are shown in Figure 2.4 and Figure 2.5.

Table 2.1: Algorithm applied to test videos

ORIG	Original
SIMUL	Simulcast coding
S1T1	Stereo coding, no spatial, no temporal scaling
S1T2	Stereo coding, no spatial, temporal scaling 2 for right frames
S1T2L	Stereo coding, no spatial, temporal scaling 2 for left and right frames
S1T3	Stereo coding, no spatial, temporal scaling 3 for right frames
S2T1	Stereo coding, spatial scaling 2, no temporal scaling
S4T1	Stereo coding, spatial scaling 4, no temporal scaling
S4T3	Stereo coding, spatial scaling 4, temporal scaling 3 for right frames

Table 2.2: Normalized Bitrates of Algorithms

	BALN	FLOW	BOTA	TRAIN	Av.
SIMUL	2	2	2	2	2
S1T1	1.901	1.927	1.452	1.881	1.79
S1T2	1.606	1.692	1.289	1.601	1.547
S1T2L	1.324	1.45	0.923	1.336	1.258
S1T3	1.489	1.586	1.228	1.492	1.449
S2T1	1.242	1.267	1.065	1.252	1.207
S4T1	1.091	1.095	1.012	1.085	1.071
S4T3	1.053	1.069	1.006	1.049	1.044

As a result a total of 41 evaluation pairs, including first 5 stabilizing pairs, are shown to the assessors and it is assured that each test does not take more than 30 minutes.

All the test videos are encoded with the modes explained in Table 2.1. Intra period of 25 and Quantization Parameter (QP) of 28 are used while encoding. Total bitrate for simulcast coding is interpreted as twice the data required compared to single view coding and the bitrates of all other algorithms are normalized accordingly and can be found in Table 2.2. By only spatial subsampling of right video with 2 in both dimensions we have approximately matched 1.2 times the single view bitrate.

After all the assessors finish the test, the scores are evaluated and normalized. Average MOS scores and confidence intervals for each algorithm is shown in Table 2.3 and Table 2.4. Due to the normalization, 0 (best quality) is mapped to 38, and the success of the algorithms can be measured by closeness of their mean to 38. As it can be seen, the mean of the original video is also not exactly 38, which is due to the misjudgment of the assessors and it is expected.

Simulcast (SIMUL) coding and stereo coding without scaling (S1T1) have similar or better

Table 2.3: MOS Scores of algorithms

Algorithm	MOS	Normalized Bitrate	SNR Overall	SNR Left	SNR Right
ORIG	43.6	N/A	INF	INF	INF
SIMUL	41.93	2	36.09	36.05	36.13
S1T1	42.11	1.79	35.96	36.05	35.88
S2T1	46.68	1.207	31.05	36.05	28.8
S1T2	48.68	1.547	32.35	36.05	31.11
S1T2L	48.71	1.258	31.15	31.19	31.11
S4T1	53.02	1.071	27.82	36.05	25.15
S1T3	55.57	1.449	30.58	36.05	28.92
S4T3	59.26	1.044	26.13	36.05	23.37

performances over original video. Since QP is low, reconstructed video quality is visually lossless (with average PSNR of 36 dB) and misjudgment is expected for these algorithms as well. Also DCT based coded images are reported [41] to be preferred by assessors comparing to original.

We can see from the results that scaling with 3 or 4 in both spatial and temporal domain is not acceptable. According to the bitrate and MOS score, only spatial scaling looks like the optimum solution. Spatial scaling by 4 corresponds to 16:1 reduction in image size; therefore its performance is not acceptable. Spatial scaling with non-dyadic factors and better filters for upsampling might keep the visual quality at desired levels with bitrate similar to single view coding bitrates.

According to the video characteristics (slow motion video), temporal scaling in either right channel or both channels might give good results as well. By analyzing the characteristics of the video in each GOP (each chunk of video sequence that can be decoded without use of other parts of the sequence), appropriate scaling can be applied to decrease bitrate without visual quality degradation.

By improving coding efficiency of stereoscopic video, prediction between views are introduced, which makes the bitstream less robust to errors. Multiple Description Coding (MDC) is another way to improve system performance in case of losses. By using this asymmetric stereoscopic video codec, we proposed several schemes for MDC of stereoscopic video in [42].

Table 2.4: MOS Scores and confidence intervals for each video sequence

	BALN	FLOW	BOTA	TRAIN
ORIG	43.0 +- 0.6	44.1 +- 0.8	42.0 +- 0.4	45.2 +- 0.8
SIMUL	42.4 +- 0.9	40.0 +- 0.8	43.7 +- 1.0	41.7 +- 1.3
S1T1	41.5 +- 1.1	39.7 +- 0.9	45.5 +- 0.9	41.8 +- 1.5
S2T1	46.9 +- 1.0	47.5 +- 1.2	46.0 +- 1.0	46.3 +- 1.1
S1T2	45.3 +- 1.4	45.8 +- 1.3	57.0 +- 1.8	46.6 +- 1.1
S1T2L	47.2 +- 1.5	51.7 +- 1.8	52.0 +- 1.1	43.9 +- 1.4
S4T1	51.5 +- 1.6	52.9 +- 1.7	54.0 +- 1.7	53.7 +- 1.4
S1T3	56.5 +- 1.7	54.7 +- 1.7	60.0 +- 1.7	51.0 +- 1.4
S4T3	53.3 +- 1.6	61.2 +- 1.5	67.5 +- 1.7	55.0 +- 1.6

Figure 2.4: Sample frames of Train sequence

Figure 2.5: Sample frames of Botanical sequence

2.3 Evaluation of Stereo Video Coding Schemes for Mobile Devices

Mobile devices such as mobile phones, personal digital assistants and personal video/game players are somehow converging and getting more powerful, thus enabling 3D mobile devices a reality. In order to store or transmit stereo video in these devices, coding techniques from both monoscopic video coding and multi-view video coding can be used.

In literature, there exist a few 3D mobile device prototypes [43, 44, 45, 46] which are based on auto-stereoscopic 3D displays either parallax barrier or lenticular lens structures or stereoscopic display which can be observed with anaglyph glasses. For coding, they use different technologies. In [43], stereo video is encoded using H.264/AVC MVC extension; however some of the tools such as Hierarchical B pictures are not used to decrease decoding complexity. In [44], stereo video of QVGA size is coded using simulcast MPEG-4 encoder with asymmetric coding (Left and right videos are encoded with different resolutions). In [45], both stereo video and video plus depth representations are used. Stereo video of QVGA size is first converted into monoscopic video by tiling the images (side-by-side) and then encoded with MPEG-4 (simple profile). Video and depth are encoded by MPEG-4 as separate streams. In both representations, 24 fps can be achieved on the decoder side. In [46], stereo video is fed into H.264/AVC monoscopic video encoder as an interlaced video. 10 fps can be achieved on the decoder side.

Even though in this preliminary studies different techniques are used, in order to efficiently store or transmit stereo video to mobile devices, coding techniques from both monoscopic video coding and multi-view video coding should be examined in detail. We examined the video codec performances for stereoscopic videos with mobile device resolutions with different profiles.

Mobile devices have smaller displays and the current prototypes mostly use QVGA resolution. Previous MVC experiments were also performed on Multi-View Video sequences with multiple cameras. When the number of cameras is only two, coding efficiency decreases. Also using MVC requires larger Decoder Picture Buffer (DPB), causing problems with mobile devices. Besides deciding on the video codec to use, there is also an issue of selecting profile and coding tools that are going to be used. Selecting a profile changes both coding efficiency and decoding complexity. Higher profiles increase coding efficiency with the ex-

pense of decoding complexity. Such prototypes tend to use baseline/simple profiles of the video encoders due to limited processing power.

2.3.1 Video Coding Tools and Properties

In this subsection, tools and properties of this codec that is related to this work are presented. More information can be found in [21].

H.264/AVC has several profiles to suit the needs of different applications: Baseline Profile (BP), Main Profile (MP), Extended Profile (XP), High Profiles (HiP). In mobile applications, mostly BP is used.

There are 3 picture types in H.264/AVC. I-pictures are encoded without the use of motion compensation, thus they are independently decoded. P-pictures are predicted using only the previously decoded frames. B-pictures are bi-directionally predicted (both from past and future frames). B-pictures are not supported in BP.

Video frames are encoded with Group-of-Pictures (GoP). Each GoP starts with I frame and followed by B or P frames. By increasing GoP size coding efficiency increases while capability of dealing with losses decreases with having less frequent I-pictures. Hierarchical B-pictures [26] can also be used within the syntax of H.264/AVC and achieve better coding efficiency, however the decoding complexity increases. Pictures at the GoP boundaries are encoded as I-frames and frames in between are encoded as B-frames in an hierarchical order. For example for GoP size of 8, Frame#0 and Frame#8 is encoded as I-frame. Then B-frames are encoded in the following order: Frame#4[0,8], Frame#2[0,4], Frame#6[4,8], Frame#1[0,2], Frame#3[2,4], Frame#5[4,6], Frame#7[6,8] *(Frame numbers in the brackets show the pictures used in motion estimation for encoding the required frame.)*

In H.264/AVC, two different entropy coding method can be used. Context-adaptive binary arithmetic coding (CABAC) is using the probabilities of syntax elements in a given context to losslessy compress syntax elements. Context-adaptive variable-length coding (CAVLC) is lower-complex algorithm to encode those elements. Only CAVLC is used in BP.

MVC extension of H.264/AVC is based on High Profile and explained in Section2.1.1.

Tested configurations in the experiments are given below and in Table 2.5.

Table 2.5: Tools used by different coding schemes

	B-frames	*CABAC*	*# of Reference Frames (Forward - Backward)*
IPP	No	No	1-0
IPP-CABAC	No	Yes	1-0
IBP	Yes	Yes	2-2
Hier	Yes	Yes	4-4
IPP-Stereo	No	No	2-0
IPP+CABAC-Stereo	No	Yes	2-0
MVC-Simp	Yes	Yes	6-6
MVC-General	Yes	Yes	8-8

IPP: Left and right videos are encoded separately using H.264/AVC with baseline profile settings and pictures are encoded as I-frame followed by P-frames for each GoP. No B-frames are used. Entropy coder is CAVLC.

IPP+CABAC: Similar to IPP with additional CABAC entropy coding instead.

IBP: Left and right videos are encoded separately using H.264/AVC with main profile and pictures are encoded as I-frame followed by P- and B-frames. Entropy coder is CABAC.

Hier: Left and right videos are encoded separately using H.264/AVC with main profile and Hierarchical-B pictures.

IPP-Stereo: Left and right videos are interleaved into a single sequence and encoded using IPP settings.

IPP+CABAC-Stereo: Left and right videos interleaved into a single sequence and encoded using IPP+CABAC settings.

MVC-Simp: Left and right videos are encoded using MVC Extension (High Profile, Hierarchical B-pictures, CABAC and Disparity Compensation). Right view pictures can only be predicted from right view pictures and I frames of left view.

MVC-General: Similar to MVC-Simp with the prediction structure allowing right view pictures to be predicted from all left pictures.

Figure 2.6: Stereo test sequences: (a) Rena, (b) Adile and (c) Ice.

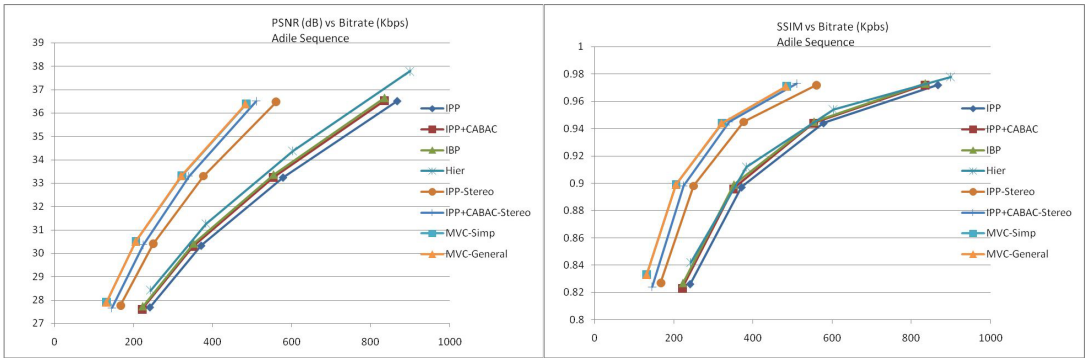
2.3.2 Experimental Results

The results are provided for stereoscopic video pairs ”Rena” (Recorded by cameras with a stereo distance and provided by Tanimoto Laboratory, Nagoya University [47]), ”Adile” (Computer generated animation by Momentum [48]) and ”Ice” (Converted to 3D from 2D scene using [49] [Source: BBC documentation ”Planet Earth”]) . Videos are first downsampled to QVGA sizes. Resolution of Rena and Adile sequences is 320x240 and resolution of Ice sequence is 320x192. Frames from the sequences can be seen in Figure 2.6. GoP size is selected as 8 frames. Monoscopic codec used is H.264/AVC Reference Software JM 14.2 [50] and Multi-view codec is H.264 MVC Reference Software JMVC 2.0 [51]. First 81 frames are encoded from both left and right sequences. Fixed Quantization Parameters (QP) 26, 32, 36, and 40 are used to generate rate-distortion curves. Distortion metrics are PSNR and SSIM [52] and averaged over both left and right frames.

RD-curves for each sequence are given in Figure 2.7. In all sequences difference between MVC-Simp and MVC-General is negligible. In case of MVC encoding MVC-Simp is preferable as stated in [27]. Similarly, MVC schemes provide a significant improvement over IPP and IPP-CABAC for all sequences. However in lower bitrates, difference between IPP+CABAC-Stereo and MVC-Simp is about 0.5-1 dB.

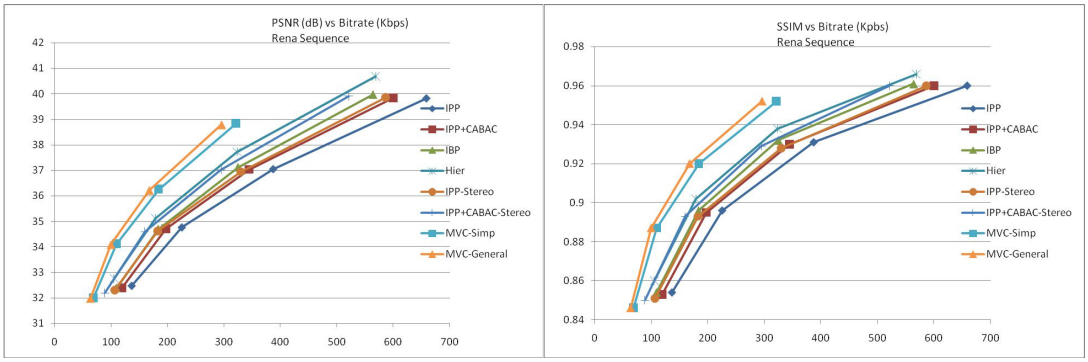
In [53], H.264/AVC encoder and decoder usage is extensively studied for complexity and memory usage. It is stated that

- B-frames are one of the main tools that affect the access frequency and the decoding speed,



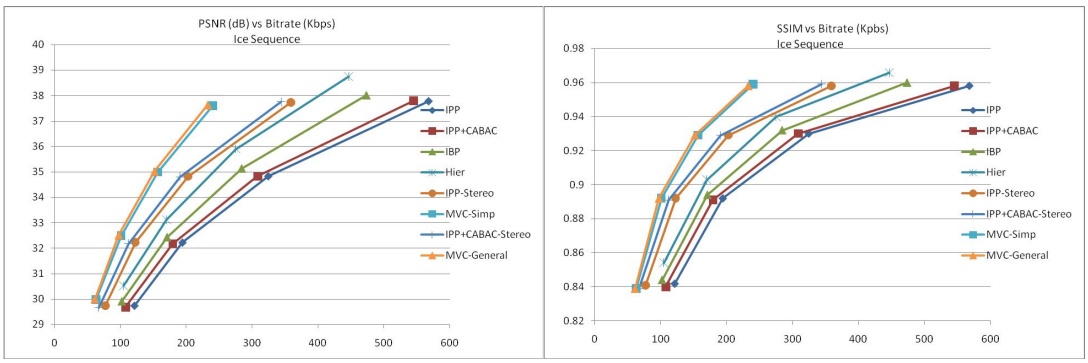
(a)

(b)



(c)

(d)



(e)

(f)

Figure 2.7: PSNR/SSIM vs bitrate for the sequences (a)-(b): Adile, (c)-(d): Rena, (e)-(f): Ice

Table 2.6: Decoding performance of coding schemes

Coding Scheme	*Frames per second (fps)*
IPP	68.43846
IPP-CABAC	64.76772
IBP	59.61354
Hier	57.13102
IPP-Stereo	64.87067
IPP+CABAC-Stereo	64.36886
MVC-Simp	47.14757
MVC-General	46.41807

- Complexity increase due to CABAC is minor,
- Multi-reference frame usage causes a linear increase in memory peak usage.

Although the decoders in reference softwares are not optimized for speed, an analysis on decoding speed of the compressed bitstreams are given in Table 2.6. Since decoding is fast, decoding speeds are calculated by decoding the compressed videos (Rena sequence encoded with QP=32) 100 times and then averaging the results on a PC with 3.4 Ghz processor and 3 GB RAM.

Depending on the processing power and memory of the mobile device the following two schemes can be used: H.264/AVC MVC extension with simplified referencing structure (MVC-Simp) and H.264/AVC monoscopic codec with IPP+CABAC settings over interleaved stereoscopic content (IPP+CABAC-Stereo).

CHAPTER 3

3D Decoding

In this chapter complexity problems of 3D decoding with a proposed solution is explained. The current trend in designing more powerful general-purpose processors is based on multi-core architectures [54]. However, increasing the number of cores does not automatically yield performance gain if the software is not designed to efficiently and effectively divide the workload among multiple cores. Therefore, software developers should consider multi-threaded architectures to take advantage of the state of the art processors. MVV encoding/decoding is one of the areas where such consideration must be taken.

3.1 Multi-Threaded MVC-Compliant Multi-View Video Decoding

MVV requires more advanced video compression algorithms to keep the data rates at manageable levels, which in turn require more powerful processors for real-time encoding/decoding and display performance. In the meantime, the pace of increase in the speed of processors has recently reached saturation. As a result, the CPU manufacturers have moved away from designing processors at higher clock frequencies, and started to add multiple cores at these saturated clock frequencies. However, the increase in the number of cores does not automatically provide proportionally high performance gains. Carefully designed and implemented multi-threaded software architectures will become a critical factor in order to utilize the processing power of future multiple-core processors [54].

In the literature, there are different proposals to decode multi-threaded H.264/AVC sequence in a multi-threaded manner [55, 56, 57]. However, multi-threaded MVC decoding is a relatively fresh topic and there are only two approaches, [58] and [59], in the literature. The

first one [58] is an early version of our own work where the second multi-threading method is adopted, and we propose generating independently decodable MVC streams with a limited amount of redundant decoding operations. The second approach [59] is an extension of a multi-threading implementation for H.264/AVC to MVC. In this approach, the key idea is to start decoding a predicted frame as soon as the necessary macroblocks in the reference frame is decoded.

3.1.1 State of the art in Multi-threaded MVC

A new coding structure is proposed in [59] to allow parallel encoder/decoder operation for different views without a significant change in the coding efficiency and it is adopted in the upcoming MVC standard. Parallel encoding/decoding is enabled by using constraints on the available reference area that a macroblock can depend from the other views. One example case for available reference area selection is depicted in Figure 3.1. In this example, only the first two row of macroblocks in View-0 are necessary for the first row of macroblocks in View-1. That means decoding of View-1 can be initiated once that region is decoded by the first thread. Similarly for the second row of macroblocks in View-1, decoder needs to wait until first 3 rows are available. Only SEI messages are sent from the encoder to signal the available area in order to start pipelining of views. Experimental results show similar coding efficiency with significant parallelism available with this structure. However, this approach requires frequent synchronization between threads and introduces some delay. Moreover, the implementation of SEI messages are optional and only decoders supporting the SEI can take advantage of this technique.

3.1.2 Proposed Multi-Threaded MVC

The proposed solution for multi-threaded processing is to decompose the input N-view stream into M independently decodable sub-streams, and then perform the decoding of each sub-stream by separate threads using multiple instances of an optimized MVC decoder. There are two important criteria in generating such independent streams: i) minimum loss of coding efficiency and ii) minimum processing overhead. The processing overhead refers to redundant decoding of some data at more than one cores in order to achieve independent parallel decoding. For example, in the case of simulcast, all streams can be encoded/decoded independent

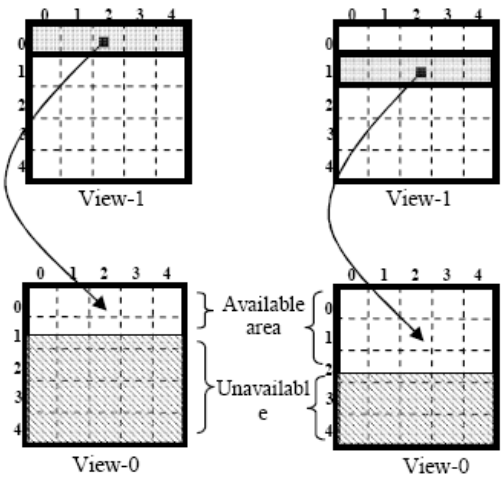
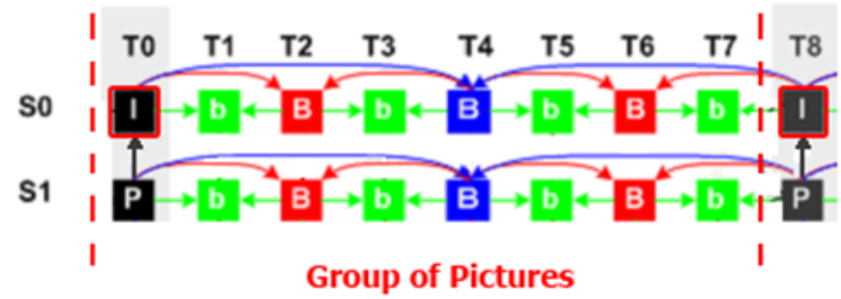


Figure 3.1: Reference area selection

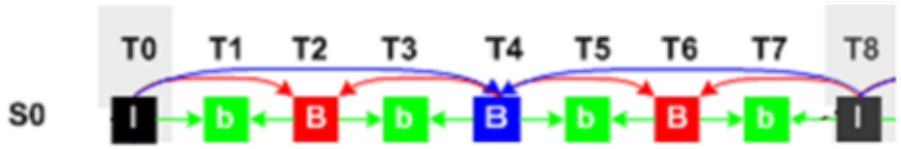
of each other in separate threads with no processing overhead, but simulcast results in loss of coding efficiency compared to MVC. The case of MVC with simplified inter-view prediction for stereo coding (N=2) is illustrated in Figure 3.2. In Figure 3.2(a), the MVC stream for single thread stereo video encoding/decoding is depicted. Two separate streams required for independent decoding of two views on two cores are shown in Figure 3.2(b) and 3.2(c). If two separate threads are used to perform the decoding operation, the I-Frame in each group of pictures (GOP) needs to be decoded twice, which results in processing overhead, but there is no loss of encoding efficiency compared to single thread decoding of the stereo stream shown in Figure 3.2(c). In order to obtain such independently decodable sub-streams for N>2 and M>2, the video must be encoded using special inter-view prediction schemes depending on the number of cores. Special prediction schemes for N=8 views on dual (M=2) and quad (M=4) core platforms will be described in the following.

3.1.2.1 MVC Decoding using Dual Core Platform

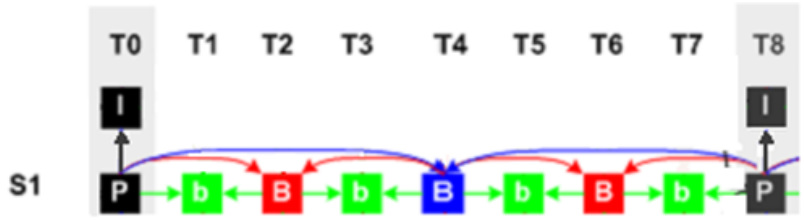
For MVV with N=8, it is possible to generate two independently decodable sub-streams by letting view 4 to be the independent view as shown in Figure 3.3(b). Then, views 4, 5, 6 and 7 can be decoded by the first thread and views 4, 3, 2, 1, and 0 can be decoded by the second thread. Notice that the second thread does not have to decode all frames from view 4, redundant decoding of only I frames will be sufficient. Therefore, the processing overhead



(a)



(b)



(c)

Figure 3.2: Decomposition of a stereo stream into two independent sub-streams. (a) Stereo stream, N=2, for single-thread decoding (b) Sub-stream for view 0 (c) Sub-stream for view 1

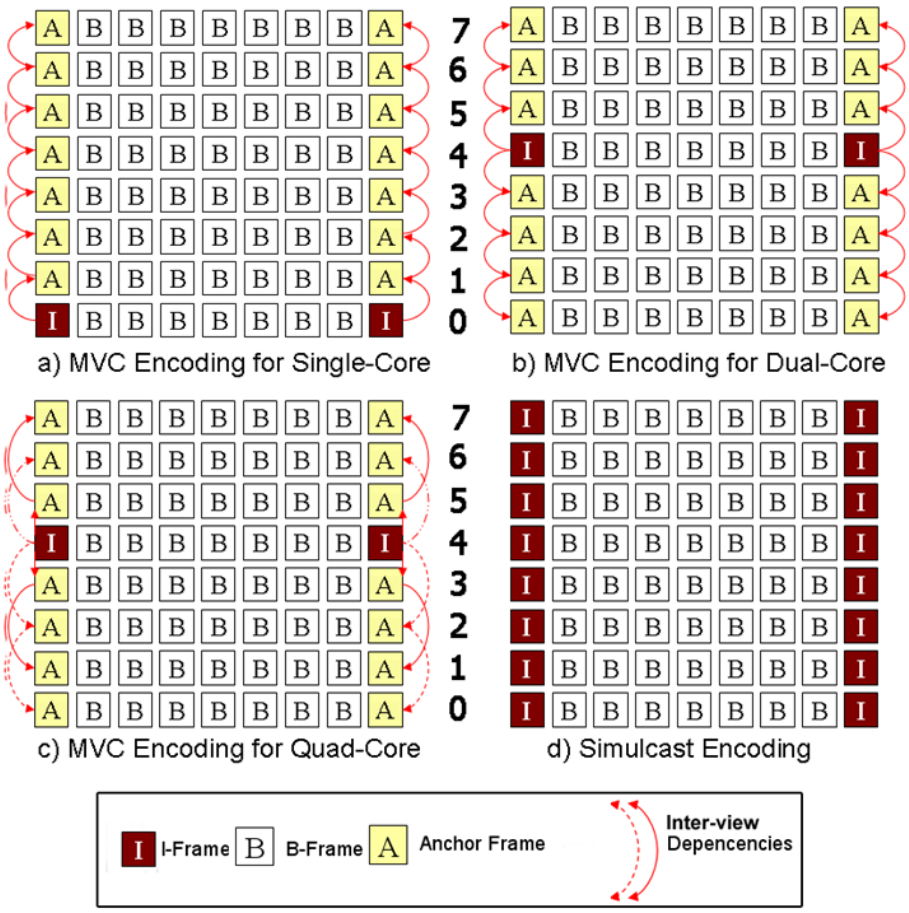


Figure 3.3: I-frames (Intra) are intra-coded and can be decoded without requiring any other frame. A-frames (Anchor) are predicted from frames that are from the same time instant but different view. B frames (bi-predictive) can be decoded using both past and future frames from the same view.

per GOP is the decoding I frame of the independent view, and there is no loss of encoding efficiency compared to single core decoding as in Figure 3.3(a).

3.1.2.2 MVC Decoding using Quad Core Platform

For the quad-core case, we require four independently decodable sub-streams, which can be generated by splitting each sub-stream of the dual-core case into further two sub-streams. This is done by defining inter-view dependencies between every second view as show in Figure 3.3(c). The drawback of defining such dependencies is slight increase in the overall bitrate since the similarity between the views start to diminish as the baseline between cameras in-

creases. Therefore, encoding efficiency may slightly decrease due to prediction structures required for M>2 threads. The loss of coding efficiency is demonstrated by results in the following subsection. The processing overhead is again due to decoding I-frames of the independent view. However, this time it is performed by four threads, and therefore, the overhead is decoding I-frame three times.

3.1.3 Experiments and Results

We have created a testbed that is composed of personal computers with single, dual and quad core processors at identical clock frequencies (2.4 GHz). Four different test sequences are encoded at multiple resolutions, namely 320x240 (CIF), 640x480 (SD), and 1280x960 (HD). Moreover we have selected three different display types, legacy monoscopic display (1 view), stereoscopic 3D display (2-view) such as [60], and autostereoscopic display (8-view) such as [61].

We have used 4 different video sequences which are recorded with 8 cameras. Each source video has different characteristics; Adile is a computer generated video stream at 25 fps and can be encoded with higher efficiency when compared to other streams due to lack of noise that is introduced by recording devices. It has 5 cm space between 1D array of virtual cameras [48]. Information for the other sequences, namely, Ballroom, Race and Rena, can be found in the JVT document [62].

3.1.3.1 Is multi-threading required for real-time decoding?

Our first objective is to find the case when an optimized decoder fails to sustain the frame rate. The results that are presented in Table 3.1 reveal that a single-threaded decoding cannot deliver frames at an adequate rate for MVV display (8-view) at HD-Ready resolution and the addition of multiple cores does not provide performance gain when multi-threading is not utilized. Figure 3.4 depicts results for HD-Ready resolution. Clearly, the decoding performance is affected by the number of views and the test sequence. The results are in number of 3D scenes per second. Notice that frames from all views are used in order to generate a 3D scene for a time instant. In other words, 240 monoscopic frames should be decoded per second for playing an 8-view display at 30 fps. One important remark is that, these results are only for

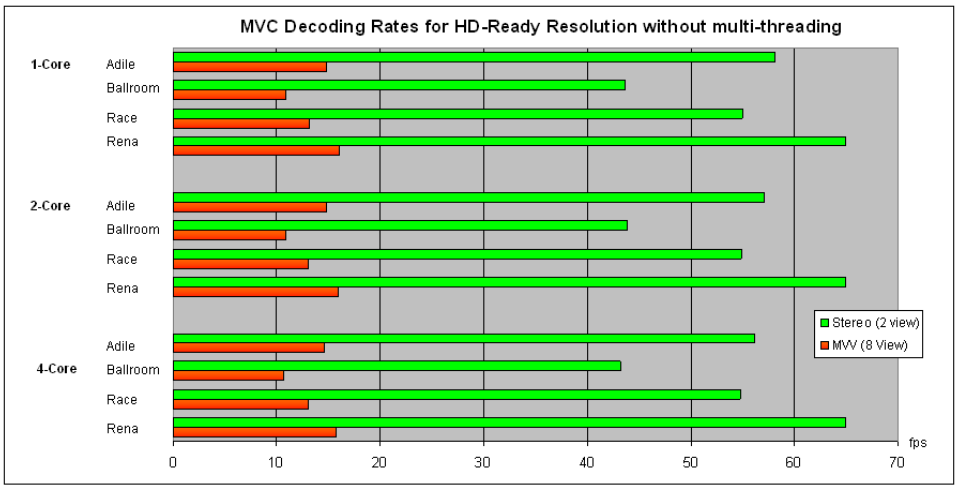


Figure 3.4: MVC Decoding rates for HD-Ready resolution without multi-threading

MVC decoding operations. In most display systems post processing is required for decoded frames to generate a 3D scene. Since the time required for this operation depends on the display, we have not accounted these operations. But in order to provide time for post processing the decoder should be faster than the display rate. Finally, for the CIF and SD resolutions single threaded decoding can deliver at 30 fps.

3.1.3.2 What is the performance gain with the proposed architectures?

Significant performance gain is achieved when the proposed architectures are used in multi-core platforms. Table 3.2 shows the decoding rate for all cases and Figure 3.5 depicts the results of 8-view display at HD-Ready resolution. These results show that addition of second core yields almost linear performance gain. Further addition of two cores improves the performance significantly as well, and the decoding rate reaches three times of what it has been without multi-threading. Moreover, it is possible to extend the encoding schemes for 8-core platforms to further increase the achieved rates for more powerful platforms.

3.1.3.3 What is the multi-threading overhead for platforms with single-core?

It is possible that an application cannot determine the number of cores present in the system due to user restrictions or the operating system depended configurations. Similarly in video

Table 3.1: MVC decoding rates without multi-threading (fps)

		Platform											
		Single-Core				Dual-Core				Quad-Core			
Content		Adile	Ballroom	Race	Rena	Adile	Ballroom	Race	Rena	Adile	Ballroom	Race	Rena
HDR	8-view	15.8	12.99	10.74	14.63	15.96	13.07	10.88	14.8	16.1	13.1	10.92	14.86
	2-view	64.92	54.74	43.19	56.17	64.93	54.88	43.88	57.05	64.99	54.94	43.59	58.11
	1-view	131.05	112.87	87.51	115.62	131.91	113.01	88.36	117.44	131.81	113.42	88.49	117.75
SD	8-view	58.18	45.29	39.6	52.87	58.24	45.51	39.8	52.75	58.41	45.9	39.7	53.75
	2-view	241.68	191.51	161.14	221.9	241.35	191.73	150.76	223.14	242.04	192.26	150.69	223.24
	1-view	493.75	406.98	337.52	463.54	493.73	408.14	339.04	464.64	495.28	409.09	339.63	464.76
CIF	8-view	202.14	149.5	143.26	188.37	202.21	149.58	144.46	188.76	203.07	149.99	144.87	188.93
	2-view	889.3	721.28	618.49	818.64	892.14	722.28	619.85	823.53	894.87	723.37	622.79	829.89
	1-view	1815.05	1471.44	1312.27	1695.26	1830.65	1472.03	1315.9	1730.11	1846.9	1472.44	1315.27	1730.12

Table 3.2: MVC Decoding with multi-threading (fps)

Content		**Platform**							
		CPU: Dual-Core / Stream: Dual				CPU: Quad-Core / Stream: Quad			
Res	# of view	Adile	Ballroom	Race	Rena	Adile	Ballroom	Race	Rena
HDR	8-view	28.84	24.04	20.25	26.01	46.098	40.71	35.78	43.88
	2-view	106.68	89.18	74.45	100.69	102.34	86.28	79.66	105.12
	1-view	130.84	109.85	89.81	120.51	125.86	107.9	90.24	122.26
SD	8-view	101.73	83.05	76.37	101.85	157.84	135.59	129.77	162.61
	2-view	381.12	318.53	293.1	388.26	375.87	303.35	290.14	386.67
	1-view	483.34	393.61	346.12	476.28	483.99	390.46	339.17	460.97
CIF	8-view	375.68	283.81	267.71	362.69	552.93	420.06	464.13	573.41
	2-view	1411.5	1106.1	1042.7	1408.3	1367.2	1075.2	1081.6	1408.3
	1-view	1790.8	1356.9	1258.8	1781.5	1801.4	1345.2	1296.8	1757.3

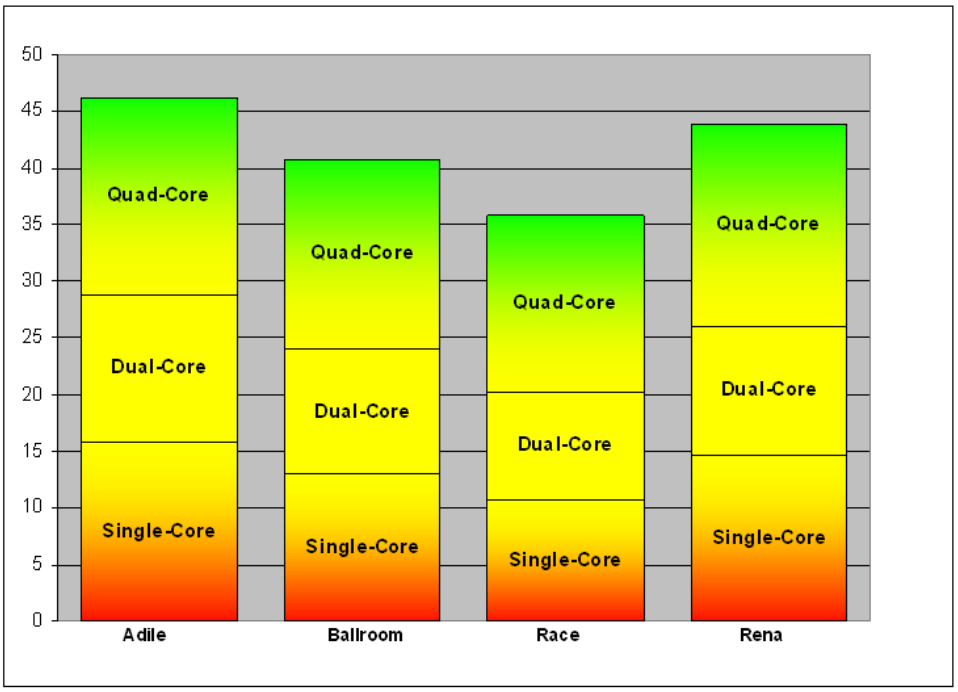


Figure 3.5: Decoding rates for autostereoscopic display at HD-Ready resolution using multi-threading

streaming the server has limited information about the client. In such cases it would be good to be able to decode a stream that is intended for multi-core platforms using single-core platform without significant loss in decoding efficiency.

There are two major reasons for performance loss, multi-threading overhead and redundant decoding. In the proposed multi-threaded architecture some frames are decoded multiple times to provide independence among streams. Such operations become redundant if only single core is present and degrades the decoding performance. In order to investigate the effect of redundancy we have decoded streams that are intended for quad-core platforms using a single-cored PC. Table 3.3 presents decoding rates for such streams and also provides the case when correct stream is used to make an easy comparison. The results reveal that enabling multi-threading for single-core platforms can generate a loss of up to %5 decoding performance.

Table 3.3: Effect of enabling multi-threading over single-cored platforms (fps)

Content		Adile		Ballroom		Race		Rena	
Res	# of view	Quad	Single	Quad	Single	Quad	Single	Quad	Single
HDR	8-view	14.78	15.8	12.31	12.99	10.02	10.74	14.24	14.63
	2-view	61.75	64.92	51.27	54.74	41.19	43.19	54.42	56.17
SD	8-view	55.46	58.18	44.07	45.29	38.97	39.6	51.91	52.87
	2-view	231.97	241.68	183.76	191.51	152.27	161.14	206.66	221.9
CIF	8-view	200.36	202.14	143.42	149.5	134.09	143.26	185.08	188.37
	2-view	850.45	889.3	665.21	721.28	560.29	618.49	759.01	818.64

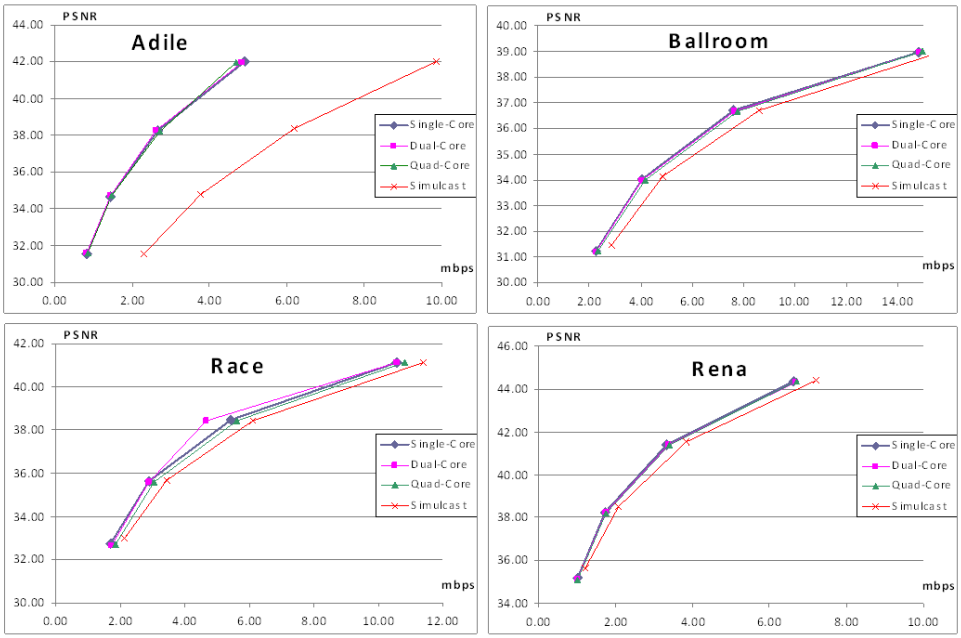


Figure 3.6: RD curves for MVV content for multi-threaded encoding schemes

Table 3.4: Encoding efficiency of the proposed schemes relative to simulcast approach

	Proposed (Single Core)	Proposed (Dual Core)	Proposed (Quad Core)
Adile	+ 5.40 dB	+ 5.47 dB	+ 5.66 dB
Ballroom	+ 0.62 dB	+ 0.59 dB	+ 0.50 dB
Race	+ 0.69 dB	+ 0.96 dB	+ 0.43 dB
Rena	+ 0.56 dB	+ 0.51 dB	+ 0.48 dB

3.1.3.4 What is the RD performance of multi-threaded encoding schemes?

We propose different prediction schemes based on the number of cores of the target platform and but it has minor effect on the encoding performance. In the dual-core case the only requirement is to change the view ID of the independent view, which can benefit or hurt the encoding efficiency with the same probably. For the quad-core case however, we define dependencies for views that are not adjacent to each other. The rate distortion performances of the sequences are given in Figure 3.6. The results show that the cost of defining proposed dependencies has minor effect over the encoding efficiency. The results are also shown according to BD-rates compared with simulcast approach as described in [63] in Table 3.4.

3.1.3.5 Comparison with an Alternative Solution

The performance comparison is performed in terms of decoding speed and coding efficiency, assuming that there exists a single core for each view. In [59], authors have presented their results as the required increase in the speed of the decoder in order to achieve the same frame rate for the case of standard monocular video. For example 'x8' refers to a case in which the decoder should be 8 times faster. Clearly a smaller value refers to a better case.

The prediction structure used in [59] is given below as the list of views and their respective dependencies in parentheses:

View 0(-), 1(0,2), 2(0), 3(2,4), 4(2), 5(4,6), 6(4), 7(6)

In [59], number of macroblock rows required for decoding a view is a parameter which is sent through SEI messages. In their work this number is selected as 2. So in order to start decoding view 2, decoder needs to wait until first 2 rows of view 0 is decoded. If we call this duration as δ (for an image height of 480, 2 rows corresponds to 32 pixels and 32/480 of a frame time).

For 2-core systems, decoding of views are distributed to cores according to decoding order as follows:

Core-0: View 0,1,3,5

Core-1: View 2,4,6,7

In order to start decoding View 2, Core-1 needs to wait δ duration. So total duration for a single frame will be $(4+\delta)$ frame time. With δ=0.07, 2-core system requires 4.07 times faster decoder.

Similarly for 4-core system distribution is as follows:

Core-0: View 0,3

Core-1: View 2,6

Core-2: View 1,5

Core-3: View 4,7

Table 3.5: The theoretical decoding speed requirement depending on the number of cores

# Core	Proposed (GOP Size 12)	Proposed (GOP Size 16)	Standard
2	4.08	4.06	4.07
4	2.08	2.06	2.13
8	1.08	1.06	1.27

In order to start decoding View 2, Core-1 needs to wait δ duration. In order to start decoding View-1, Core-2 needs to wait $2 * \delta$ duration. Similarly, Core-3 needs to wait similarly $2 * \delta$ duration. So total duration for a single frame will be $(2 + 2 * \delta)$ frame time. 4-core system requires 2.13 times faster decoder.

For 8-core system, every core will decode a single view. For this system, we will denote core numbers as the corresponding view numbers. Each core will wait until required number of macroblocks are ready from the required views. Core-2 will wait for δ duration for View-0. Core-1 and Core-4 will wait for $2 * \delta$ duration for View-2. Core-3 and Core-6 will wait for $3 * \delta$ for View-4. Core-5 and Core-7 will wait for $4 * \delta$ for View-6. So total duration for a single frame will be $(1 + 4 * \delta)$ frame time. 8-core system requires 1.27 times faster decoder.

In order to find the required increase in the MVC decoder for our case, we have to use the ratio of overall decoding operations to number of obtained 3D scenes as in Equation 3.1. For example, if ten frames are decoded in parallel to generate five 3D scenes, then the decoder should be twice as fast. In our case the redundancy is decoding one I-frame for each GOP. Therefore, we use GOP size as a parameter in our calculations it is defined as in Equation 3.2. Table 3.5 presents comparison of the derived results between our theoretical solution, which is based on Equation 3.2 and the standard solution based on the methods defined in [59].

$$requiredRateOfIncrease = \frac{totalDecodingOperations}{usefulDecodingOperations} \tag{3.1}$$

$$requiredRateOfIncreaseProposed(GOPSize) = \frac{1 + ViewsPerCorexGOPSize}{GOPSize} \tag{3.2}$$

When we compare the observed rates in Table 3.2 and the theoretical rates in Table 3.5 we see that there is a strong correlation but as expected the observed rate is slower than the theoretical rate. As far as we can identify there are three major reasons for this behavior; i) the throughput

Table 3.6: Simulations results vs theoretical calculations

# Core	Proposed (observed)	Proposed (theoretical)	Standard(theoretical)
1	8.99	8	8
2	4.9	4.06	4.07
4	3	2.06	2.13

Table 3.7: Encoding efficiency of the schemes relative to simulcast approach

	Proposed (Quad-Core)	Standard
Ballroom	+ 0.50 dB	+ 1.17 dB
Race	+ 0.43 dB	+ 0.92 dB

of the multi-core processors are not exactly linear with the number of cores and that generates an upper bound. ii) It takes a longer time to decode a reference picture than predicted one but in the formulations the weight of all frames are equal. This is the case for both the theoretical study of the proposed and the standard solution. iii) The threads have to be synchronized periodically. In our solution this period is once in a GOP while in the proposed solution it is in the order of macroblocks. Based on these facts we believe that, if the standard solution is implemented on an optimized MVC decoder, it would run much slower than the theoretical expectations.

Encoding efficiency for the proposed scheme is compared against [59] using difference PSNR using the method described in [63] relative to the simulcast coding. Test conditions used are advised by the JVT standardization committee [62]. Simulcast is encoded with the same settings but without inter-view prediction. Table 3.7 summarizes the results for joint set of test sequences of both proposed approach and [59] and show that both method has similar performance.

3.1.4 Conclusions

From the experimental results, we can clearly see that single-threaded decoding is not adequate for MVV display (8-view) @ HD-Ready resolutions. In order to solve the problem, adding multiple cores to the system does not provide any performance gain as well using the standard encoding structures. However by using our proposed architecture, we achieved 2 times performance for dual-core systems and 3 times performance for quad-core systems.

Loss in the encoding efficiency is not significant compared with the solution proposed in [59] and the advantage of the proposed system is its simplicity and ease to implement.

CHAPTER 4

Streaming of 3D over Internet

In this chapter, problems and solutions for streaming 3D over internet is addressed.

4.1 End-to-end Stereoscopic Video Streaming with Content-Adaptive Rate and Format Control

In this section an end-to-end stereoscopic video streaming system using content-adaptive rate and format control is introduced. Efficient compression and real-time streaming of stereoscopic video over the current Internet is addressed in this work. In order to implement this system, we introduce content-adaptive stereo video coder (CA-SC) based on asymmetric stereoscopic video coder explained in Section 2.2. We also developed stereoscopic 3D video streaming server and clients by modifying available open source platforms, where each client can view the video in mono or stereo mode depending on its display capabilities.

Recently, a 3DTV prototype system, similar to our system, with real-time acquisition, transmission and autostereoscopic display of dynamic scenes, has been offered by MERL. Multiple video streams are encoded and sent over a broadband network. The 3D display shows high-resolution stereoscopic color images for multiple viewpoints without special glasses. This system uses light-field rendering to synthesize views at the correct virtual camera positions [23].

The proposed content-adaptive stereo encoder (CA-SC) is motivated by the suppression theory and reduces the spatial resolution and/or the frame (temporal) rate of the target (right) sequence adaptively according to its content-based features. The principle behind content adaptive video coding is to parse video into temporal segments. Each temporal segment

can be encoded at different spatial, temporal and SNR resolution (hence at a different target bitrate) depending on its low and/or high-level content-based features. Even though this approach has been used for monoscopic video encoding [64, 65, 66, 67], there are no such studies in the literature for content-adaptive stereoscopic coding. The proposed CA-SC codec [14] is an extension of the stereo codec (SC) in [25] which is based on H.264/AVC explained in Section 2.2. We note that CA-SC can also be implemented as an compliant way using the recently standardized MVC codec [11].

Spatial Scene Complexity and Temporal Activity measures are used for the classification of temporal segments. According to the classification, spatial, temporal or spatio-temporal downsampling is applied to right video. Motion vector statistics and Pixel Variance are used to for classification. More details can be found in [15].

The streaming system is based on open source platforms with the following modifications. Encoder is a modified version of H.264/AVC reference software. Encoded bitstream is encapsulated into RTP/UDP/IP packetization. In order to establish a streaming channel between server and client Session Description Protocol (SDP) is used and modified to signal for the stereoscopic content. In order to decode stereoscopic video in real-time, FFMPEG H.264/AVC decoder is modified in order to decode bitstream encoded by [14]. VideoLAN Media Player (VLC) is used as client with the modified FFMPEG decoder and modified display modules.

4.2 A Standards-Based, Flexible, End-to-End Multi-View Video Streaming Architecture

In order to improve previously introduced streaming system, we propose a standards-based, flexible, end-to-end MVV streaming architecture in [32, 68] as shown in Figure 4.1. This system supports several different display types including monoscopic, autostereoscopic displays (2-view or 8-view) and stereoscopic displays with anaglyph glasses. System also supports VpD encoded streams for single view and 2-view displays. Modifications for SDP are extended for multi-view video with the newly added dependency lists. Encoder is updated to use standard MVC reference software (JMVM 3.0.2) and decoder is modified FFMPEG decoder compliant with this encoder. In order to cope with losses, multi-view error concealment is also

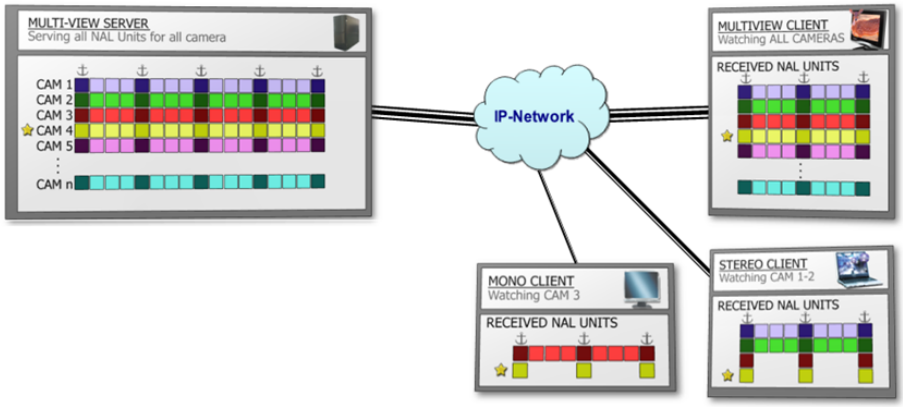


Figure 4.1: Standards-Based, Flexible, End-to-End Multi-View Video Streaming Architecture.

implemented as explained in [69]. Encoded bitstream is sent through several ports classified according to view id and picture type (anchor/non-anchor). Depending on the required views, only required set of ports are connected by the client as shown in Figure 4.1. This enables adaptation of MVC stream on the fly depending on the client capabilities. MVV decoding for 8-views in real-time requires multi-threading on multi-core systems as explained in Chapter 3, therefore multi-threaded decoding is also implemented for the client application. In this version instead of VLC, new client application is developed with required functionalities.

4.3 Rate-Distortion Optimization for Stereoscopic Video Streaming with Unequal Error Protection

Existing stereoscopic techniques compress the data by exploiting the dependency between the left and right views; however, the compressed video is more sensitive to data losses and needs added protection against transmission errors. To make matters more complicated, the rate of packet losses in the transmission channel is typically time-varying. Hence, one faces a difficult joint source-channel coding problem where the goal is to find the optimal balance between the distortion created by lossy source compression and the distortion caused by packet losses in the transmission channel. We address this problem by (i) proposing a heuristic methodology for modeling the end-to-end RD characteristic of such a system, and (ii) dynamically adjusting the source compression ratio in response to channel conditions so as to minimize the overall distortion.

As opposed to stereoscopic video streaming, various studies exist in the literature for layered or non-layered monoscopic video on optimal rate allocation and error resilient streaming on error prone channels such as Packet Erasure Channel (PEC). The early studies on monoscopic video streaming mainly concentrate on non-layered video and the optimal bit control and bit rate allocation for the video elements [70, 71, 72, 73]. RD optimization is the most widely used optimization method for the quality of video and it is a mechanism that aims to calculate optimal redundancy injection rate into the network while adapting the video bit rate accordingly in order to match the available bandwidth estimate. Redundancy may be generated by means of either retransmissions or forward error correction (FEC) codes and this redundancy is used to minimize the average distortion resulting from network losses during a streaming session [74, 75, 76, 77]. Even though retransmission methods can be used in video streaming applications as in [78], it may bring large latency for video display. On the other hand, FEC schemes insert protection before the transmission and do not utilize retransmissions. In literature, FEC methods are studied for video streaming as in [79], [80] and [81].

A novel technique that recently becomes popular for error protection in lossy packet networks is Fountain codes, also called rateless codes. The Fountain coding idea is proposed in [82] and followed by practical realizations such as LT codes [83], online codes [84] and Raptor codes [85]. Following the practical realizations, Fountain codes have gained attention in video streaming in recent years [86], [87], [88]. The main idea behind Fountain coding is to produce as many parity packets as needed while streaming. This approach is different than the general idea of FEC codes where channel encoding is performed for a fixed channel rate and all encoded packets are generated prior to transmission. The idea is proven to be efficient in [83] for large source data sizes, as in the case of video data, and does not utilize retransmissions.

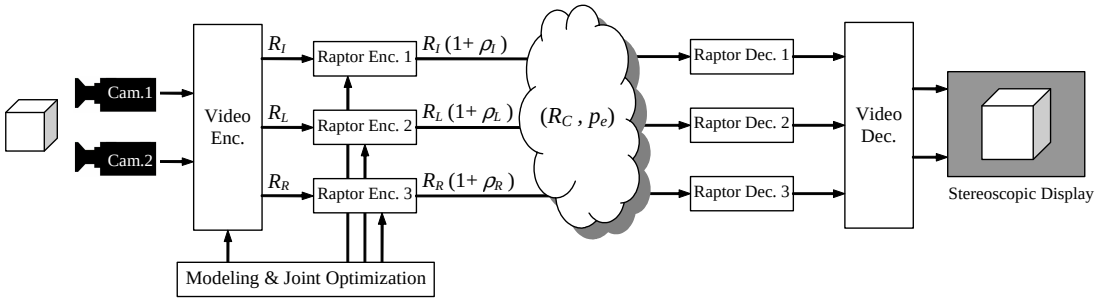


Figure 4.2: Overview of the stereoscopic streaming system

Due to a more intense prediction structure, stereoscopic video, the main focus of this work, is

more prone to packet losses compared to monoscopic video. Inter-dependent coding among views may result in quality distortion for both views if a packet from one view is lost. Even though FEC codes and optimal bit rate allocations are studied in depth for monoscopic video streaming, only a few studies exist for stereoscopic video streaming [89]. In [89], stereoscopic video is layered using data partitioning but an FEC method specific to stereoscopic video is not used. We aim at filling the gap in the literature on optimal error resilient streaming of stereoscopic video.

An overview of our proposed stereoscopic streaming system is presented in Fig. 4.2. Initially, the scene of interest has to be captured with two cameras to obtain the raw stereoscopic video data. The video capture process is not in the scope of our work, thus we use publicly available raw video sequences. We encode the raw stereoscopic video data with an H.264 based multi-view video encoder. We use the codec in stereoscopic mode and generate three layers which are denoted with the symbols I, L and R. I-frames are the intra-coded frames of the left view, L and R-frames are the inter-coded frames of the left view and right view. The video encoder can encode each layer with different quantization parameters, thus with different bit rates R_I, R_L and R_R. Due to lossy compression, the encoding process causes a distortion of D_e in the video quality. After the stereoscopic encoder, we apply FEC to each layer separately where we use Raptor codes as the FEC scheme. The channel of interest in our system is a packet erasure channel of loss rate p_e and the available bandwidth of the channel is R_C. We apply different protection rates ρ_I, ρ_L and ρ_R to each layer, because they contribute differently to the video quality. After the lossy transmission, some of the packets are lost and Raptor decoder operates to recover the losses. However, some packets still may not be recovered and the loss of these packets causes a distortion of D_{loss} in the video quality. In this system, our goal is to obtain the optimal values of encoder bit rates R_I, R_L and R_R and protection rates ρ_I, ρ_L and ρ_R by minimizing the total distortion $D_{tot} \triangleq (D_e + D_{loss})$. In order to execute the minimization, we obtain the analytical models of each part of our system. We start with the modeling of the RD curve of each layer of the stereoscopic video encoder. Then, we define the analytical model of the performance of Raptor codes. Finally, we estimate the distortion on the stereoscopic video quality caused by packet losses.

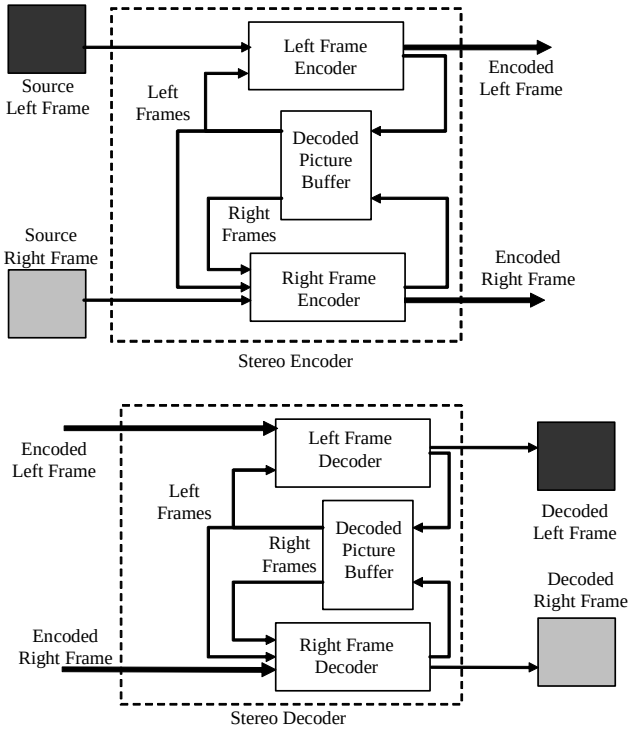


Figure 4.3: Stereoscopic encoder and decoder structure

4.3.1 Stereoscopic Codec

The general structure of a stereoscopic encoder and decoder is given in Fig. 4.3. In order to maintain backward compatibility to monoscopic decoders, left frames are encoded with prediction only from left frames, whereas right frames are predicted using both left and right frames. This enables standard monoscopic decoders to decode left frames.

Any video codec with this basic structure can be used with the proposed streaming system in this work. Multi-View Extension of H.264 standard [90] (JMVM software) is one of the candidate codecs for this work. However, hierarchical B picture coding used in this codec increases the complexity. In order to decrease complexity and simplify decoding procedure, we have used [25] explained in Section 2.1.1 with the structure given in Fig. 4.3. The results can easily be extended for JMVM codec.

The referencing structure of the codec in [25] is given in Fig. 4.4, where we set the GOP size to 4. Let $\mathbf{I}_L, \mathbf{P}_L$ and $\mathbf{P}_R$ denote the set of I-frames of left view, P-frames of left views

and P-frames of right views respectively. The set of frames can be written in open form as $\mathbf{I}_L = \{I_{L1}, I_{L5}, ...\}$, $\mathbf{P}_L = \{P_{L2}, P_{L3}, ...\}$, $\mathbf{P}_R = \{P_{R1}, P_{R2}, ...\}$, where L and R indicate the frames of left and right video.

Although this coding scheme is not layered, frames are not equal in importance. We can classify the frames according to their contribution to the overall quality and use them as layers of the video. Since losing an I-frame causes large distortions due to motion / disparity compensation and error propagation, I-frames should be protected the most. Among P-frames, left frames are more important since they are referred by both left and right frames. According to this prioritization of the frames, we form three layers as shown in Fig. 4.4. Layers can be coded with different quality (bit rate) by using either spatial scaling as explained in Section 2.2 or quantization. We used quantization parameter to adjust the quality of different layers.

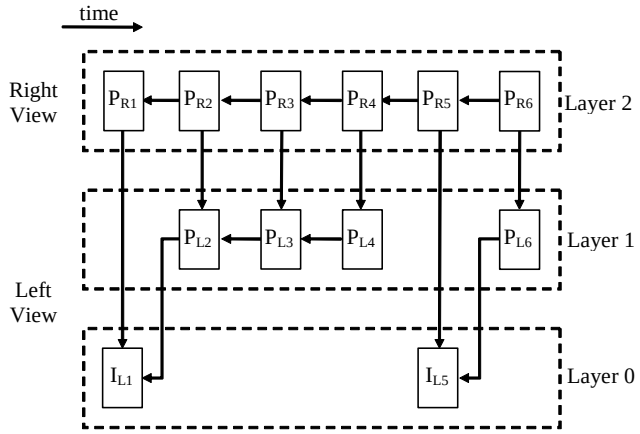


Figure 4.4: Layers of stereoscopic video and referencing structure

In the case of slice losses in transmission, we employ different error concealment techniques for different layers in the decoder. For layer 0, since there is no motion estimation, we use spatial concealment based on weighted pixel averaging [91]. For layer 1, we use temporal concealment. Co-located block from the previous layer-1 frame is used in place of the lost block. For layer 2, we use temporal concealment but with a slight modification. In this case, co-located block can be taken either from previous layer-2 frame or from the layer-1 frame from the same time index. Depending on the neighboring blocks motion vectors, appropriate frame is selected and co-located block from the selected frame is used in the place of the lost block.

4.3.2 Analytical Model of the RD Curve of Encoded Stereoscopic Video

In this subsection, we model the RD curve of stereoscopic video. The RD curve of video is widely used for optimal streaming purposes [74, 75, 76, 77], which provides the optimal streaming bit rate for a given distortion in video quality and vice versa. In [92], a simple analytical RD curve model that can accurately approximate a wide range of monoscopic video sequences is presented. The model in [92] has the form

$$D_e(R) = \frac{\theta}{R - R_0} + D_0 \tag{4.1}$$

where $D_e(R)$ is the mean-squared error (MSE) at the video encoder output at the encoding rate of R bits/sec. There are 3 parameters to be solved which are θ, R_0 and D_0. The parameters R_0 and D_0 do not correspond to any rate or distortion values and they are not initial values. At least three samples of the RD curve are required to solve for the parameters.

The proposed analytical model in (4.1) can be used for each layer of video separately as stated in [92]. However, the model is not suitable for the cases when the layers are dependent. In our experiments, when we applied the analytical model in (4.1) separately to each one of our layers we observed that the models were not accurate enough to approximate the RD curve. Thus, the analytical models had to be modified for dependent layers.

We have extended the analytical RD model of monoscopic video proposed in [92] to stereoscopic case. We modified the model in order to handle the dependencies among the layers. The structure of the layers of our stereoscopic codec is described in subsection 4.3.1 and presented in Fig. 4.4. The primary layer is layer 0 (I-frame) which consists of intra frames and it does not depend on any previous frames. Thus, the distortion of layer 0 only depends on the encoder bit rate of layer 0. The second layer is layer 1 whose frames are coded dependent on previous frames of layer 1 and layer 0. Thus, the distortion of layer 1 depends on the encoder bit rates of layer 1 and layer 0. The third layer is layer 2 whose frames are coded dependent on previous frames of layer 2, layer 1 and layer 0. Thus, the encoder distortion of layer 2 depends on the encoder bit rates of all layers. We modeled the RD curves of each layer to include the stated dependencies.

Table 4.1: Encoder RD Curve Parameters for ‘Rena’ Video

Layer 0			θ_I	R_{0I}	D_{0I}
			1.605e+011	6050	-289860
Layer 1		c_1	θ_L	R_{0L}	D_{0L}
		0.616	3.483e+013	51858	6142922
Layer 2	c_2	c_3	θ_R	R_{0R}	D_{0R}
	0.308	0.086	4.535e+013	50000	4056654

Table 4.2: Encoder RD Curve Parameters for ‘Soccer’ Video

Layer 0			θ_I	R_{0I}	D_{0I}
			2.978e+011	10249	120330
Layer 1		c_1	θ_L	R_{0L}	D_{0L}
		0.456	1.513e+014	-23018	2209000
Layer 2	c_2	c_3	θ_R	R_{0R}	D_{0R}
	0.333	0.235	1.496e+014	19482	6003200

4.3.2.1 RD Model of Layer 0

The RD curve model of layer 0 is given in (4.2). Layer 0 is encoded as an independent monoscopic video; hence, we model its RD curve using the same framework as in (4.1) and set the model as

$$D_e^I(R_I) = \frac{\theta_I}{R_I - R_{0I}} + D_{0I} \tag{4.2}$$

Here, $D_e^I(R_I)$ is the MSE coming from layer 0 when layer 0 is allocated a rate of R_I bits/sec. The model parameters are θ_I, R_{0I} and D_{0I}, which have to be solved.

4.3.2.2 RD Model of Layer 1

The next analytical model is realized for layer 1 which consists of predicted frames of left view. As stated previously, the encoder distortion of layer 1 depends on the encoder bit rate of layer 1 and layer 0. We modify the model in (4.1) to handle this dependency as

$$D_e^L(R_L, R_I) = \frac{\theta_L}{R_L + c_1 R_I - R_{0L}} + D_{0L} \tag{4.3}$$

Here, $D_e^L(R_L, R_I)$ is the MSE coming from layer 1 when layer 1 and layer 0 are allocated the rates of R_L and R_I bits/sec, respectively. The model parameters are θ_L, c_1, R_{0L} and D_{0L}, which also have to be solved. The term $c_1 R_I$ in the denominator is inserted to handle the dependency of the distortion of layer 1 to layer 0 where the encoder bit rate of layer 0 is weighted with the parameter c_1.

4.3.2.3 RD Model of Layer 2

The last analytical model is realized for layer 2 which consists of the frames of right view. Since the distortion of layer 2 is dependent on all layers, the analytical model has to include the encoder bit rates of all layers. We modify the model in (4.1) to handle this dependency as

$$D_e^R(R_R, R_L, R_I) = \frac{\theta_R}{R_R + c_2 R_I + c_3 R_L - R_{0R}} + D_{0R} \tag{4.4}$$

Here, $D_e^R(R_R, R_L, R_I)$ is the MSE coming from layer 2 when layer 2, layer 1 and layer 0 are allocated the rates of R_R, R_L and R_I bits/sec, respectively. The model parameters are θ_R, c_2, c_3, R_{0R} and D_{0R}, which also must be solved. The terms $c_2 R_I$ and $c_3 R_L$ in the denominator are inserted to handle the dependency of layer 2 to layer 0 and layer 1.

4.3.2.4 Results on RD Modeling

In order to construct the RD curve models of stereoscopic videos, i.e., to obtain the model parameters, we used curve fitting tools. We used the stereoscopic videos 'Rena' and 'Soccer' explained in subsection 4.3.5.2 and obtained the RD curve models of these videos for the analytical models in (4.2) to (4.4). We used a general purpose non-linear curve fitting tool which uses the Levenberg-Marquardt method with line search [93]. Before the curve fitting operation we obtained many RD curve samples of the video by sweeping the quantization parameters of each layer from low to high quality. We obtained more RD samples than required in order to be able to observe the curve fitting performance. Then, we chose some of the RD samples and inserted into the curve fitting tool. The resulting analytical model parameters of the curve fit process are given in Tables 4.1 and 4.2 for the chosen videos. The parameters are in accordance with the properties of the videos. 'Rena' has static background with moving

objects and 'Soccer' has a camera motion. Since the 'Soccer' video has a camera motion, while encoding a right frame, correlation with the current left frame can be more than the previous right frame. This shows why the c_3 parameter of layer 2 of the 'Soccer' video is high when compared with the results of the 'Rena' video.

In Figs. 4.5 to 4.10, we present the results of analytical modeling of the RD curves. In Figs. 4.5 and 4.6, we give the results for layer 0 where the analytical models are constructed using the model in (4.2) with the corresponding parameters from Tables 4.1 and 4.2. The RD samples correspond to the actual RD values obtained from the video encoder before the curve fitting process. Later, the results for layer 1 are presented in Figs. 4.7 and 4.8 and those of layer 2 are presented in Figs. 4.9 and 4.10. In the figures for layer 1 and 2, we present two cross-sections of the RD curves. The cross sections are obtained by fixing the encoder bit rates of the layers other than the corresponding layer of interest. The average difference between analytical models and RD samples for the 'Rena' video are 3.62%, 7.60% and 9.19% for layer 0,1 and 2 respectively, and those of the 'Soccer' video are 1.00%, 5.87% and 8.89%. Thus, for both of the videos, which have different characteristics, satisfactory results are achieved where the analytical model approximates the RD samples accurately.

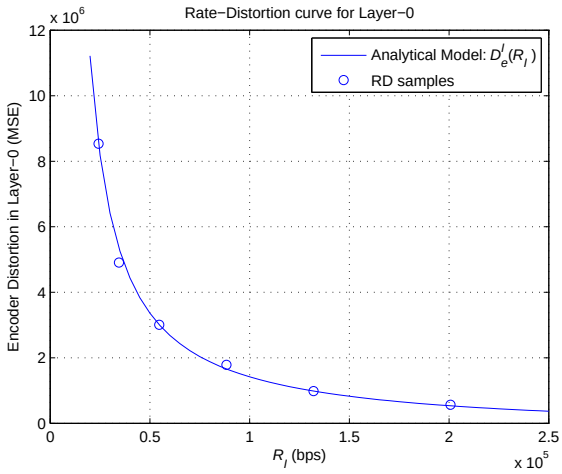


Figure 4.5: RD curve for layer 0 of the 'Rena' video

4.3.3 Analytical Modeling of The Performance Curve of Raptor Codes

Analytical model of the performance curve of Raptor codes is modeled heuristically as defined in [17]. This model is going to be used for the derivation of optimal parity packet allocation

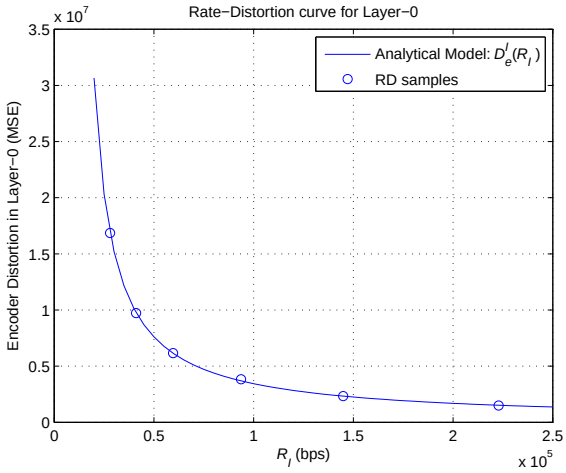


Figure 4.6: RD curve for layer 0 of the 'Soccer' video

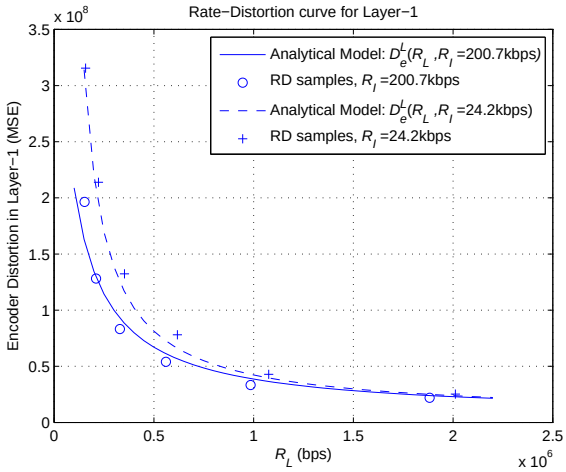


Figure 4.7: RD curve for layer 1 of the 'Rena' video

to layers in Section 4.3.5 in the end-to-end distortion minimization. The analytical model is defined as

$$N_u(N_i, N_r, \rho) = \begin{cases} N_i - \frac{N_r}{(1+\rho)} & N_r \leq N_i \\ N_i \frac{\rho}{(1+\rho)} 2^{(N_i - N_r)} & N_r > N_i \end{cases} \tag{4.5}$$

In (4.5), $N_u(N_i, N_r, \rho)$ is the analytical model of the number of undecoded input symbols which is a function of N_i, N_r and ρ.

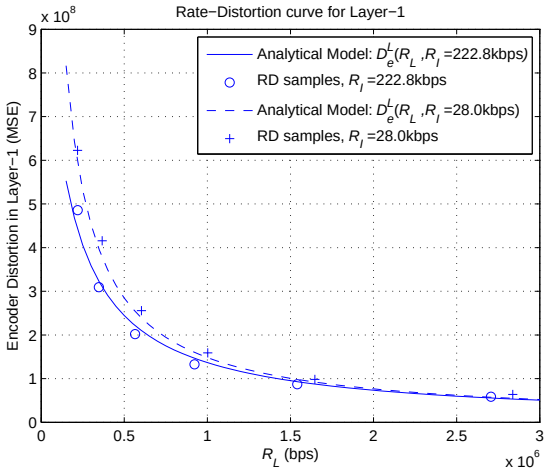


Figure 4.8: RD curve fit for layer 1 of the 'Soccer' video

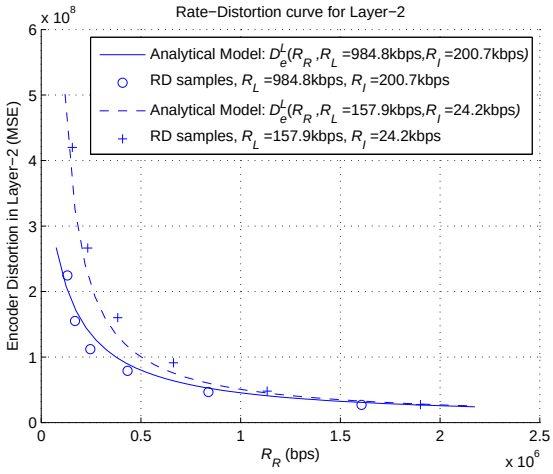


Figure 4.9: RD curve for layer 2 of the 'Rena' video

4.3.4 Estimation of Transmission Distortion

In this subsection, our aim is to estimate the loss distortion in video remaining after the Raptor decoder and stereoscopic video decoder (D_{loss}). We explain the estimation of residual loss distortion step by step.

4.3.4.1 Lossy Transmission

The channel of interest in our work is PEC as mentioned previously. During the transmission of stereoscopic video layers from PEC, Network Abstraction Layer (NAL) units are lost with

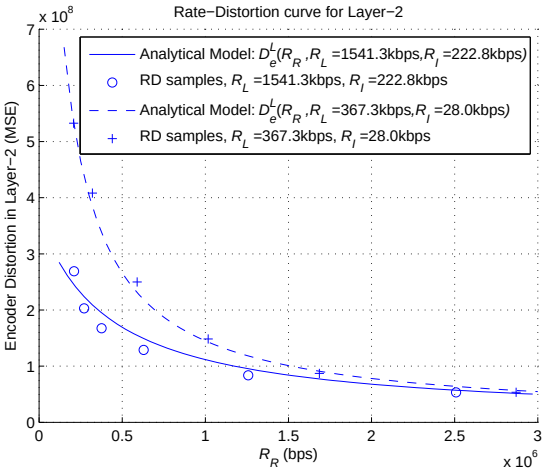


Figure 4.10: RD curve for layer 2 of the 'Soccer' video

probability p_e. In the remaining part of our work, for simplicity, X will represent the layer denotations I, L and R. We have three layers of video with source bit rate R_X which are Raptor encoded separately with inserted parity rate ρ_X. Thus, $N_i^X(1+\rho_X)$ output symbols are created and transmitted for each layer. After lossy transmission, the number of received output symbols in Raptor decoder can be calculated as

$$N_r^X = N_i^X(1+\rho_X)(1-p_e) \tag{4.6}$$

Here, we use the average loss probability for simplified modeling purposes only. The experimental results in Section 4.3.5.2 reflect the actual distortions over lossy channels where a single packet is lost with probability p_e.

After receiving N_r^X output symbols Raptor decoder operates to solve for the input symbols. We use the model of the performance curve of Raptor codes to obtain the average number of undecoded input symbols using (4.5). The average number of undecoded input symbols (the residual number of lost NAL units) can be calculated as

$$N_u^X = N_u\left(N_i^X, N_r^X, \rho_X\right) \tag{4.7}$$

4.3.4.2 Propagation of Lost NAL units in Stereoscopic Video Decoder

Due to the recursive structure of the video codec, the distortion of a NAL unit loss not only causes distortion in the corresponding frame but it also propagates to subsequent frames in the video. Initially, since each NAL unit contains a specific number of macroblocks (MBs), we estimate the distortion in a frame when a single MB is lost . The distortion is calculated after error concealment techniques, explained in subsection 4.3.1, are applied for the lost MB. Then, we calculate the average propagated distortion of a single MB and, consequently, a NAL unit.

In [92], a model for distortion propagation is proposed where the propagated error energy (distortion) at frame t after a loss at frame 0 is given as

$$\sigma_u^2(t) = \frac{\sigma_{u0}^2}{1+\gamma t} \tag{4.8}$$

Here, σ_{u0}^2 is the average distortion per lost unit, and γ is the leakage factor which describes the efficiency of the loop filtering in the decoder to remove the introduced error ($0 < \gamma < 1$). We assume $\gamma \approx 0$ which results in worst case propagation where the distortion propagates equally to all subsequent frames ($\sigma_u^2(t) = \sigma_{u0}^2$). In the following paragraphs, we calculate the propagated NAL unit loss distortion for each layer separately where we set MBs as the video unit.

NAL unit loss from Layer 0 The expression in (4.9) gives the average distortion of spatial error concealment when a lost MB is concealed by the average of its neighboring MBs. In (4.9), S_{MB}, MB_i, $S_{\mathrm{MB},i}$, N_i' and N_{MB}^I represent the set of macroblocks, the i^{th} macroblock, the set of i^{th} MB's neighbors, the number of neighbors of i^{th} MB and the number of MBs of layer 0 respectively. $\mathrm{I_I}(x, y, 0)$ denotes the pixel in position (x, y) of the intra frame of layer 0. Layer 0 consists of a single intra frame, thus only spatial error concealment can be used due to intra coding as described in Section 4.3.1.

$$\sigma_{I0}^2 = \frac{1}{N_{MB}^I} \sum_{k \in S_{MB}} \left[\sum_{x,y \in MB_k} \left(\mathrm{I_I}(x,y,0) - \sum_{x',y' \in MB_k'} \mathrm{I_I}(x',y',0) / N_k' \right)^2 \right] \tag{4.9}$$

In Fig. 4.11, the propagation of an MB loss in an I-frame is demonstrated. The black box in the frame $\mathrm{I_{L1}}$ represents a possible loss in the I-frame. The loss causes a distortion of σ^2_{I0} as calculated in (4.9) for the frame $\mathrm{I_{L1}}$. The loss propagates to all subsequent frames with equal distortion on the average since both L-frames and R-frames refer initially to the I-frame. If we denote the GOP size as T, then the average of total propagated loss distortion when an MB is lost from layer 0 can be calculated as

$$D^I_{\mathrm{MB}prop} = 2\mathrm{T}\sigma^2_{I0} \tag{4.10}$$

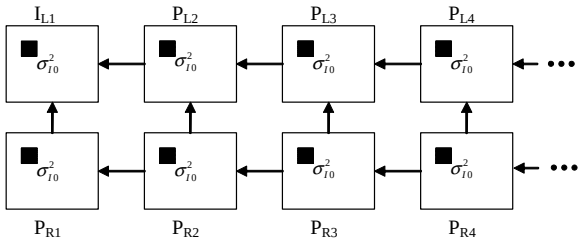


Figure 4.11: Propagation of a MB loss from I-frame

In order to calculate the average distortion of losing a NAL unit from layer 0 ($D^I_{\mathrm{NAL}loss}$), we have to calculate the average number of MBs in a NAL unit. Let N^I_{MB} denote the number of MBs in layer 0. Then, $D^I_{\mathrm{NAL}loss}$ can be calculated as

$$D^I_{\mathrm{NAL}loss} = \left(\frac{N^I_{\mathrm{MB}}}{N^I_i}\right) \cdot D^I_{\mathrm{MB}prop} \tag{4.11}$$

NAL unit Loss from Layer 1 The expression in (4.12) gives the average distortion of temporal error concealment when a lost NAL unit is concealed from the previous frame of layer 1. In (4.12), N^L_{MB} and T represent the number of MBs of layer 1 and GOP size respectively. $\mathrm{I_L}(x, y, i)$ denotes the pixel in position (x, y) of i^{th} frame of layer 1. Layer 1 consists of predicted frames of left view. In our stereoscopic codec, we used temporal error concealment for layer 1 as described in Section 4.3.1.

$$\sigma^2_{L0} = \frac{\frac{1}{\mathrm{T}-1}\sum_{i=1}^{\mathrm{T}-1}\sum_{x,y}\left[\mathrm{I_L}(x, y, i) - \mathrm{I_L}(x, y, i-1)\right]^2}{N^L_{\mathrm{MB}}} \tag{4.12}$$

In Fig. 4.12, the propagation of an MB loss in an L-frame is demonstrated. The black box in the frame P_{L2} represents a possible loss in the L-frame. The loss causes a distortion of σ^2_{L0} as calculated in (4.12) for the frame P_{L2}. The loss propagates to all subsequent L-frames with equal distortion since each L-frame refers to the previous L-frame. Let m denote the frame index of loss in a GOP, then the average propagated loss to L-frames can be calculated as

$$\frac{1}{\mathrm{T}-1}\sum_{m=1}^{\mathrm{T}-1}(\mathrm{T}-m)\,\sigma^2_{L0} \tag{4.13}$$

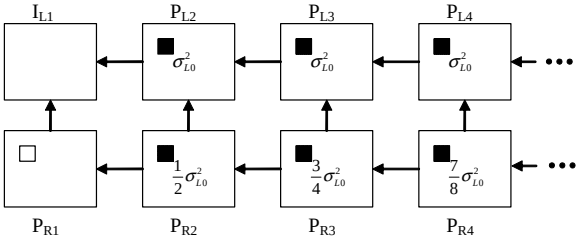


Figure 4.12: Propagation of a MB loss from L-frame

The MB loss also propagates to R-frames. However, R-frames not only refer to current L-frames but also previous R-frames. Due to this fact, the distortion in P_{R2} can be calculated as $\sigma^2_{L0}/2$ using the previous undistorted MB (white box in P_{R1}). In the frame P_{R3} the propagated distortion can be calculated as $\left(\sigma^2_{L0}/2+\sigma^2_{L0}\right)/2 = \frac{3}{4}\sigma^2_{L0}$. In the subsequent frames the propagated distortion is calculated similarly as shown in Fig. 4.12. The average of total propagated distortion in an R-frame caused by the loss of an L-frame MB can be calculated as

$$\frac{1}{\mathrm{T}-1}\sum_{m=1}^{\mathrm{T}-1}\sum_{n=1}^{\mathrm{T}-m}\left(\left(1-\frac{1}{2^n}\right)\sigma^2_{L0}\right) \tag{4.14}$$

Thus, the average of total propagated distortion when an MB is lost from layer 1 can be calculated as

$$D^L_{\mathrm{MB}prop} = \frac{1}{\mathrm{T}-1}\sum_{m=0}^{\mathrm{T}-2}\sum_{n=0}^{m}\left(\left(2-\frac{1}{2^{n+1}}\right)\sigma^2_{L0}\right) \tag{4.15}$$

In order to calculate the average distortion of losing a NAL unit from layer 1 ($D^L_{\mathrm{NAL}loss}$), we have to calculate the average number of MBs in a NAL unit. Let N^L_{MB} denote the number of MBs in layer 1. Then, $D^L_{\mathrm{NAL}loss}$ can be calculated as

$$D^{L}_{\mathrm{NAL}loss} = \left(\frac{N^{L}_{\mathrm{MB}}}{N^{L}_{i}}\right) \cdot D^{L}_{\mathrm{MB}prop} \tag{4.16}$$

NAL unit Loss from Layer 2 The expression in (4.17) gives the average distortion of temporal error concealment when a lost NAL unit is concealed from the frames of layer 2 and layer 1. In (4.17), N^{R}_{MB} and T represent the number of MBs of layer 2 and GOP size respectively. $\mathrm{I_R}(x, y, i)$ denotes the pixel in position (x, y) of i^{th} frame of layer 2. Layer 2 consists of predicted frames of right view. In our stereoscopic codec, we used temporal error concealment for layer 2 where the frames are referred to previous layer 2 and current layer 1 frames as described in subsection 4.3.1.

$$\sigma^{2}_{R0} = \frac{\sum\limits_{x,y}\left[\mathrm{I_L}(x,y,0) - \mathrm{I_R}(x,y,0)\right]^{2}}{(\mathrm{T}-1)N^{R}_{\mathrm{MB}}} + \frac{\sum\limits_{i=1}^{\mathrm{T}-1}\sum\limits_{x,y}\left[\left(\frac{\mathrm{I_R}(x,y,i-1)+\mathrm{I_L}(x,y,i)}{2}\right) - \mathrm{I_R}(x,y,i)\right]^{2}}{(\mathrm{T}-1)N^{R}_{\mathrm{MB}}} \tag{4.17}$$

In Fig. 4.13, the propagation of an MB loss in an R-frame is demonstrated. The black box in the frame $\mathrm{P_{R2}}$ represents a possible loss in the R-frame. The loss in an R-frame propagates only to the subsequent R-frames. A loss in the frame $\mathrm{P_{R2}}$ creates a distortion of σ^{2}_{R0} as calculated in (4.17). In frame $\mathrm{P_{R3}}$, the propagation distortion can be calculated as $\sigma^{2}_{R0}/2$ using the undistorted MB in the L-frame (white box in $\mathrm{P_{L3}}$). In each of the following R-frames the propagated distortion is the half of the previous R-frame. Thus, the average of total propagated distortion when an MB is lost from layer 2 can be calculated as

$$D^{R}_{\mathrm{MB}prop} = \sum_{m=0}^{\mathrm{T}-1}\frac{1}{\mathrm{T}}\sum_{n=0}^{m}\left(\frac{1}{2^{n}}\sigma^{2}_{R0}\right) \tag{4.18}$$

In order to calculate the average distortion of losing a NAL unit from layer 2 ($D^{R}_{\mathrm{NAL}loss}$), we have to calculate the average number of MBs in a NAL unit. Let N^{R}_{MB} denote the number of MBs in layer 2. Then, $D^{R}_{\mathrm{NAL}loss}$ can be calculated as

$$D^{R}_{\mathrm{NAL}loss} = \left(\frac{N^{R}_{\mathrm{MB}}}{N^{R}_{i}}\right) \cdot D^{R}_{\mathrm{MB}prop} \tag{4.19}$$

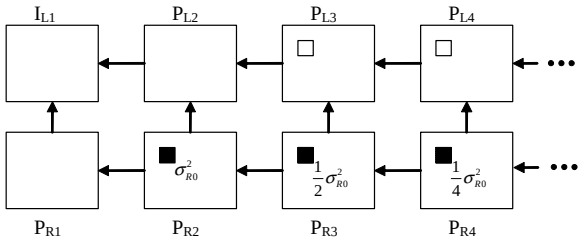


Figure 4.13: Propagation of a MB loss from R-frame

4.3.4.3 Calculation of Residual Loss Distortion

In this part, we calculate the average transmission distortion after Raptor decoder and stereoscopic video decoder. Let D^X_{loss} denote the residual transmission distortion. In (4.20), we calculate D^X_{loss} by multiplying the number of undecoded input symbols with the average distortion of losing a NAL unit.

$$D^X_{loss}\left(R_X, \rho_X, p_e\right) = N_u\left(N^X_i, N^X_r, \rho_X\right) \cdot D^X_{\mathrm{NAL}loss} \tag{4.20}$$

Here, we use the assumption that the NAL unit losses are uncorrelated which is met for low number of losses after the Raptor decoder. Thus, the accuracy of the model may reduce for high loss rates.

4.3.5 End-to-End Distortion Minimization and Performance Evaluation

As the last part of our system, we minimize the total end-to-end distortion to find the optimal encoder bit rates and UEP rates and evaluate the performance of the system. We present the minimization as

$$\begin{aligned} &\min_{(R_I,R_L,R_R,\rho_I,\rho_L,\rho_R)} D_{tot} \\ &\text{s.t. } (1+\rho_I)\,R_I + (1+\rho_L)\,R_L + (1+\rho_R)\,R_R = R_C \end{aligned} \tag{4.21}$$

The minimization aims at obtaining the optimal encoder bit rates R_I, R_L and R_R, and optimal parity ratios ρ_I, ρ_L and ρ_R for given p_e and R_C. The constraint ensures that the final bit

rate satisfies a total transmission bandwidth of R_C including both the encoder bit rates and protection data bit rates. In (4.22), we present the calculation of D_{tot} where $D_e^I(.)$, $D_e^L(.)$ and $D_e^R(.)$ are the encoder distortions defined in (4.2), (4.3) and (4.4), and $D_{loss}^I(.)$, $D_{loss}^L(.)$ and $D_{loss}^R(.)$ are the residual loss distortions defined in (4.20).

$$\begin{aligned} D_{tot} = & \tfrac{1}{3}\left[D_e^R(R_R, R_L, R_I) + D_{loss}^R(R_R, r_r, p_e)\right] \\ & + \tfrac{2}{3}\left[D_e^I(R_I) + D_e^L(R_L, R_I) + D_{loss}^I(R_I, \rho_I, p_e)\right. \\ & \left. + D_{loss}^L(R_L, \rho_L, p_e)\right] \end{aligned} \tag{4.22}$$

Total distortion in left and right frames is weighted to handle the objective stereoscopic video quality as stated in [94]. The weighting parameters in [94] are found by Least Squares Fitting of the subjective results with the distortion values. In [94], there are three parameters used for coding, number of layers, quantization parameter for left view and temporal scaling. In our codec, we are only using quantization parameter for adjusting the bit rates. Although both codecs are not the same, they are both extensions of H.264 JM and JSVM softwares. So the distortions become similar if we consider only the case where quantization parameter is used to adjust the bit rates. Also, subjective results for our codec with temporal and spatial scaling can be found in [14], where we have similar results given in [94].

4.3.5.1 Results on the Minimization of End-to-End Distortion

We solve the minimization in (4.21) by a general purpose minimization tool which uses sequential quadratic programming where the tool solves a quadratic programming at each iteration as described in [95]. In our work, we obtain the optimal encoder bit rates and parity ratios for $p_e \in \{0.03, 0.05, 0.1, 0.2\}$ and $R_C \in \{500, 750, 1000, 1500, 2000, 2500\ (\text{kbps})\}$ for 'Rena' video and $R_C \in \{1000, 1500, 2000, 2500, 3000, 3500\ (\text{kbps})\}$ for 'Soccer' video. Thus we perform 24 optimizations per video using (4.21).

In Tables 4.3 and 4.4, the optimal encoder bit rates and protection rates for the proposed method are given for the 'Rena' and 'Soccer' stereoscopic videos for $p_e = 0.10$. The encoder bit rates of the right view is lower than that of the left view, which is caused by the unequal weighting in the total distortion expression in (4.22). The protection rate of I-frame is the largest due to low bit rate and high distortion of losses.

In Tables 4.3 and 4.4, the protection rates of equal error protection (EEP) and Protect-L cases are also given. These protection rates are non-optimal and will be compared with the proposed optimal protection rates by simulations. In order to construct the EEP case, the resulting bit rate of proposed protection is distributed to the layers so that each layer has the same protection ratio. Protect-L case is constructed similarly, using the results of [96], where the bit rate of protection is distributed to only layers of left view (layer 1 and layer 0) so that these layers have same protection ratio. EEP and Protect-L cases are encoded using the same bitrate as the proposed algorithm.

4.3.5.2 Simulation Results

In this part, we evaluate the performance of the proposed stereoscopic video streaming system on lossy channels via simulations. We use two stereoscopic videos 'Rena' (Camera 38, 39) (640×480, first 30 frames) and 'Soccer' (720×480, first 30 frames) for performance evaluation. We encode the stereoscopic videos with the bit rates obtained by the minimization in (4.21) for given p_e and R_C, and NAL unit size is fixed to 150 bytes. The number of NAL units per layer can be calculated by dividing the given encoder bit rate to NAL unit size which yields the number of input symbols for the channel coder.

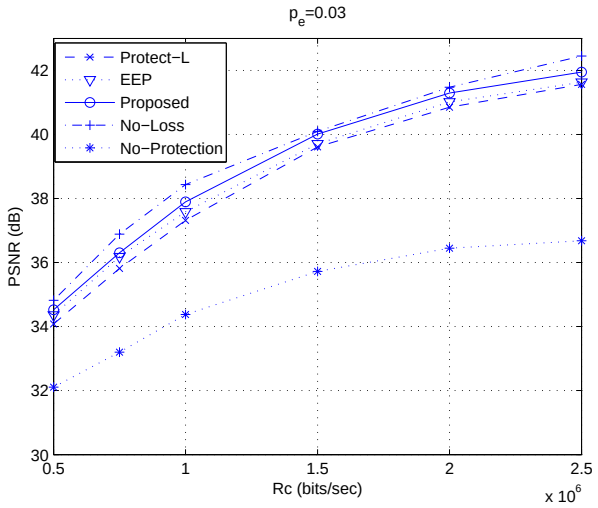


Figure 4.14: Results for $p_e = 0.03$ for 'Rena' video

For channel protection, we use systematic Raptor codes based on their suitability for our case as explained in [17]. We applied Raptor encoding to the source encoded video data using the

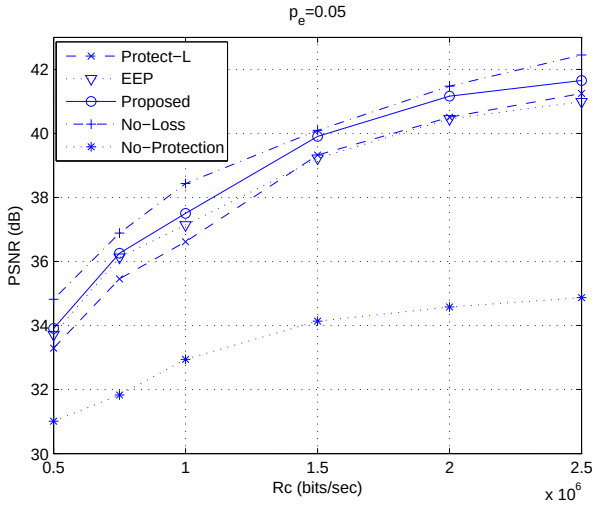


Figure 4.15: Results for $p_e = 0.05$ for 'Rena' video

protection rates obtained by the minimization in (4.21) for given p_e and R_C. The proposed optimal streaming scheme is compared with EEP, Protect-L, no-loss and no-protection cases. The no-loss case represents the quality of the video when the stereoscopic video is encoded with all available channel bandwidth and no transmission occurs. The no-protection case represents the transmission of the video of no-loss case without any channel protection and only error concealment is used at the decoder.

The simulation results give the average of 100 independent lossy transmission simulations for each p_e and R_C, where each packet is lost with a probability of p_e. Simulation results are based on the weighted PSNR measure. If we denote the average left and right per pixel distortions in MSE as D_{left} and D_{right} then the total PSNR distortion D(dB) can be calculated as

$$D\,(\text{dB}) = 10 \cdot \log_{10}\left(\frac{255^2}{\frac{2}{3}D_{left} + \frac{1}{3}D_{right}}\right) \tag{4.23}$$

We give the simulation results of stereoscopic video pair 'Rena' in Figs. 4.14 to 4.17 and those of 'Soccer' in Figs. 4.18 to 4.21. The gap between the results of the no-loss and the proposed case is caused by the reduction of the encoder bit rates of video where the remaining bit rate is used for channel protection. The simulation results demonstrate the superiority of the proposed scheme compared to non-optimized schemes. For low bit rates the difference is not clear but for high bit rates the difference is 1dB for $p_e = 0.10$ and nearly 2dB for

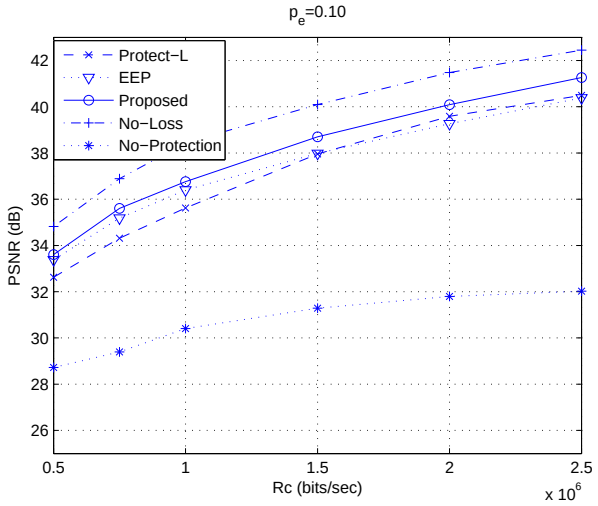


Figure 4.16: Results for $p_e = 0.10$ for 'Rena' video

$p_e = 0.20$. The results of the no-protection case clearly points out the need for FEC utilization in stereoscopic video streaming.

4.3.6 Conclusions

We presented a rate-distortion optimized error-resilient stereoscopic video streaming system with Raptor codes and evaluated its performance via simulations. We investigated all aspects of an end-to-end stereoscopic streaming system. Initially, we defined the layers of the stereoscopic video depending on the inter-view dependencies. Then, we obtained the analytical models for the RD curve of these layers where we extended the model of monoscopic video according to the dependencies. We showed that the analytical model of the RD curve accurately approximates the actual RD curve of the layers. Then, we obtained the analytical model of Raptor codes, which also accurately approximates the actual performance. Then, we estimated the transmission distortion for each layer where we also considered the propagation of NAL unit losses to following frames. Finally, we combined the two analytical models and the estimated transmission distortions in an end-to-end distortion minimization to obtain optimal encoder bit rates and UEP rates for the defined layers.

We evaluated the performance of the system via simulations where we used two stereoscopic videos 'Rena' and 'Soccer', which have different video characteristics. For both of the videos, the simulation results yielded the superiority of the proposed system compared to

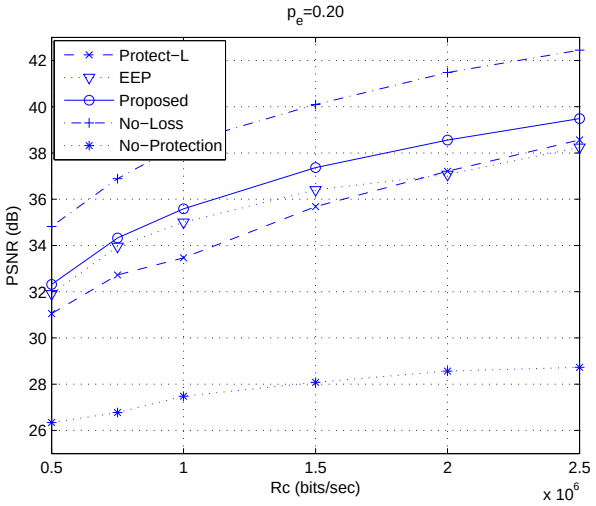


Figure 4.17: Results for $p_e = 0.20$ for 'Rena' video

non-optimized schemes. Also, the necessity of the utilization of FEC codes, such as Raptor codes, for stereoscopic video streaming on lossy transmission channels is clearly observed by examining the quality gap between the protected and non-protected streaming schemes.

The proposed system can be applied to any layered stereoscopic or multi-view streaming system for error resiliency.

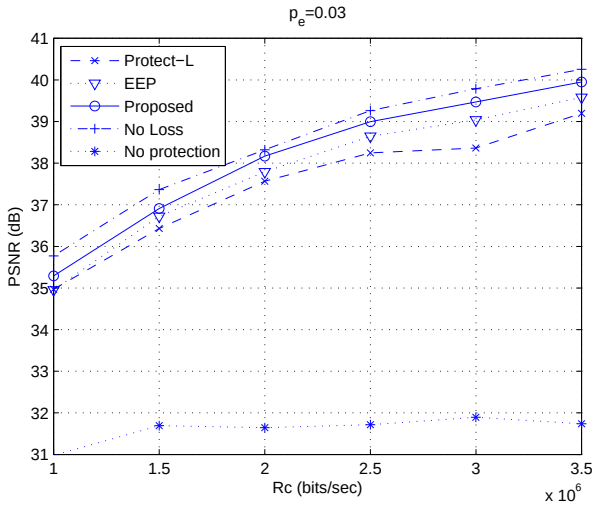


Figure 4.18: Results for $p_e = 0.03$ for 'Soccer' video

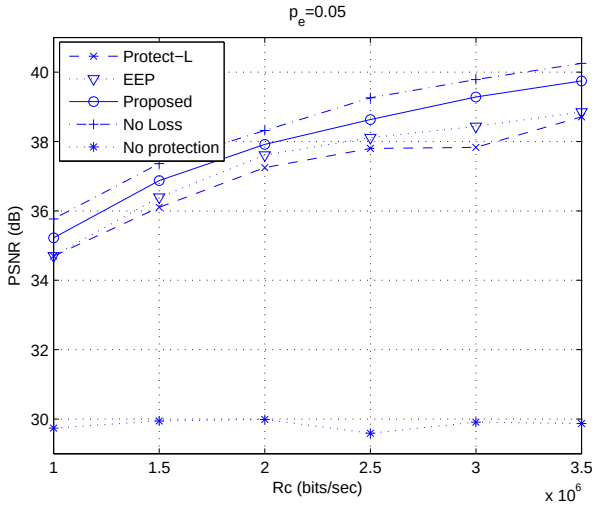


Figure 4.19: Results for $p_e = 0.05$ for 'Soccer' video

Table 4.3: Video Encoder Bit Rates and Raptor Encoder Protection Rates for ‘Rena’ Video

$p_e = 0.1$												
R_C(Kbps)	Encoder Bit Rates (Kbps) (Optimal)			Protection Rates								
				Proposed (Optimal)			EEP			Protect-L		
	R_I	R_L	R_R	ρ_I	ρ_L	ρ_R	ρ_I	ρ_L	ρ_R	ρ_I	ρ_L	ρ_R
500	33.5	216.6	169.8	0.489	0.177	0.147	0.190	0.190	0.190	0.320	0.320	0.000
750	51.5	337.8	250.7	0.389	0.158	0.143	0.172	0.172	0.172	0.282	0.282	0.000
1000	69.6	460.0	332.2	0.332	0.148	0.139	0.160	0.160	0.160	0.260	0.260	0.000
1500	106.0	705.6	496.0	0.270	0.138	0.133	0.147	0.147	0.147	0.237	0.237	0.000
2000	142.4	951.9	660.3	0.236	0.132	0.129	0.140	0.140	0.140	0.224	0.224	0.000
2500	178.9	1198.7	824.8	0.215	0.128	0.127	0.135	0.135	0.135	0.216	0.216	0.000

Table 4.4: Video Encoder Bit Rates and Raptor Encoder Protection Rates for 'Soccer' Video

$p_e = 0.1$												
R_C(Kbps)	Encoder Bit Rates (Kbps) (Optimal)			Protection Rates								
				Proposed (Optimal)			EEP			Protect-L		
	R_I	R_L	R_R	ρ_I	ρ_L	ρ_R	ρ_I	ρ_L	ρ_R	ρ_I	ρ_L	ρ_R
1000	68.4	543.0	245.9	0.349	0.147	0.156	0.166	0.166	0.166	0.233	0.233	0.000
1500	96.0	833.8	373.7	0.294	0.136	0.145	0.151	0.151	0.151	0.211	0.211	0.000
2000	123.7	1125.3	501.9	0.260	0.130	0.138	0.142	0.142	0.142	0.199	0.199	0.000
2500	151.3	1417.2	630.3	0.238	0.127	0.134	0.137	0.137	0.137	0.192	0.192	0.000
3000	179.0	1709.3	758.7	0.222	0.125	0.131	0.133	0.133	0.133	0.186	0.186	0.000
3500	206.6	2001.6	887.3	0.209	0.123	0.128	0.131	0.131	0.131	0.183	0.183	0.000

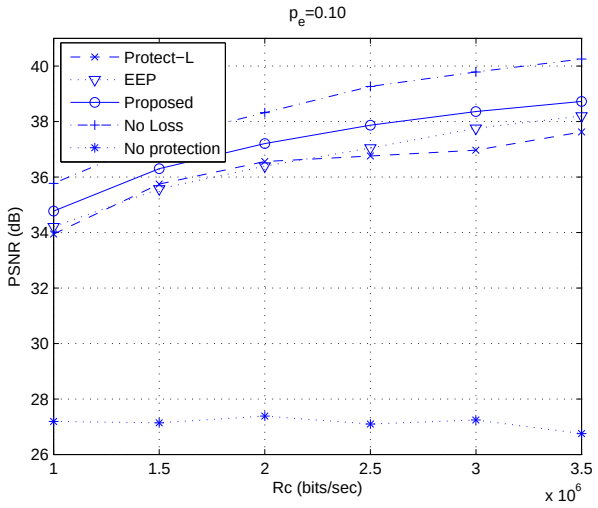


Figure 4.20: Results for $p_e = 0.10$ for 'Soccer' video

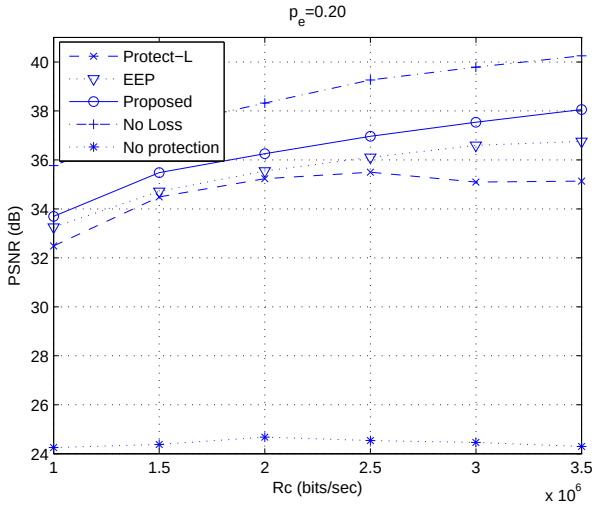


Figure 4.21: Results for $p_e = 0.20$ for 'Soccer' video

CHAPTER 5

Broadcast of 3D over DVB-H

Mobile TV has recently received a lot of attention worldwide with the advances in technologies such as Digital Multimedia Broadcasting (DMB), Digital Video Broadcasting - Handheld (DVB-H) and MediaFLO [97]. On the other hand, 3DTV is a new approach to watching TV, shifting it from being a passive experience to an interactive and more realistic one. With the merge of these two technologies it will be possible to have 3DTV products based on cell phone platforms with switchable 2D/3D autostereoscopic displays in the near future. Currently there are a number of projects conducting research on this issue such as the Korean 3D T-DMB [44], the European 3D Phone [98] and Mobile3DTV [1]. The latter one, specifically addresses the delivery of 3DTV to mobile users over DVB-H system.

There are several issues which have to be researched for 3D transmission over DVB-H such as the appropriate coding technique, error resilience, display factors etc. In this chapter, we try to answer questions related to developing 3D video specific error resilient techniques and experimenting with delivery under different channel conditions.

5.1 DVB-H

Basic elements of a DVB-H coder and transmitter are shown in Figure 5.1 [99][100].

DVB-H is the extension of DVB Project for the mobile reception of digital terrestrial TV. It is based on the existing DVB-T physical layer [101] with introduction of two new elements for mobility: MPE-FEC and time slicing [99]. Time slicing enables the transmission of data in bursts rather than a continuous transmission; explicitly signaling the arrival time of the next burst in it so that the receiver can turn off in between and wake up before the next burst arrives.

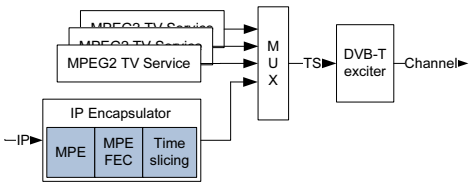


Figure 5.1: Basic elements of a DVB-H encapsulator and transmitter.

By this way the power consumption of the receiver is reduced.

Multi-Protocol Encapsulation is used for the carriage of IP datagram in MPEG2-TS. IP packets are encapsulated to MPE sections each of which consisting of a header, the IP datagram as a payload, and a 32-bit cyclic redundancy check (CRC) for the verification of payload integrity.

Forward Error Correction in link layer of DVB-H is implemented using Reed-Solomon (RS) codes calculated over the application data during MPE encapsulation. This procedure is illustrated in Figure 5.2. MPE Frame table is constructed by filling the table with IP datagram bytes column-wise. The DVB-H standard defines the table size; i.e. the number of rows that are allowed (256,512,768,1024) and the maximum number of Application Data (AD) columns (191) and RS columns (64) to be used. Therefore, if all the columns are used, an RS code of (255,191) is obtained. This case corresponds to a moderately strong code ratio of 3/4, where the code ratio is defined as the ratio of number of data symbols and total number of symbols including RS symbols. In order to achieve stronger or weaker code ratios, zero padding of AD columns and/or puncturing of RS columns can be employed. The RS codeword is calculated row-wise, which provides the interleaving of the data to some extent. In case the application data do not fill all 191 columns, zero padding is utilized. After the construction of the MPE-FEC Frame table, each IP packet is encapsulated into an MPE section as the payload column-wise. MPE sections, beside the payload data as the IP datagram, include also a header and a 32-bit cyclic redundancy check (CRC) for the verification of payload integrity. Encapsulation of more than one IP datagram into a single section is not allowed. RS data is not sent together with the application data, they are encapsulated into MPE-FEC sections having a similar format with the MPE sections. End of Application Data Table is signaled in the header of the last MPE section so that an MPE-FEC unaware user or a receiver who receives all the MPE sections correctly do not wait for MPE-FEC sections.

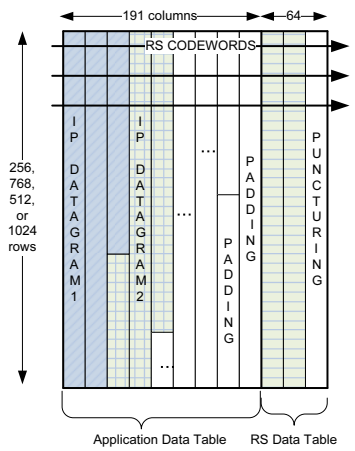


Figure 5.2: MPE-FEC frame structure.

After encapsulation of IP packets and embedding into MPEG-2 Transport Stream (TS) packets, the next block is the DVB-T modulator. In addition to the 2K and 8K modes of DVB-T, DVB-H also uses an intermediate 4K mode with a 4096-point Fast Fourier Transform (FFT) in the OFDM modulation. The objective of the 4K mode is to improve the network planning flexibility. To further improve robustness of the DVB-T 2K and 4K modes in a mobile environment and impulse noise reception conditions, an in-depth symbol interleaver is also standardized.

5.2 Error Resilient 3D Video Transmission over DVB-H

In this section, we present a complete framework of an end-to-end error resilient transmission of 3D video over Digital Video Broadcasting - Handheld (DVB-H) and provide an extensive analysis of coding and transmission parameters. We performed the analysis for different coding, error resilience and error corrections schemes using different contents coded at different bitrate levels. We integrated the slice interleaving error resilient coding tool into the reference software. Throughout the experiments, we investigate the effects of video content type, video bitrate, coding method, slice size and unequal protection level for different channel conditions.

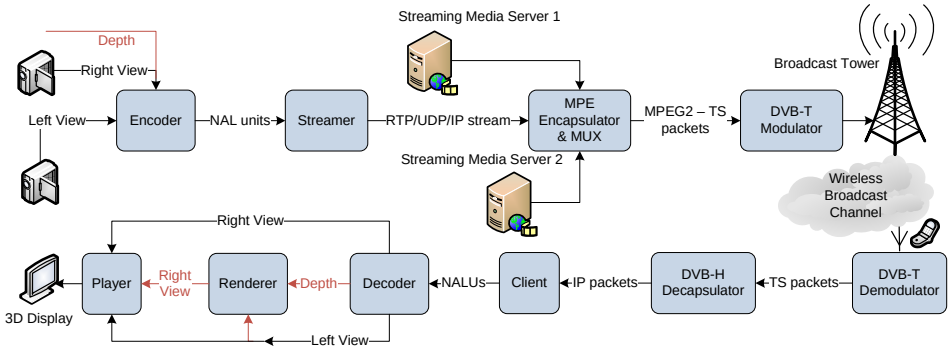


Figure 5.3: Block Diagram of the End-to-End Stereo Video Transmission over DVB-H Simulation System.

5.2.1 End-to-End Stereo Video Transmission System over DVB-H

The building blocks of the overall system used for the analysis of the error resilient tools are given in Figure 5.3 [102]. The transmission side consists of three main blocks, namely the encoder, streamer and DVB-H link layer encapsulator. Then, the system has an offline block to model the real physical transmission at the channel. The rest is the receiver side blocks which are the transport stream decapsulator, the decoder and a player. For proper display of 3D video, one would also need a multi-view video player and a 3D dedicated display.

In the encoder, coding method to be used is determined according to the representation type of input 3D video. Based on the input formats (MVV and VpD), three different coding methods are utilized at the encoder as:

- Simulcast Coding: This method is used with stereo video representation. Left and right views are compressed individually using the state-of-the-art monoscopic video compression standard H.264/AVC [20]. Since the inter-view dependency is not exploited, this method show worse compression performance than the following methods.

- MVC: This method is also used with stereo video as in simulcast. However, this time right view is encoded by exploiting the inter-view dependency using MVC [103] extension of H.264/AVC. IPP structure with simplified prediction scheme as shown in Figure 5.4 due to its simplicity and better error resilience performance resulting from reduced dependency between views. This scheme reduces the decoding complexity, but the compression efficiency slightly decreases as well.

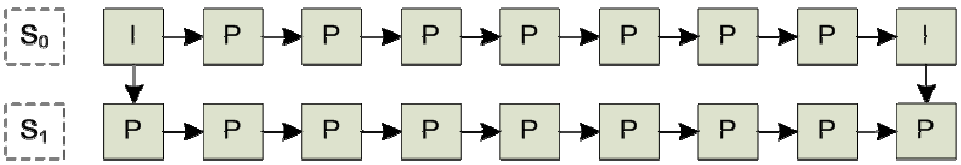


Figure 5.4: IPP Encoding Structure with Simplified Prediction Scheme of the MVC codec with inter-view references in the anchor frames.

- VpD Coding: This method is used with VpD representation. Both the left view and the depth data are compressed individually using again standard H.264/AVC [29].

Regardless of the coding method, the encoder output consists of two separate bit streams compressed and packetized appropriately for transmission. The encoder output is called Network Abstraction Layer (NAL) units (NALU). NALUs are fed to the stereo video streamer.

The streamer is basically responsible from the control of the transport. It encapsulates the NAL units into Real Time Transport Protocol (RTP), User Datagram Protocol (UDP) and finally Internet Protocol (IP) datagram and feed them to the DVB-H link layer encapsulator. RTP/UDP/IP packetization also allows the broadcasting of multiple contents from several resources each being a separate program.

DVB-H link layer encapsulator, which is the third block on the transmitter side of the system, firstly creates MPE sections and optional MPE-FEC section for forward error correction. We have modified the open source MPE encapsulator software, FATCAPS [104][100], in order to transmit two views simultaneously by multiplexing and to employ our error resilience schemes for robust transmission. The details of the encapsulator is given in Section 5.2.3.2.

The link layer output, MPEG-2 Transport Stream (TS) packets, are either passed to an real DVB-T modulator and transmitted or the physical layer can be simulated by modeling the DVB-T modulator/demodulator behavior and the channel conditions. In our work, we simulate the physical layer as described in [105] and the channel simulation parameters that we use are given in Section 5.2.3.3.

The receiver side of the end-to-end system starts with the DVB-H decapsulator block. This block is the major receiver block where the transition from received bitstream to compressed video stream is realized. The unpacking of TS streams, construction of MPE-FEC frame

table from sections, erasure decoding and possible recovery of lost data take place in this block. After these, the section payloads (IP datagram) are unpacked to output the NALUs.

The next block in the receiver side is the decoder. In this block, the left and right/depth NALU bitstreams are decoded and the video streams for each view are generated. For the case of VpD, the right view is rendered from the left view and the depth data. After decoding, the reconstructed left and right views are displayed on a proper 3D compatible display.

5.2.2 Error Resilience

In this section, we will describe the tools and changes incorporated into the system described in Section 5.2.1 for error resilience, namely slice interleaving and MPE-FEC protection including unequal protection strategies.

5.2.2.1 Slice Interleaving

The error resilient tools embedded in H.264/AVC standard are the data partitioning, slice interleaving, flexible macroblock (MB) ordering (FMO), SP/SI frames, reference frame selection, intra block refreshing and redundant slices [106]. Since SP/SI frames and reference frame selection requires feedback from the decoder, they are not suitable for our broadcasting based system. Data partitioning, slice interleaving, FMO, intra block refreshing and redundant slices are the candidates to be used in MVC. However, none of these tools are implemented in JMVC Reference Software for MVC extension of H.264/AVC.

We implemented the slice interleaving for error resilience and integrated it to JMVC 5.0.5 [107]. By this way, it is possible to code each different representation with different slice sizes. We varied slice size in order to see the effect of slice size on the performance of DVB-H transmission. We also modified decoder to support slice mode. Reference software decoder also does not support error handling. We integrated basic frame/slice copy concealment into the decoder to cope with losses. The details of the slice interleaving are given in the rest of the subsection.

H.264/AVC bit stream is composed of network abstraction layer (NAL) units as shown in Figure 5.5. In each NAL unit, there is a video coding layer (VCL) block. VCL can be a small

packet with information about the bitstream like sequence parameter set (SPS); picture parameter set (PPS) or supplemental enhancement information (SEI). SPS and PPS are required packets whereas SEI can be skipped. Other VCL packets are the coded video streams. Each packet is a slice containing an integer number of macroblocks. Slices are independently decodable if previous frames are available. This is achieved by using the location information in the slice header and by allowing spatial dependency only inside the slice. As a result, the resultant bitrate slightly increases by slice interleaving. In one extreme, a frame can be encoded using a single slice having maximum compression efficiency, but with worst performance in case of packet losses. The other extreme is to encode a single frame with lots of slices having best performance in case of packet losses but with worst compression efficiency.

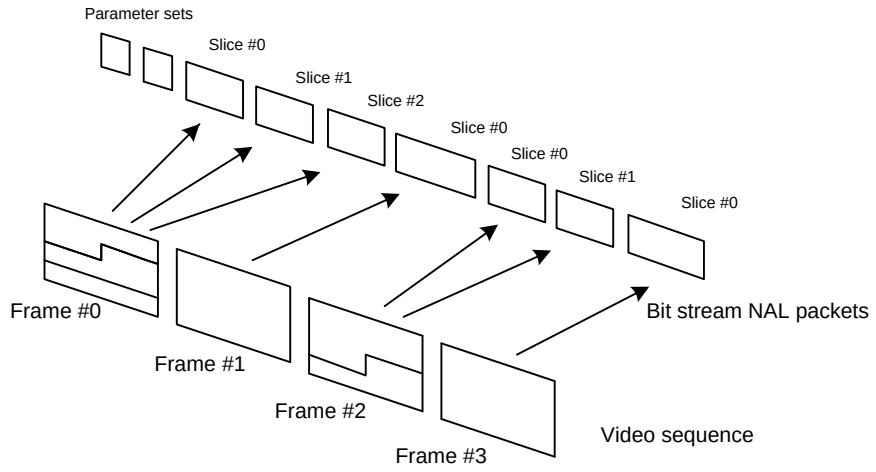


Figure 5.5: Bitstream syntax of H.264/AVC using fixed-size slices.

If NAL unit size is bigger than Maximum Transmission Unit (MTU) of the corresponding transport medium, it will be fragmented into smaller packets. In erroneous environments, some of these smaller packets can be lost, and this will cause the system to lose the entire frame, since parts of a NALU cannot be decoded by the decoder. However, by encoding a frame into several slices so that each slice size is smaller than MTU, each packet arrived at the decoder can be decoded correctly. The performance of slice encoding can be affected by the burst size of the error and also the size of the slices.

5.2.2.2 MPE-FEC Protection

In our simulations, we compare the performance of five different error protection modes which are Equal Error Protection (EEP) and Unequal Error Protection (4 different rates - UEP1, UEP2, UEP3, UEP4) modes. The definition of EEP is to protect the left and right (depth) views equally. This is implemented by adjusting the number of RS columns according to the number of application data columns and the intended FEC rate. In our experiments, we used typical 3/4 FEC rate for the left and right (depth) bursts in EEP mode. Then several unequal protection schemes are derived using this EEP structure. We realize four different UEP schemes by transferring (adding) a ratio of RS columns of the right (depth) view burst to the RS columns of the left view burst. The motivation behind unequal protection is that the independent left view is more important than the right or depth view. The right view requires the left view in the decoding process and the depth view requires the left view in order to render the right view. However, left view can be decoded without right or depth view. The ratios of RS columns to transfer are 1/4, 1/2, 3/4, and 1 for UEP1, UEP2, UEP3 and UEP4 respectively. Therefore, UEP1 is closest to the EEP among the UEP schemes and UEP4 is the other extreme where the right or depth view is not protected at all.

5.2.3 Simulation Environment

In this section, we describe the procedure used for the selection of key parameters used in the experiments and the reasoning behind it. In the following subsections 5.2.3.1, 5.2.3.2 and 5.2.3.3, we describe the procedures of encoded video selection, bandwidth allocation and channel simulation respectively. These three procedures define the simulation system that each experimental variable uses. Finally, we summarize the experimental variables that we vary and observe the resultant effect on the received quality in subsection 5.2.3.4.

5.2.3.1 Encoded Video Selection

Performance comparison of different codecs under lossy channels can be realized by assigning equal resources to each method at physical layer. We know that MVC achieves better compression for a chosen quality when compared to simulcast coding [27]. However, the question of what should be done with the remaining bit-budget does not have a straightfor-

ward answer. The available bandwidth can be used for extra protection or the video quality can be increased. During encoded video selection process, we take into account two cases, one of them being the ”equal quality” and other being the ”equal bitrate” cases. The idea is as follows: for ”equal quality” comparison, MVC, Simulcast and VpD encoded videos have the same resultant joint PSNR and for ”equal bitrate” comparison, they have the same total video bitrate. In our experiments, rate distortion curves of the encoded videos are obtained by varying the Quantization Parameter (QP) of each view. In simulcast coding, left and right views (left view and depth in case of VpD coding) are encoded independently and QP of each view can be varied independently. In case of MVC, due to inter-view dependency between left and right views, combinations of QP pairs are varied jointly. As a result of varying QP values, many rate-distortion pairs are obtained. When we try to achieve a given target bitrate for a coding scheme, we choose the QP pair which results in the target bitrate with greatest joint PSNR. Similarly, when we try to achieve a given target joint PSNR, we choose the QP pair at the target PSNR with smallest bitrate.

Before we start transmission simulation, we choose the QP pairs of the coding schemes according to the following procedure: We first assume that we are given a target total video bitrate. In our tests, we worked with two bitrates, one bitrate around 300Kbps and the other around 600 Kbps, but due to space constraints, we will only present the results with 300 Kbps. For the target video bitrate, we find the QP pairs of simulcast, MVC and VpD coding schemes and label them as Simulcast, MVC and VD. This part corresponds to ”equal bitrate” case. Then, we find QP pairs for MVC and VpD at the same PSNR with Simulcast whose QP pair was obtained in the previous part and label them as MVC2 and VD2. This part corresponds to ”equal quality” case. An example QP selection case is illustrated in Figure 5.6.

5.2.3.2 Bandwidth Allocation

Multiplexing of multiple services into a final transport stream in DVB-H can be realized either statically by assigning fixed durations for each service or dynamically by using a variable burst duration assignment algorithm. In this study, we used the fixed burst duration method as this is recommended by guideline [100] and commonly used in the current DVB-H systems. In order to reduce power consumption, each burst is transmitted for its burst duration and

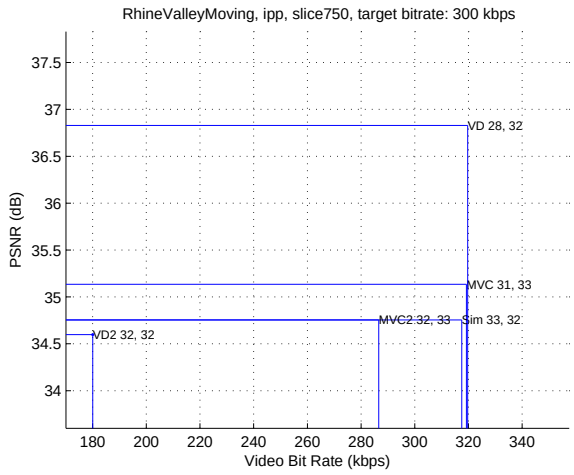


Figure 5.6: PSNR and bitrate values for selected QP pairs of the coding methods in RhineValleyMoving, Slice750, 300 Kbps tests.

then sleeps until the next burst for an off-time. The receiver knows when the next burst starts by the the delta-t parameter which signals the start of next burst. We assign two bursts/time slices for left and right/depth views with different program identifiers (PID) as if they are two separate streams to be broadcasted as shown in Figure 5.7. The reason behind assigning two separate bursts is to achieve backward compatibility such that a 3D non-compatible user can still receive only the left burst to play monoscopic video. This can be achieved by adjusting the delta-t's of both bursts to signal the start of next left view burst. Therefore a mono-capable receiver will be able to turn off after the end of the first burst discarding the right view. In order to minimize variance of number of bytes used for each burst, we insert an integer multiple of number of Group of Pictures (GOP) in a burst.

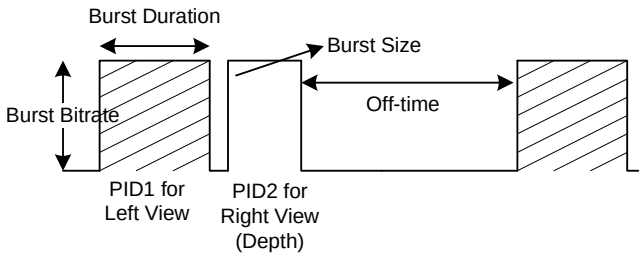


Figure 5.7: Burst Duration allocation for different views

Figure 5.8 illustrates example resultant maximum burst duration requirements for different transmission schemes. Although the videos were encoded at same target video bitrate, the resultant burst duration requirements slightly varies. In the experiments, since we choose the

same burst duration according to the scheme with maximum burst duration, some schemes cannot utilize all the available burst duration with initial number of RS columns. In order for each scheme to fully utilize available bitrate, we assign more RS columns that can be placed in the excess burst duration. In this way, if a scheme has a smaller burst duration requirement because of possibly better compression, it is automatically protected with more FEC, ensuring a fair comparison. Figure 5.9 illustrates an example of resultant FEC ratios for several schemes tested. Left1 and Right1 (Depth1) correspond to initial intended FEC ratios and Left2 and Depth2 correspond to resultant FEC ratios after more RS columns added due to unused available bitrate.

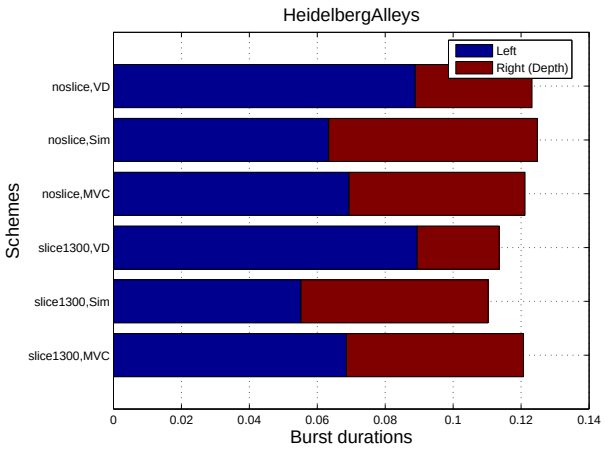


Figure 5.8: Burst Duration distributions for several schemes. (HeidelbergAlleys)

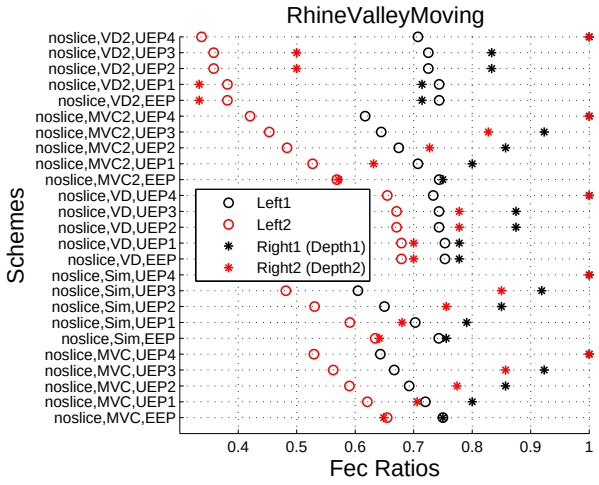


Figure 5.9: FEC Ratios vs Schemes (RhineValleyMoving)

5.2.3.3 Channel Simulation

In our experiments, we simulated the physical layer operations and transmission errors using the DVB-H physical layer modelling introduced in [105]. DVB-H has several physical layer parameters that affect the transmission bitrate and RF performance. We worked with a commonly used set of parameters: 16QAM as the modulation scheme, 2/3 convolutional code rate, 1/4 guard interval, 8K FFT mode and 666 MHz carrier frequency which results in a channel capacity of 13.2 Mbps. For the wireless channel simulation, we used the mobile channel model Typical Urban 6 taps (TU6) [108] with 38,9 km/h receiver velocity relative to source (which corresponds to a maximum Doppler frequency = 24 Hz).

For the comparison of methods being tested in this study, it is needed to identify different channel conditions accurately. One option is to identify channel conditions according to channel SNR values. This option has the disadvantage that, especially in lower channel SNR values, the experiments tend to have a large spectrum of loss conditions. This is due to the time varying behavior of TU6 channel. For this reason, we partition each channel SNR into loss rate intervals and analyze the performance according to the intervals as well. For the loss rate measurements, one can use the MPE - Frame Error Rate (MFER) defined by the DVB Community in order to represent the losses in DVB-H transmission system, which is given as:

$$MFER(\%) = \frac{Number\ of\ erroneous\ Frames\ After\ FEC x 100}{Total\ Number\ of\ Frames} \tag{5.1}$$

In this definition, an MPE-FEC frame is considered to be lost (erroneous) when the FEC decoding fails to correct even a single row of the MPE frame. This causes a large variation in the video frame losses corresponding to a single MFER value. Also the same error pattern results in a variety of MFER values for different videos making the metric extremely content and implementation dependent. Therefore, we prefer to use TS Packet Loss Rate (PLR) which is defined as:

$$TS\ \ PLR = \frac{Number\ of\ erroneous\ TS\ \ packets}{Number\ of\ total\ TS\ \ packets} \tag{5.2}$$

Figure 5.10 shows the TS PLR distribution for several channel SNR values and as mentioned

before, one can see that TS PLR has a high variance in especially lower SNR values.

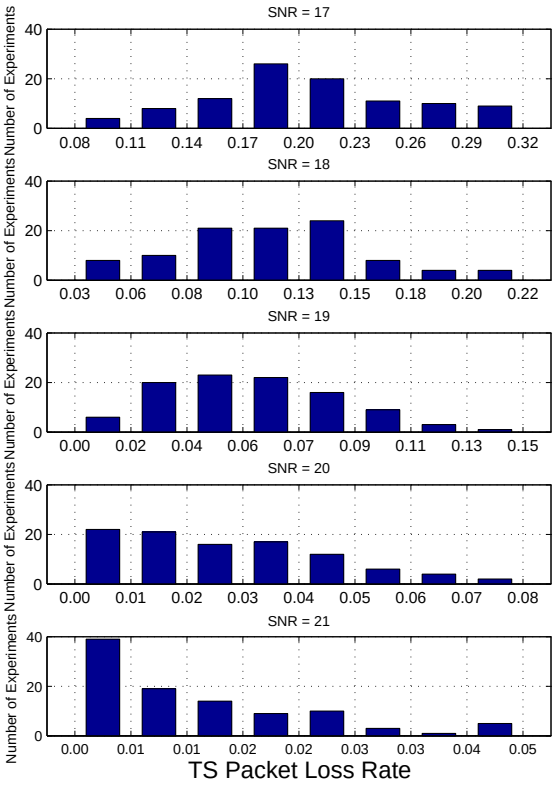


Figure 5.10: TS Packet Loss Rate at SNR values 17 to 21 dB

5.2.3.4 Experimental Variables

The parameters which determine the number of transmission tests are summarized in Table 5.1. We conducted experiments on four different video contents, whose characteristics are given in Table 5.2. Each video is encoded for target video bitrates of 300 kbps and 600 kbps. We varied the channel conditions for SNR values between 17 and 21 dB. For each channel SNR, we conduct 100 different experiments. In the experiments, we seek to find the most suitable coding method, slice mode and protection structure for the given DVB-H channel condition. There are five different coding methods where Sim, MVC and VD correspond to equal bitrate encodings whereas MVC2 and VD2 correspond to equal quality cases as described in Section 5.2.3.1. Slice sizes to be tested are 1300, 1000, 750, 500 Bytes and also using full frame slice mode. Finally, we test 5 protection structures, namely EEP and

UEP 1 to 4, as described in Section 5.2.2.2. Due to space limitations, we present the results for HeidelbergAlleys and RhineValleyMoving sequences encoded at 300 kbps simulated at channel SNRs between 17 and 19 dB. Note that, we observed very similar trend for the other contents and the bitrate. For the channel SNR values greater than 19 dB, the channel behaves close to lossless and same results with channel SNR of 19 dB is observed.

Table 5.1: Parameters of the Tests

Contents	4x	HeidelbergAlleys,KnightsQuest, RollerBlade, RhineValleyMoving
Coding Methods	5x	Simulcast, MVC (2 modes), VpD (2 modes)
Slice Modes	5x	Full-Frame and Fixed Slice Sizes of 1300, 1000, 750, 500 Bytes
Protection Structures	5x	EEP, UEP(4 different rates)
Video Bitrates	2x	300 Kbps, 600 Kbps
Channel SNR Range	5x	17-21 dB
Number of Experiments	100x	100 different error pattern for each transmission

Table 5.2: Spatial and temporal characteristics of the contents

Content	Characteristics	Width	Height	Fps
HeidelbergAlleys	Low Motion, High Detail	432	240	12.5
KnightsQuest	Computer - Generated,	432	240	12.5
RhineValleyMoving	High Camera and Object Motion, Low Detail	432	240	12.5
RollerBlade	High Object Motion	320	240	15

5.2.4 Results

In this section, we present the result of experimental results conducted according to the procedures described in Section 5.2.3. In the experiments, we evaluated 3D video quality by the objective joint PSNR metric which is calculated using the formulas below:

$$PSNR_j = 10 \cdot \log_{10}\left(\frac{255^2}{(MSE_l + MSE_r)/2}\right) \tag{5.3}$$

In this expression, MSE_l and MSE_r correspond to mean square error between original and distorted left and right sequences respectively. In case of VpD sequences, since even for the lossless case there is an existing distortion (for PSNR metric) due to imperfections during depth estimation and rendering, original right view is not taken as the reference sequence. During the calculation of MSE_r, instead of the original right view, the right view rendered from original left and original depth is used.

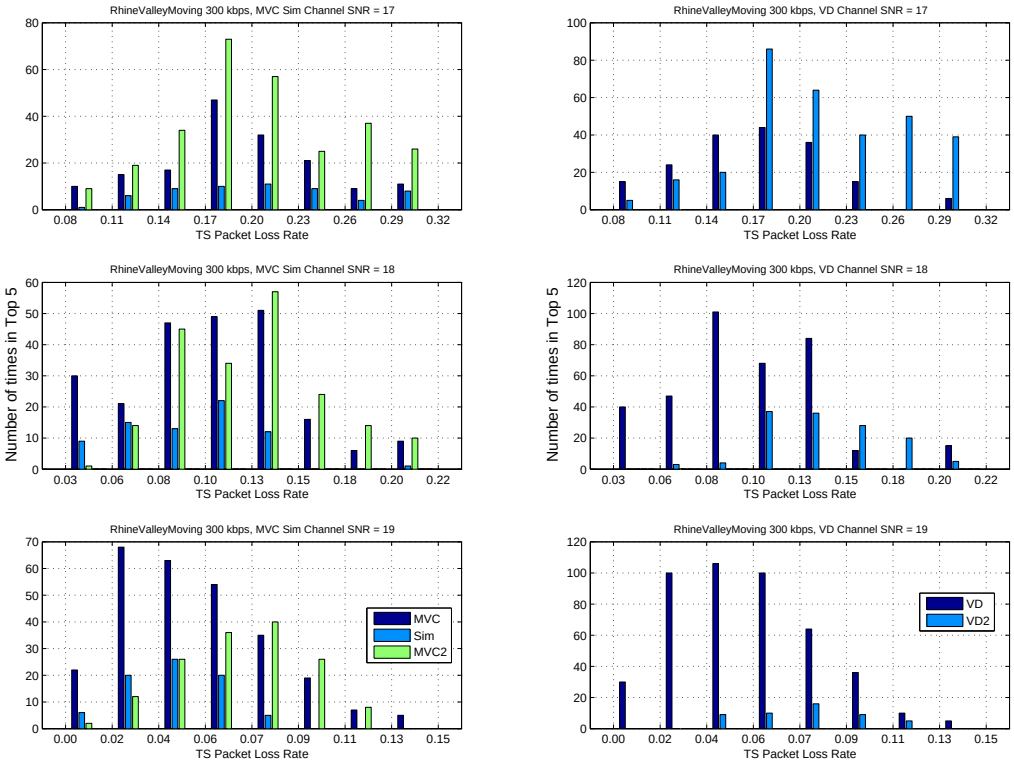
In the following three subsections results of the transmission tests are presented. They are coding method comparison, slice mode comparison and protection method comparison. During comparisons, we employ PSNR values of 100 experiments carried out for each channel SNR. A common way of evaluating error resilience techniques is to compare the average distortion results of each method for each value of the channel SNR range that is being worked. Although looking at the average resultant PSNR values according to each channel SNR provide the overall tendencies of the methods tested; since a channel SNR value may contain many TS PLRs within as shown in Figure 5.10. This comparison method may not reveal all the information that can be extracted from the tests. Therefore we try to evaluate the transmission schemes in a more accurate way by also presenting the PSNR values of each TS PLR existing in a channel SNR. Hence each channel SNR has its own figure where distortion PSNRs are provided for the TS PLRs. Since all the schemes are granted same amount of burst durations and the error traces in the experiments affect the same locations in the transport stream, every scheme is assumed to encounter same TS PLR for an experiment.

For the evaluation of the results, we provide the PSNR vs. TS PLR figures of not all 100 experiments but a subsample of them and defined the winner top 5 experiments. Winner top 5 means the experiments with the highest 5 PSNR values for the specific TS PLR. In the winning scheme based evaluations, we divide the TS PLRs range that are encountered in the experiments for an SNR into sub regions. In each sub region, we consider the experiments that resulted in a TS PLR value falling onto the corresponding sub region. We calculate the PSNR values of all schemes for all the experiments and sort the PSNR values in descending order in each subregion. Then, for each experiment in the sub regions, we consider the first five (Top 5) values. Finally, we present the number of occurrences of competing schemes in Top 5. The motivation behind dividing TS PLR values into sub regions is to make more reliable and meaningful comparisons within similar TS PLR values.

Due to differences in PSNR calculation of VpD, there may exist a significant difference between PSNR values of VD and VD2 compared to MVC, MVC2 and Sim. Therefore, comparing all methods together may lead to a misjudgement of the winning based evaluations. In order to compare VD approaches with MVC and Simulcast reliably, it is necessary to measure the quality of coding methods subjectively so that the resultant qualities can be reliably compared. Hence, we provide the comparison of MVC, MVC2 and Sim methods and VD, VD2 methods separately.

5.2.4.1 Comparison of Coding Methods

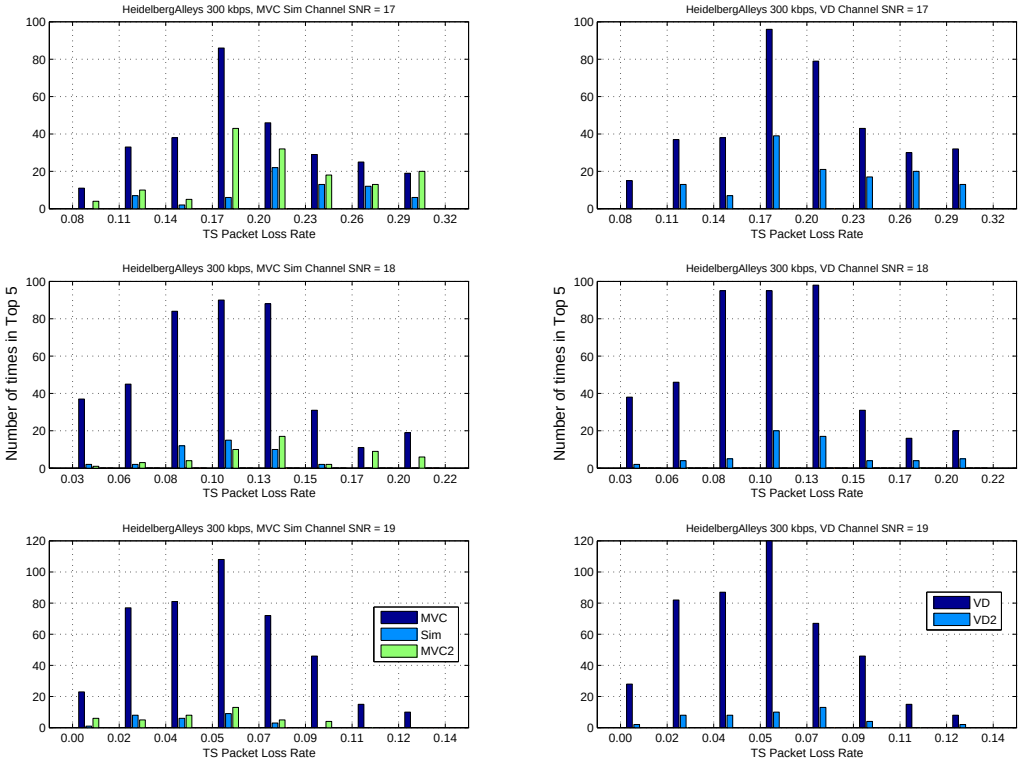
In this subsection we compare the performance of the coding methods defined in Table 5.1. Top 5 Count and PSNR graphs of MVC (MVC2,Sim) and VD (VD2) are provided. The counts of the experiment methods resulted in the highest PSNR for the TS PLRs within a channel SNR is given as a subplot. Three different channel SNRs are given on top of each other. Coding method comparison figures are provided in the range Figure 5.11 - 5.13.



(a) (b)

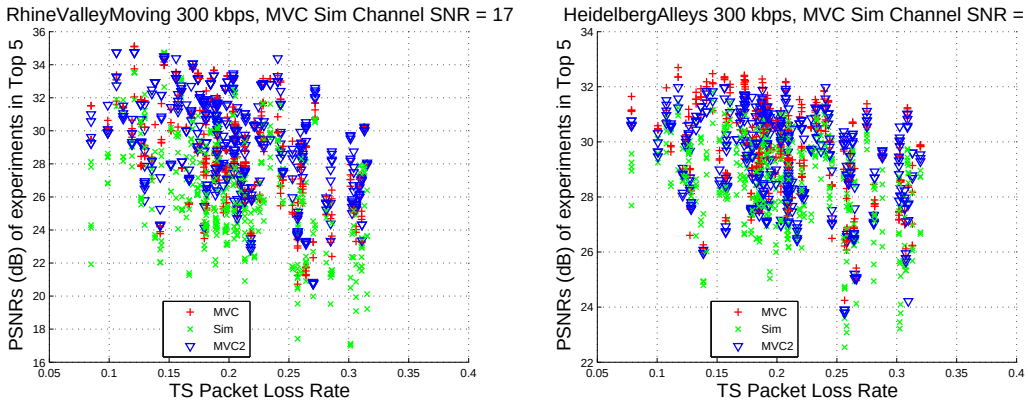
Figure 5.11: (a) TS PLR vs Top 5 Count plot of Coding Method comparison for MVC, Sim, MVC2 (RhineValleyMoving) (b) TS PLR vs Top 5 Count plot of Coding Method comparison for VD, VD2 (RhineValleyMoving)

From the figures, it can be observed that for low TS PLRs and high channel SNR, MVC has the best performance. This is an expected result since the lossless quality of MVC is higher than both MVC2 and Simulcast since it compresses better than Simulcast and has higher bitrate than MVC2. For low channel SNR values (17 dB in our case), as the TS PLR increases, MVC2 becomes the leading coding method due to its high protection rate. It is clear



(a) (b)

Figure 5.12: (a) TS PLR vs Top 5 Count plot of Coding Method comparison for MVC, Sim, MVC2 (HeidelbergAlleys) (b) TS PLR vs Top 5 Count plot of Coding Method comparison for VD, VD2 (HeidelbergAlleys)



(a) (b)

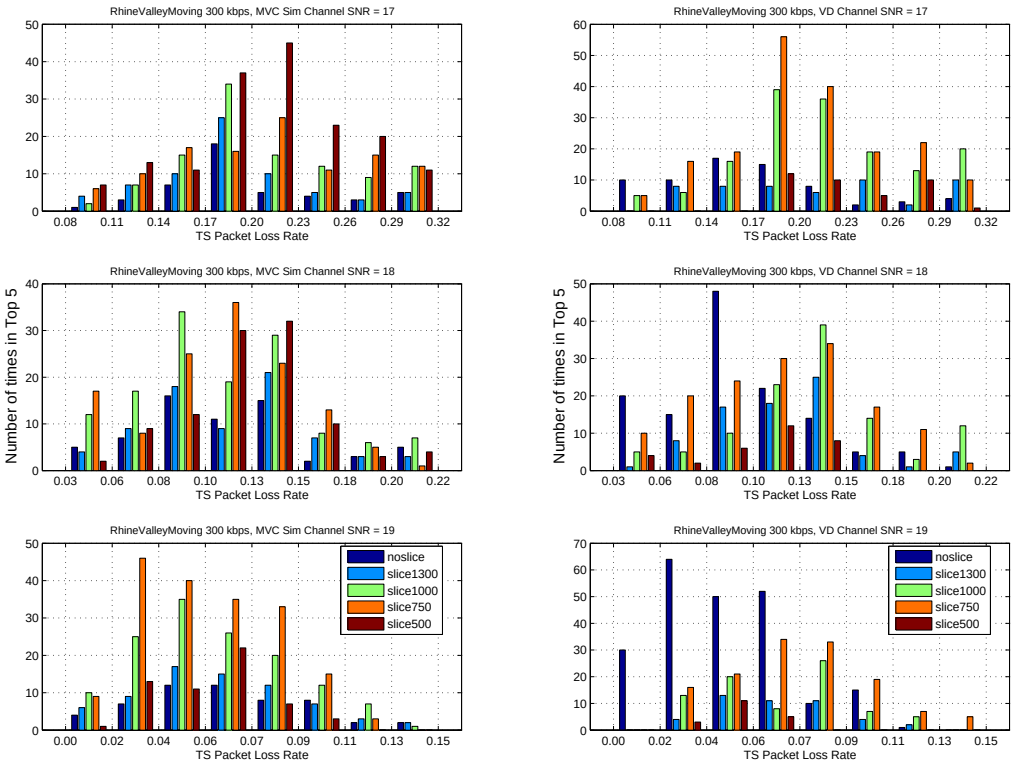
Figure 5.13: (a) TS PLR vs PSNR plot for Top 5 of Coding Method comparison for MVC, Sim, MVC2 (RhineValleyMoving) (b) TS PLR vs PSNR plot for Top 5 of Coding Method comparison for MVC, Sim, MVC2 (HeidelbergAlleys)

that Simulcast would not be the preferred coding method for any of the conditions tested. On the other hand, for SNRs higher than 18 dB MVC is clearly the winner. For channel SNR 17 dBs; since a clear winner for all TS PLR does not exist in Top 5 Count graphs, it may be helpful to check the PSNR vs TS PLR graphs for these experiments since it provides the PSNRs of the methods that follows the winner for a specific TS PLR. In most of the cases, when we check low TS PLRs, where MVC is better than MVC2, the PSNR difference of the two methods come from the original coding, since in low error rates, generally both methods can recover by the FEC. On the other hand, for high TS PLRs where MVC2 performs better, the PSNR difference between MVC and MVC2 varies according to the characteristics of the error pattern. The relationship between MVC and MVC2 is also seen between VD and VD2. Moreover, while the average PSNR value of VD decreases as loss rate increases the PSNR value of VD2 remains same.

5.2.4.2 Comparison of Slice Modes

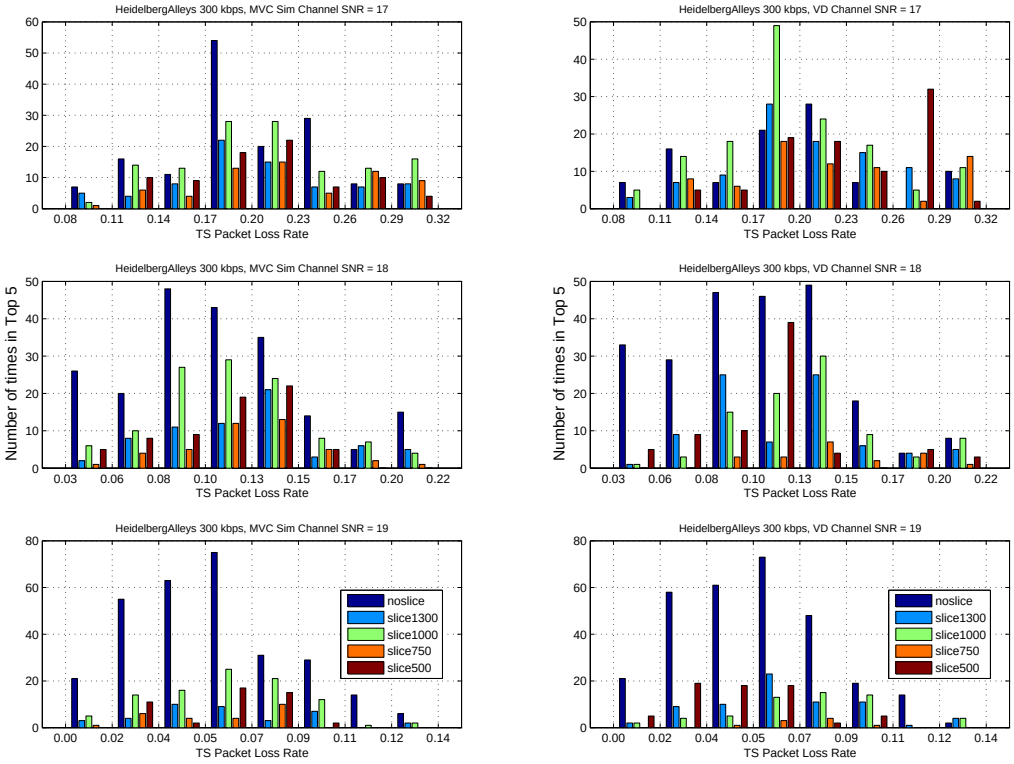
In this subsection, we compare the performance of the five different slice modes. This time, the coding (MVC, MVC2 and Sim in one figure, VD and VD2) and protection methods encoded with the same slice size are pooled into the same group. Therefore, we have 5 different cases to compare regardless of the other parameters. The results of the experiments are arranged in this way to compare the performance of slice modes and provided in Figure 5.14 - 5.16.

Looking at the figures, for MVC comparisons, it is clearly observed that slice modes have a superior performance over the full frame slice mode when the TS PLRs are high, especially at levels observed in channel SNR value equal to 17 dB. It is expected that full frame slice will be the winner for low loss rates since it has the best rate-distortion performance as it does not have any slice overhead. In the experiments, most of the time, we observe that full frame slice turns out to be the winner for low loss rates while moving up to high channel SNRs. The reason is that at these levels of loss, error decoding can correct the errors, (recover the losses) regardless of the protection scheme used. Nevertheless, there also exist figures where we see that the winner at high channel SNRs is one of the slice modes. For these cases, if we look at the lossless PSNRs, we see that the slice mode is encoded with a slightly better PSNR, which may be a result of selection of the QPs for encoding. Since the bitrate vs PSNR curve for



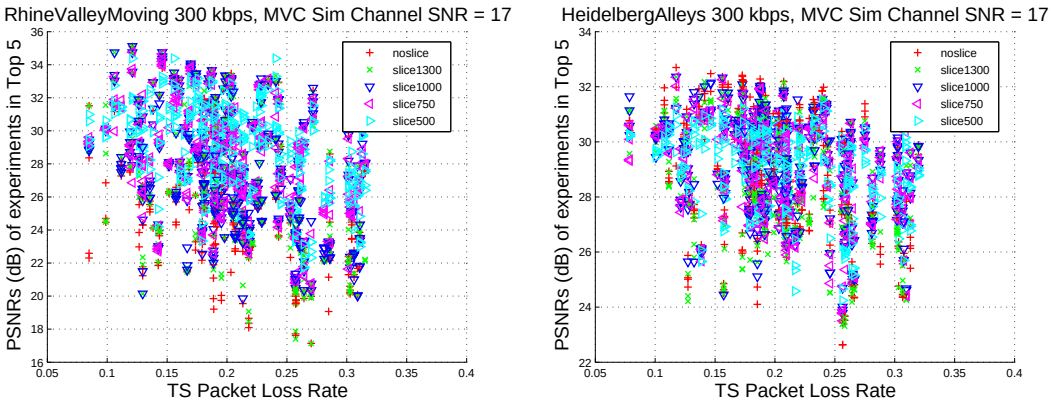
(a) (b)

Figure 5.14: (a) TS PLR vs Top 5 Count plot of Slice Mode comparison for MVC, Sim, MVC2 (RhineValleyMoving) (b) TS PLR vs Top 5 Count plot of Slice Mode comparison for VD, VD2 (RhineValleyMoving)



(a) (b)

Figure 5.15: (a) TS PLR vs Top 5 Count plot of Slice Mode comparison for MVC, Sim, MVC2 (HeidelbergAlleys) (b) TS PLR vs Top 5 Count plot of Slice Mode comparison for VD, VD2 (HeidelbergAlleys)



(a) (b)

Figure 5.16: (a) TS PLR vs PSNR plot for Top 5 of Slice Mode comparison for MVC, Sim, MVC2 (RhineValleyMoving) (b) TS PLR vs PSNR plot for Top 5 of Slice Mode comparison for MVC, Sim, MVC2 (HeidelbergAlleys)

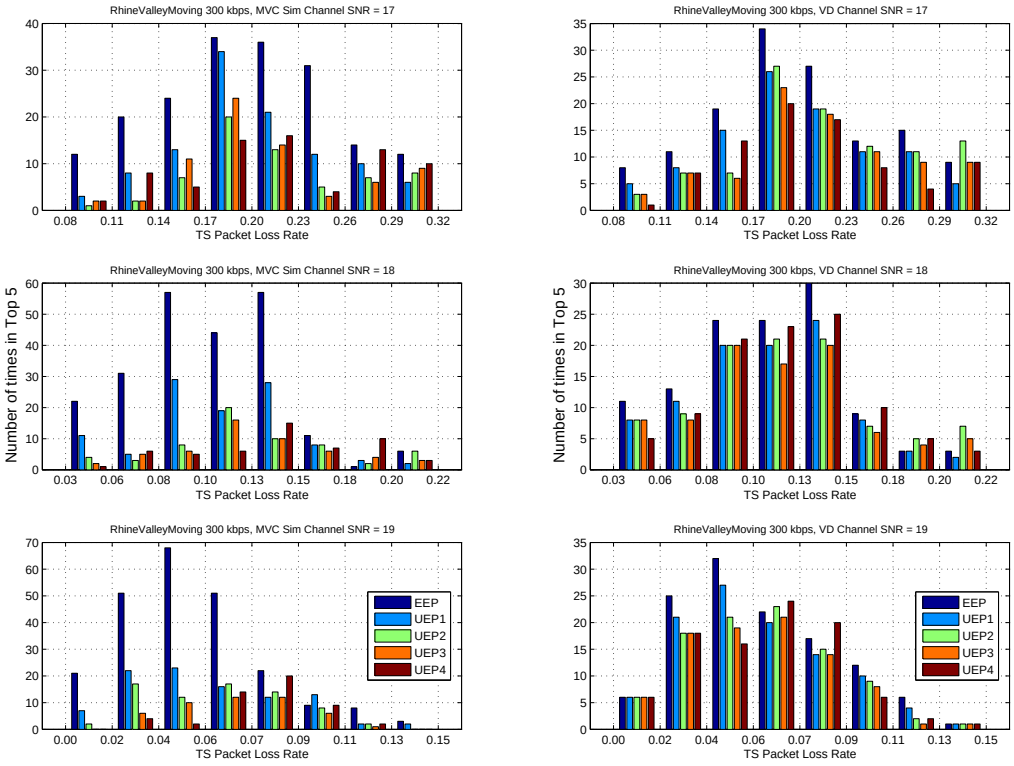
a content encoded with some coding method is not continuous and different coding methods have their own curves, choosing a target bitrate requires also defining a margin in order to come up a with a solution. Generally, this margin is small enough such that the resultant bitrate difference does not correspond to a significant change in PSNR. One final conclusion that can be derived is that even though we say that the full frame slice is the winner for low loss conditions, the maximum difference between the winner full frame slice video PSNR and following slice mode video PSNR occurs to be at most 0.5 dB. However, when there is loss such that the slice modes performance is dominant, the difference in PSNRs is generally much more significant. Therefore, using slice mode clearly provides more robustness. However, the slice size to be used should be selected according to the characteristics of the content since the results show that it is highly content dependent.

If we compare the sequences in slice modes, we see that for RhineValleyMoving sequence, slice sizes of 500 and 750 are mostly better than other sizes. However for HeidelbergAlley sequence, full frame slice perform better. This is mostly due to the low motion characteristics of the sequence. In low motion sequences, losing a whole frame does not cause higher distortions since error concealment from the previous frame can conceal the frame efficiently. However in high motion sequences, correlation between previous and current frame is less and concealment generates higher distortions. VD comparisons also show similar trends as in MVC comparisons. Most of the time, slice mode performs better than full frame slice mode.

5.2.4.3 Comparison of Protection Methods

The videos are grouped according to their protection methods regardless of the coding methods (MVC, MVC2 and Sim in one figure, VD and VD2) and slice sizes. The results of the experiments are arranged in this way to compare the performance of protection methods and they are provided in Figure 5.17 to 5.19. Looking at the figures, we see that EEP usually performs better than the others and especially when channel SNR value is greater than 17 dB, the performance difference increases. However, we note that EEP does not perform clearly superior performance than UEP as in the case of MVC vs Simulcast comparison. There are some exceptional cases where for some specific TS PLR, one of the UEP schemes (usually UEP1) becomes the winner. These cases mostly occur for the comparisons between VD and VD2 at channel SNR 17 dB and high loss rate. However, this does not affect the overall pic-

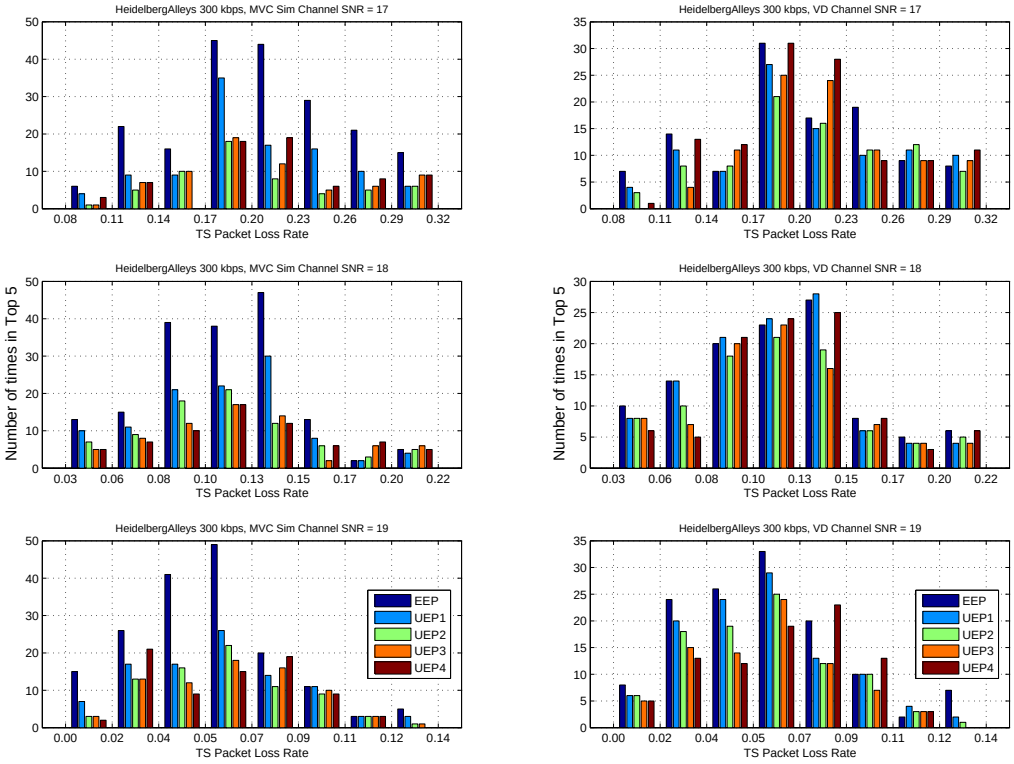
ture. Even when UEP becomes the winner, it is the one that has FEC rates closer to EEP than others. It can be observed from the figures that UEP1 is having PSNR values close to EEP where EEP is the winner.



(a) (b)

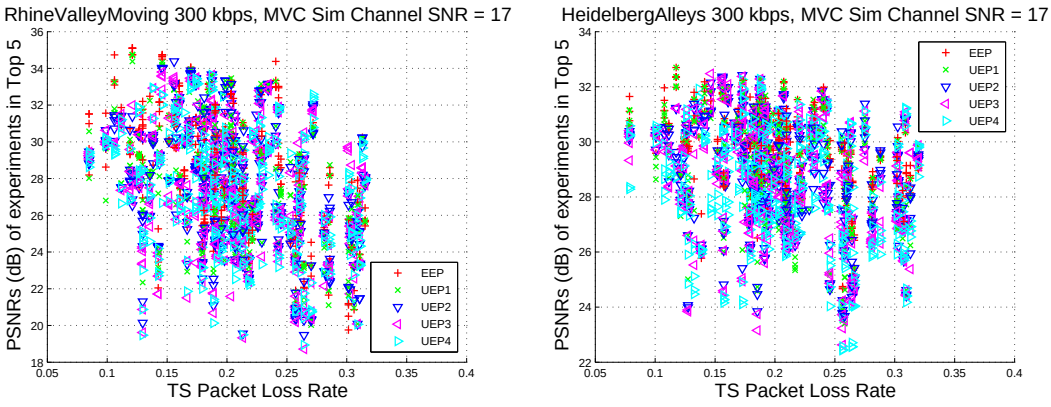
Figure 5.17: (a) TS PLR vs Top 5 Count plot of Protection Method comparison for MVC, Sim, MVC2 (RhineValleyMoving) (b) TS PLR vs Top 5 Count plot of Protection Method comparison for VD, VD2 (RhineValleyMoving)

Here it is important to note that these results are for an UEP method implemented by protecting left views more. The reference structure used in these experiments may not be suitable to achieve better robustness using UEP. In our structure, only the anchor frame (I frame) of the right view depends on the I frames of the left view. Therefore, whenever there is a loss in P frames of left view, this error does not propagate to the right view. However, in our UEP schemes, we protect the P frames of the left view more than the frames of the right view.



(a) (b)

Figure 5.18: (a) TS PLR vs Top 5 Count plot of Protection Method comparison for MVC, Sim, MVC2 (HeidelbergAlleys) (b) TS PLR vs Top 5 Count plot of Protection Method comparison for VD, VD2 (HeidelbergAlleys)



(a) (b)

Figure 5.19: (a) TS PLR vs PSNR plot for Top 5 of Protection Method comparison for MVC, Sim, MVC2 (RhineValleyMoving) (b) TS PLR vs PSNR plot for Top 5 of Protection Method comparison for MVC, Sim, MVC2 (HeidelbergAlleys)

5.2.4.4 Overall Comparison

Finally in this section, we depict the results of all aforementioned schemes as experiment PSNRs vs TS PLRs at channel SNR 17 dB. Figure 5.20 and Figure 5.21 show the results for RhineValleyMoving and HeidelbergAlley sequences respectively. These results are demonstrated in order to observe the experiments of all of the schemes. Both figures verify the comparison results we have made in the previous subsections.

5.2.5 Conclusion

We have provided an extensive analysis of coding and transmission parameters for robust end-to-end transport of 3D video over DVB-H. We perform the analysis for different coding, error resilience and error corrections schemes using four contents and two bitrate levels. Simulation results are evaluated for the combination of each coding method (MVC, Simulcast, VpD), error protection mode (EEP, UEPs with different protection rates) and error resilience structure (full frame slice, Slice mode with slice sizes 500, 750, 1000, 1300 bytes) and an overall comparison is also given after these evaluations.

First of all, for the slice modes comparison, we conclude that slice mode outperforms full frame slice mode in most of the cases. Full frame slice method outperforms the slice methods with a slight PSNR improvement at low loss cases. On the other hand, when a slice mode outperforms the full frame slice mode, the resulting PSNR difference is much more significant. In summary, we conclude that it is more appropriate to use slice mode.

Secondly, the analysis of different coding methods shows that MVC outperforms Simulcast most of the cases. Another important observation about MVC modes is the importance of allocation of FEC rate and payload for a given bitrate. When the channel SNR is greater than 17 dB, MVC (coded using better quantization parameter) is better than MVC2 (formed using more FEC). However, MVC2 has higher PSNR values when the packet loss rate increases. When we examine the performance of VD modes, we observe that using the previously described rendered right based PSNR calculation may result in much higher or lower PSNR values compared to simulcast and MVC. Therefore, in our results, we preferred to present the results separately. Nevertheless, the comparison results of VD and VD2 shows similarity to the one between MVC and MVC2. In order to compare VD approaches with MVC and

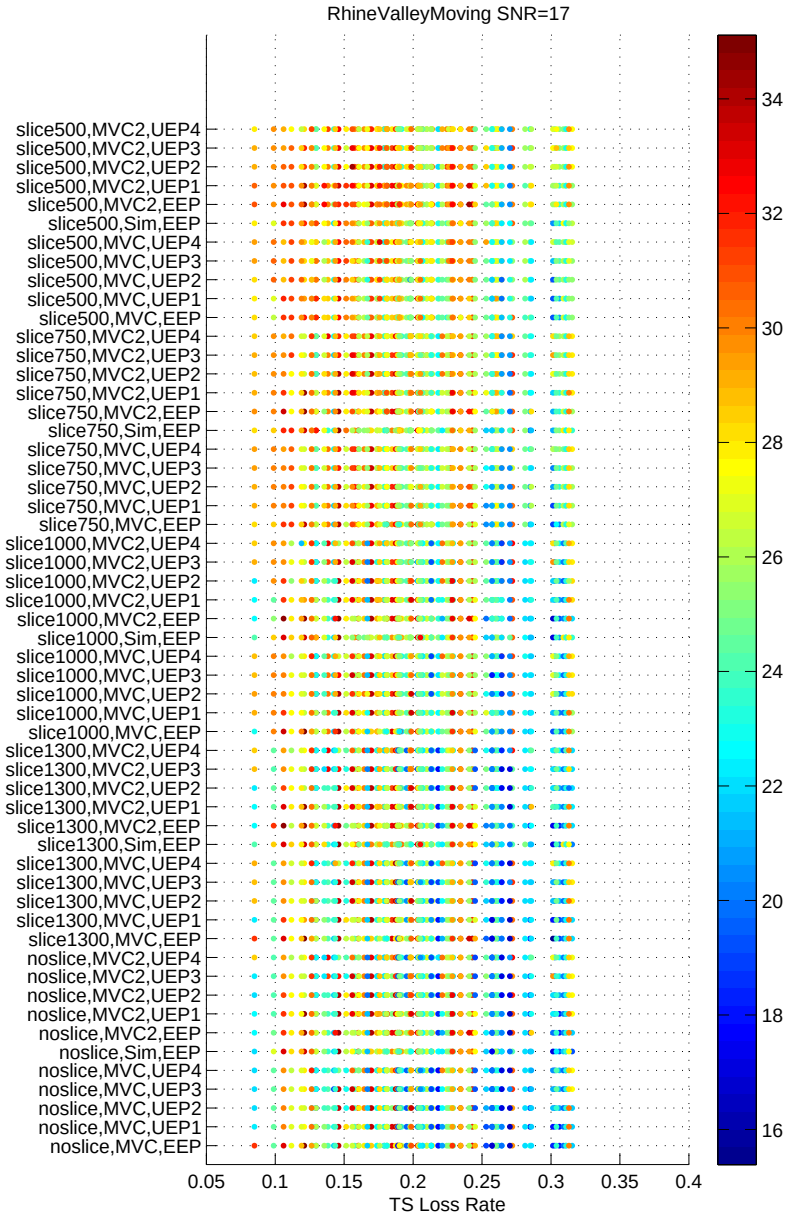


Figure 5.20: Comparison of All Coding, Protection Methods and Slice Modes at channel SNR 17dB (RhineValleyMoving)

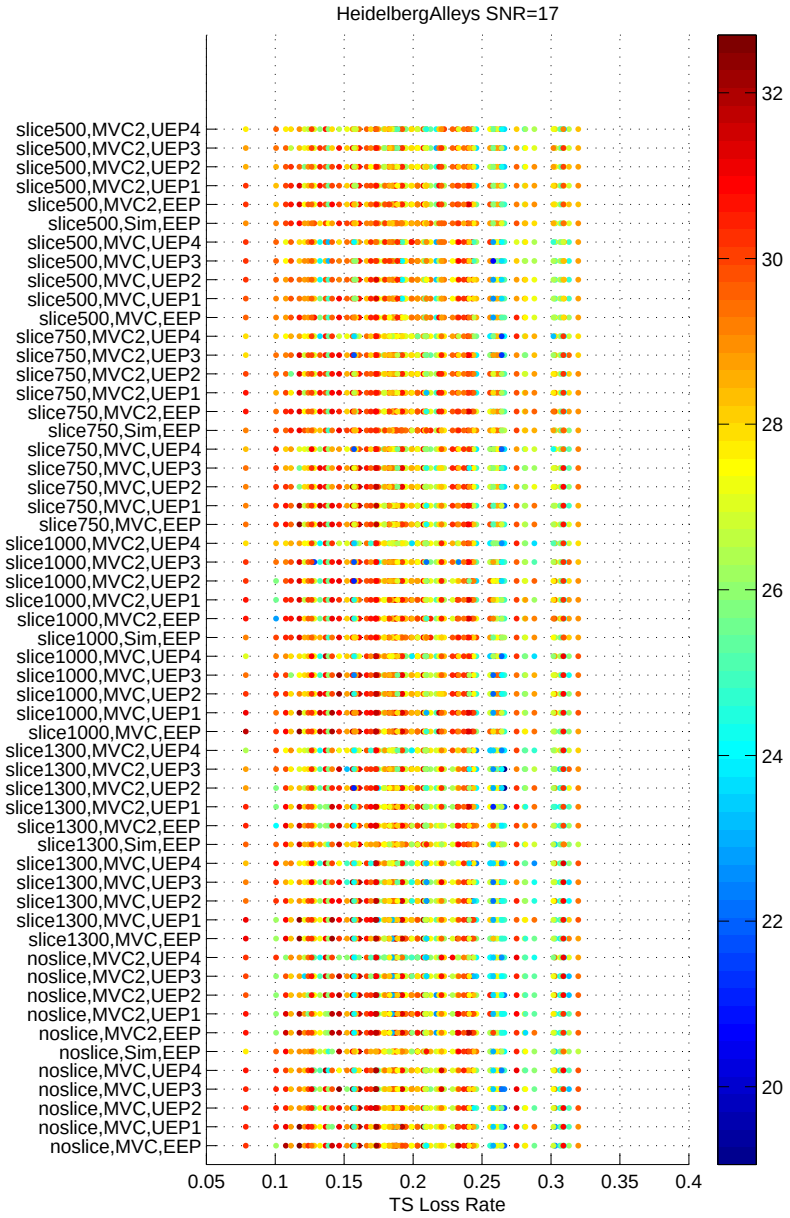


Figure 5.21: Comparison of All Coding, Protection Methods and Slice Modes at channel SNR 17 dB (HeidelbergAlleys)

Simulcast reliably, it is necessary to measure the quality of coding methods subjectively so that the resultant qualities can be reliably compared.

Thirdly, analyzing different error protection methods shows that EEP and UEP1 have better quality than other FEC modes. Indeed, EEP outperforms others when the SNR is greater than 17 dB and it shows similar characteristic to the UEP1 when the packet loss rate is high. However, we do not conclude that EEP is always better UEP in general because the results we presented only correspond to the coding and UEP structure we employed. UEP schemes which suit the coding structure better may perform better.

5.3 Optimization of Encoding and Error Protection Parameters for 3D Video Broadcast over DVB-H

In the previous section, EEP and 4 UEP schemes are compared. In this section, we try to optimize the system using modeling of the end-to-end system. First we suggest a heuristic methodology for modeling the end-to-end rate-distortion (RD) characteristic of such a system. Second, we use such a model to optimally select the parameters of the video encoder and the error correction scheme to minimize the overall distortion.

In Section 4.3, we defined 3 layers being I frames of So, P frames of So and P frames of S1. First layer (I frames of So) are approximately %7 of the total bitrate. We have tried in our simulation system for videos with 3 & 4 layers and see that performance is about 2 dB less in case of EEP and by applying UEP with more layers, we can only reach to the point of EEP performance of 2 layer video. Also by sending 3D video in 3 layers, will break backward compatibility. In this work, we will be using 2 layers, namely left view and right view. As shown in Figure 5.4, layer 0 (left view) is independent and layer 1 (right view) is dependent on layer 0.

The video encoder can encode each view with different quantization parameters, thus with different bit rates R_L and R_R . Due to lossy compression, the encoding process causes a distortion of D_e in the video quality. The available bandwidth for the 3D video is R_C . We apply different protection rates ρ_L and ρ_R to each view, because they contribute differently to the video quality. After the reception of DVB-H signal, some of the packets are erroneous and MPE-FEC decoder operates to recover from the errors. However, some packets still may

not be recovered and the loss of these packets causes a distortion of D_{loss} in the video quality. In this system, our goal is to obtain the optimal values of encoder bit rates R_L and R_R and protection rates ρ_L and ρ_R by minimizing the total distortion $D_{tot} \triangleq (D_e + D_{loss})$. In order to execute the minimization, we obtain the analytical models of each part of our system. We start with the modeling of the RD curve of each view of the 3D video encoder. Then, we define the analytical model of the performance of MPE-FEC. Finally, we estimate the distortion on the 3D video quality caused by packet losses.

5.4 Encoder Distortion Modelling

In this section, we model either rate-distortion (R-D) or rate-quality (R-Q) performance of MVC encoder. The RD curve of video is widely used for optimal streaming purposes [74, 75, 76, 77], which provides the optimal streaming bit rate for a given distortion in video quality and vice versa. [92] proposes R-D model for monoscopic video and in [17] the model is extended for stereoscopic video using 3 layers. [109] models R-Q for monoscopic video and also models relation between R & Q with the QP parameter of the codec. In the following subsections, we show how we modified the proposed models for stereoscopic video and in the last subsection the performance of both models are compared.

5.4.1 R-D Model

In this model, mean square error (MSE) at the video encoder is given in terms of encoding rate as shown in Equation 5.4 as described in [92]. We are using the same model for the left view, since it is encoded independently and is basically same as encoding monoscopic video with H.264/AVC. In our system right view is predicted from left view, therefore, R-D performance of right view is dependent on the left view. We modified the model for right view in [17], as shown in Equation 5.5.

$$D_{e,L}(R_L) = \frac{p_1}{R_L - p_2} + p_3 \tag{5.4}$$

$$D_{e,R}(R_L, R_R) = \frac{p_4}{R_R - p_5 * R_L - p_6} + p_7 \tag{5.5}$$

5.4.2 R-Q Model

In this model [109], instead of distortion, quality metric (PSNR) is modelled. Besides, encoding quantization parameter (QP) is formularized for R and Q values. QP and Q values are calculated as a quadratic function of logarithm of R as shown in Equations 5.6,5.7 and 5.8. We use the same model for left view and integrated logarithm of rate of left view for the model of right view as shown in Equations 5.9 and 5.10.

$$QP_L(R_L) = q_1 * [logR_L]^2 + q_2 * [logR_L] + q_3 \tag{5.6}$$

$$Q_L(R_L) = q_4 * [logR_L]^2 + q_5 * [logR_L] + q_6 \tag{5.7}$$

$$Q_L(R_L) = q_7 * [QP_L]^2 + q_8 * [QP_L] + q_9 \tag{5.8}$$

$$QP_R(R_L, R_R) = q_{10}*[logR_L]^2+q_{11}*[logR_L]+q_{12}*[logR_R]^2+q_{13}*[logR_R]+q_{14}*[logR_L]*[logR_R]+q_{15} \tag{5.9}$$

$$Q_R(R_L, R_R) = q_{16}*[logR_L]^2+q_{17}*[logR_L]+q_{18}*[logR_R]^2+q_{19}*[logR_R]+q_{20}*[logR_L]*[logR_R]+q_{21} \tag{5.10}$$

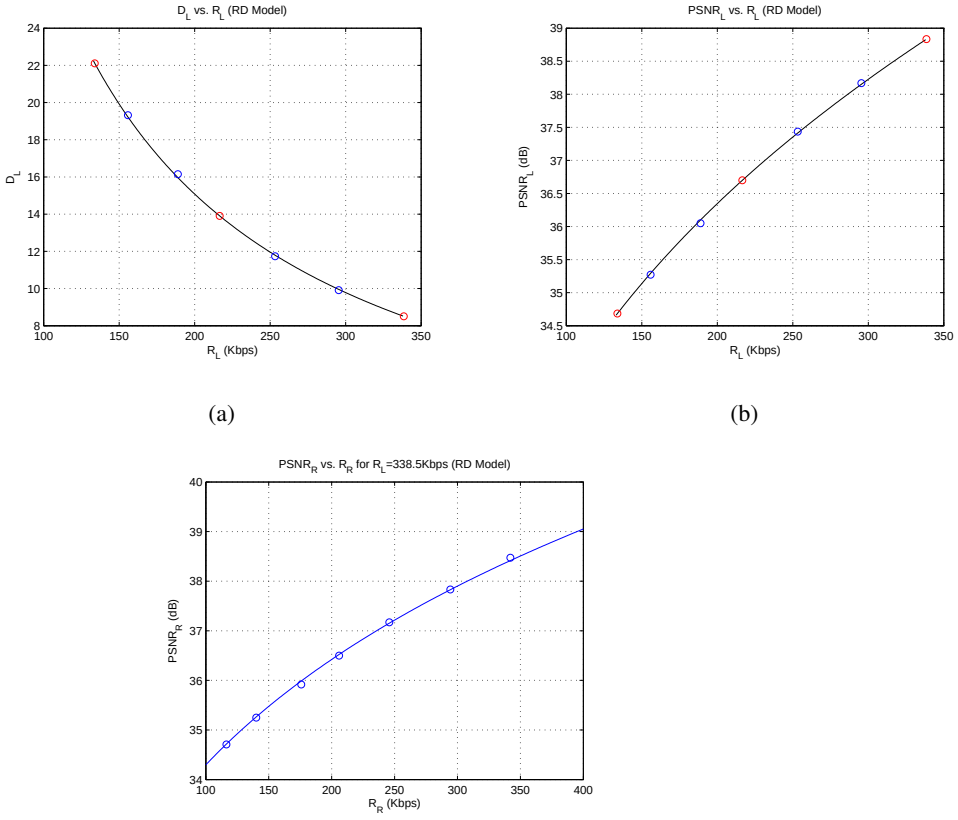
5.4.3 Evaluation of the encoder distortion models

In order to model the performance of the models, we encoded Rhine Valley video with several QP parameters with MVC encoder to obtain many RD curve samples. We chose some of the RD samples and inserted into the curve fitting tool. By using the remaining RD samples, we compare the performance of the models. We used a general purpose nonlinear curve fitting tool which uses the Levenberg-Marquardt method with line search [93] for parameters p4,p5,p6,p7. For all other parameters, least square fitting is used. After calculating the model parameters, R&D/Q values are calculated for all points. In Figure 5.22-5.25, it can be seen

Table 5.3: Performance of both models using several error metrics.

	Left			Right			Joint		
Model	MAE	MSE	MAX	MAE	MSE	MAX	MAE	MSE	MAX
R-D	0.0156	0.00051	0.0484	0.0406	0.0027	0.1223	0.0229	0.00087	0.0776
R-Q	0.0134	0.00041	0.0455	0.0156	0.00037	0.0471	0.0106	0.00019	0.0463

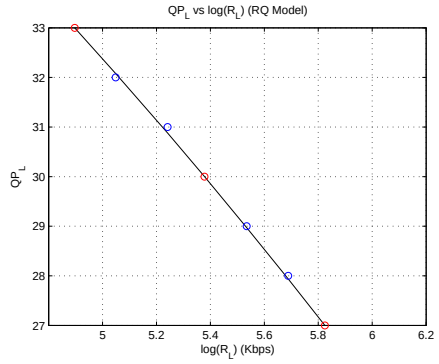
that both models fit the data points. In order to objectively compare the models, mean absolute error (MAE), mean square error (MSE) and maximum error (MAX) in dB are calculated and tabulated in Table 5.3. According to error metrics, R-Q model seems to fit data points more accurately. R-Q model also is easier to calculate by least square methods and gives relation between QP parameters as well. We will use R-Q model in the overall distortion model of the system.



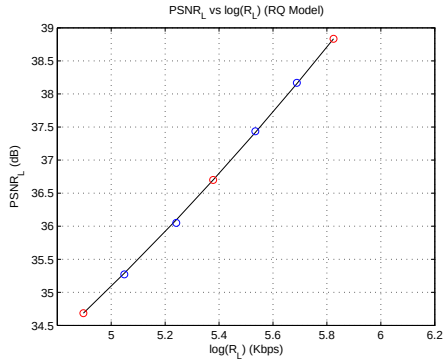
(a) (b)

(c)

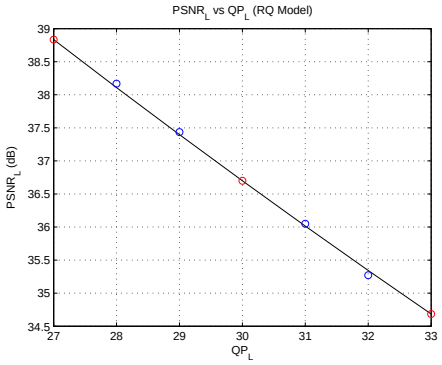
Figure 5.22: Model fitting results for RD Model (a)-(b) Left, (c) Right



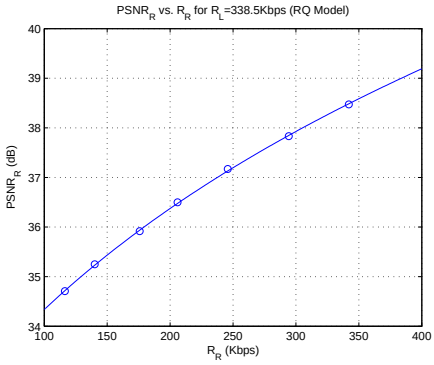
(a)



(b)



(c)



(d)

Figure 5.23: Model fitting results for RQ Model(a)-(c) Left, (d) Right

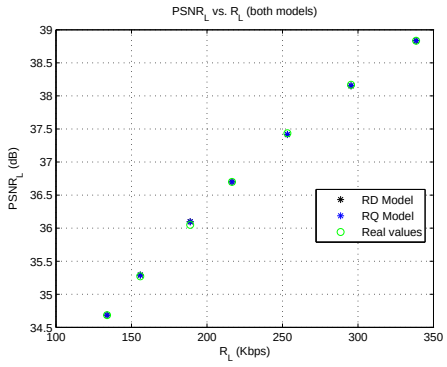
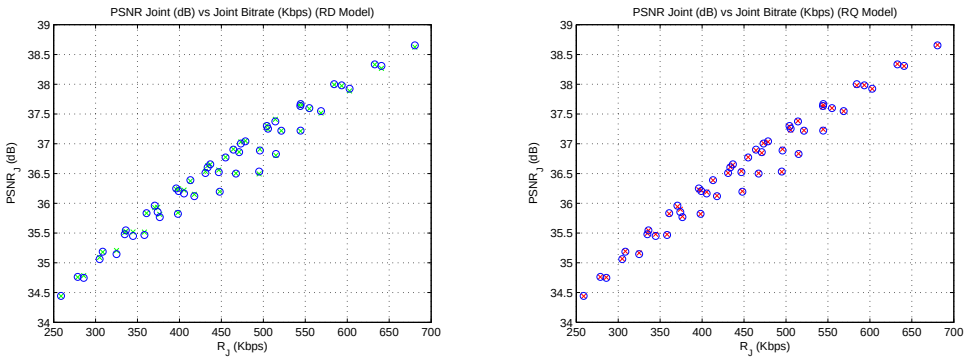


Figure 5.24: Comparison of R-D Model and R-Q Model for Left video



(a) (b)

Figure 5.25: Model for Left & Right video (a) RD, (b) RQ

5.5 Decoder Distortion Modelling

Since the error concealment used in the system is basic frame/slice concealment, a basic distortion modeling in used for decoder distortion by assuming propagation of distortion is negligible. For each macroblock (MB) available collocated MB from the previous frame is used for concealment. In order to calculate distortion, mean square error (MSE) is calculated between the current frame and the previous frame in the same view. Since video is encoded in slice mode, average number of MBs in a slice, N_{mb} is found by dividing the total number of MBs in the stream to the total number of slices. Average distortion of a single MB, D_{mb} value for each view is calculated and used in distortion modeling. According to slice loss rate (SLR) calculated using error characteristics and error protection rates), distortion is calculated using Equation 5.11 and 5.12.

$$D_{loss,L} = \alpha_L * SLR_L * D_{mb,L} * N_{mb,L} \tag{5.11}$$

$$D_{loss,R} = \alpha_R * SLR_R * D_{mb,R} * N_{mb,R} \tag{5.12}$$

α_L and α_R are constants and calculated using a small set of simulation results. For SNR 19, α_L=0.7310 and α_R=0.6993 and the error of the model calculated over whole simulation set is 0.5508 for MSE and 2.7829 for MAX error. Details of simulations setup is given in Section 5.7.

5.6 Error Protection Modelling

In the encoding of the videos, slice size is set as 1300 bytes and TS packet size is 184 bytes. A single slice will be transmitted by approximately 6 TS packets. Slice Loss Rate (SLR) can be calculated as follows:

$$SLR = 1 - (1 - TSPLR)^6 \tag{5.13}$$

In Equation 5.13, TSPLR represents TS Packet loss rates. TSPLR before MPE-FEC correction is calculated for the error patterns used in the system as shown in Figure 5.10. TSPLR after MPE-FEC correction is calculated as follows:

$$TSPLR_{after} = \begin{cases} 0 & \text{if } \rho > (1.10 * TSPLR_{before}) \\ (1 - \rho) * TSPLR_{before} & \text{otherwise} \end{cases}$$

5.7 Simulations

In order to compare end-to-end model and also investigate the behaviour of the system for the given channel models, extensive simulations are employed with different channel characteristics. Optimum parameters are found exhaustively and compared with modelling results.

5.7.1 Encoding

Video is encoded using MVC with simplified prediction scheme. Slice coding mode is on with 1300 slice size. Left view is encoded independently and right view depends on left view. Quantization Parameter for left view (QP_L) and Quantization Parameter for right view (QP_R)) are used to control rate and distortion of the encoded sequences. RhineValley video with 64 frames are encoded with Intra period of 8. Video is 12.5 fps with 432x240 resolution. QP_L and QP_R are varied between 27 and 33.

5.7.2 FEC Protection

The simulations are carried out using the same DVB-H physical layer transmission parameters as in [18]. Total TS bitrate is selected as 1024 Kbps. On average IP overhead is about %4

and TS overhead is about %30. Using these approximations, highest quality encoded stream (QP_L=27 and QP_R = 27) will be 972.38 Kbps and lowest quality encoded stream (QP_L=33 and QP_R = 33) will be 396.45 Kbps without FEC protection. The remaining bitrate will be used by FEC protection. Since each view (left and right) will be sent in a separate burst, we can use different FEC protection ratios. For each encoded video (49 cases as shown in Table 5), we calculate the remaining bitrate for FEC and then allocate FEC ratios to both left and right channels with several schemes.

Depending on the available FEC bitrate, we can have up to 20 different ratios between left and right. One of the scheme for each bitstream is EEP where FEC protection is approximately equal. Other schemes are several UEP schemes by changing the number of RS columns of left and right channels. For high quality encoded bitstreams, we do not have enough bitrate left to have several UEP schemes. For low quality encoded bitstreams, we use a step size of 4 RS columns making at most 20 UEP schemes since there can be a maximum of 64 RS columns. When we assign FEC rates, we are limited with the selection of ADT and RS columns. Since encoder is using fixed QP values, size of each GOP will be different. We are using maximum GOP size for each video to selected RS column sizes and this results in a smaller (higher protecting) total effective FEC rates.

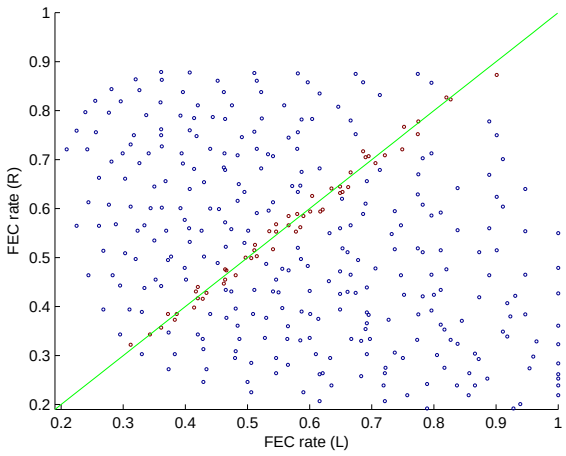


Figure 5.26: EEP and UEP schemes used in simulations

Total number of schemes are 401 and shown in the Figure 5.26. EEP schemes are shown with red colors and UEP schemes are with blue colors. Because of the integer number of columns and effective FEC rates, EEP rates will not be exactly 1 (some of the points will not be on

x=y line). We are classifying schemes as EEP is ratio of FEC rates of left to right is between 0.95 and 1.05.

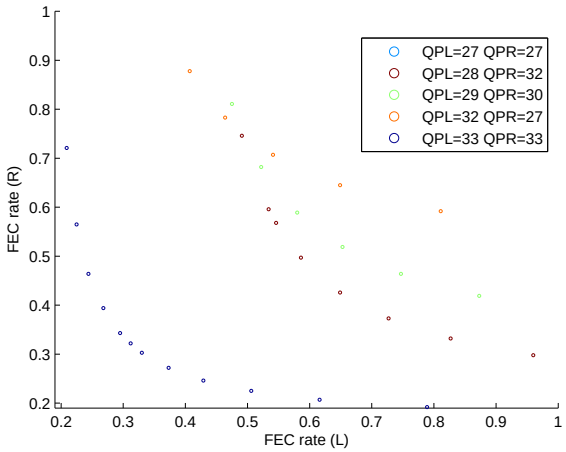


Figure 5.27: FEC distribution of selected encoded bitstreams

In Figure 5.27, we are showing the schemes used with some of the selected encoded bitstreams. It can seen from the figure that, with smaller QPs, total bitrate will be higher and only a couple of schemes can be tested. With higher QPs, more schemes are available.

5.7.3 Simulations Results

Experiments are repeated 50 times for each channel SNR (18-21) and averaged. The results for each SNR is given in Figure 5.28 - 5.31.

5.7.4 Modelling Results

For all pairs of encoding and error protection parameters, final distortion is calculated using the models defined in the previous sections. The error of the model is 1.3156 for MSE and 2.9892 for MAX error for SNR 19 and shown in Figure 5.32.

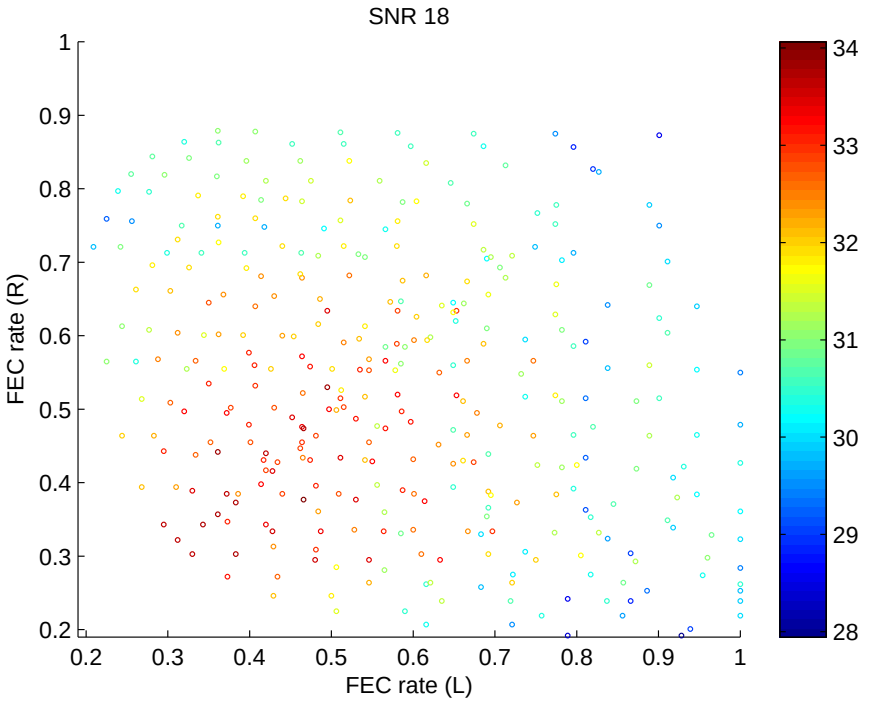


Figure 5.28: Simulation results for SNR 18

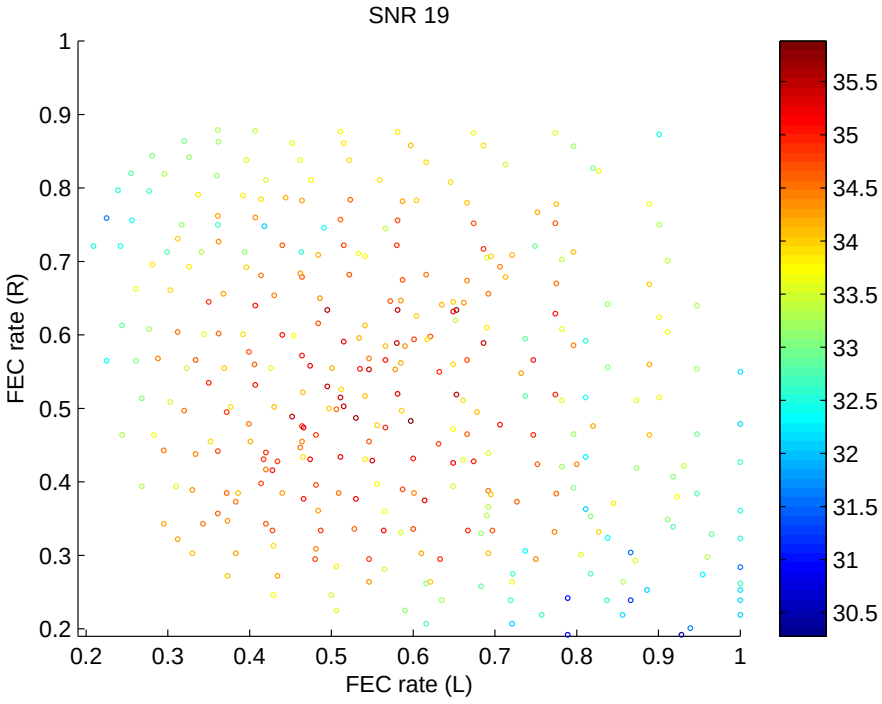


Figure 5.29: Simulation results for SNR 19

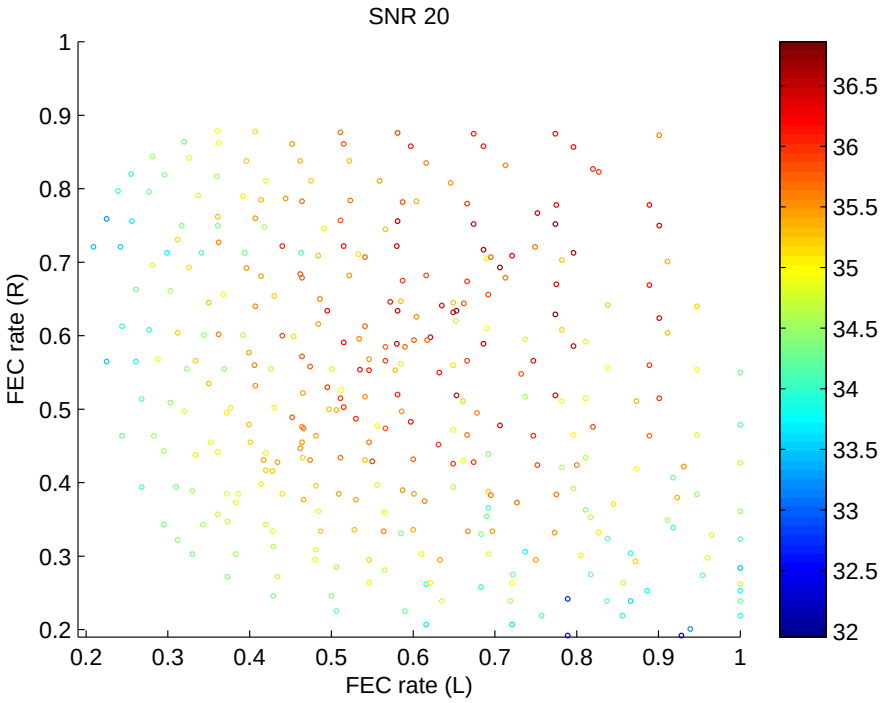


Figure 5.30: Simulation results for SNR 20

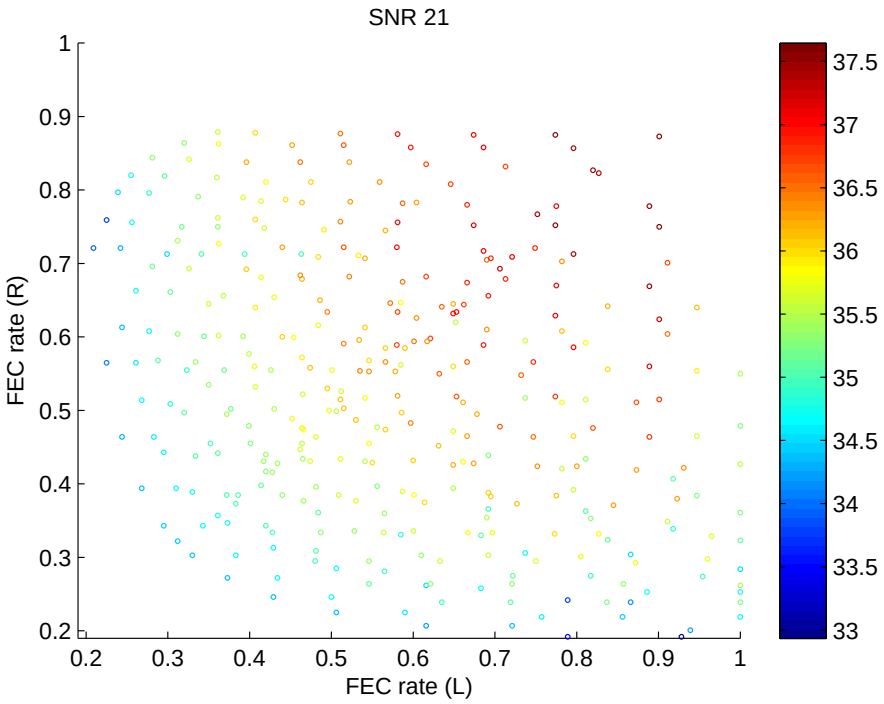


Figure 5.31: Simulation results for SNR 21

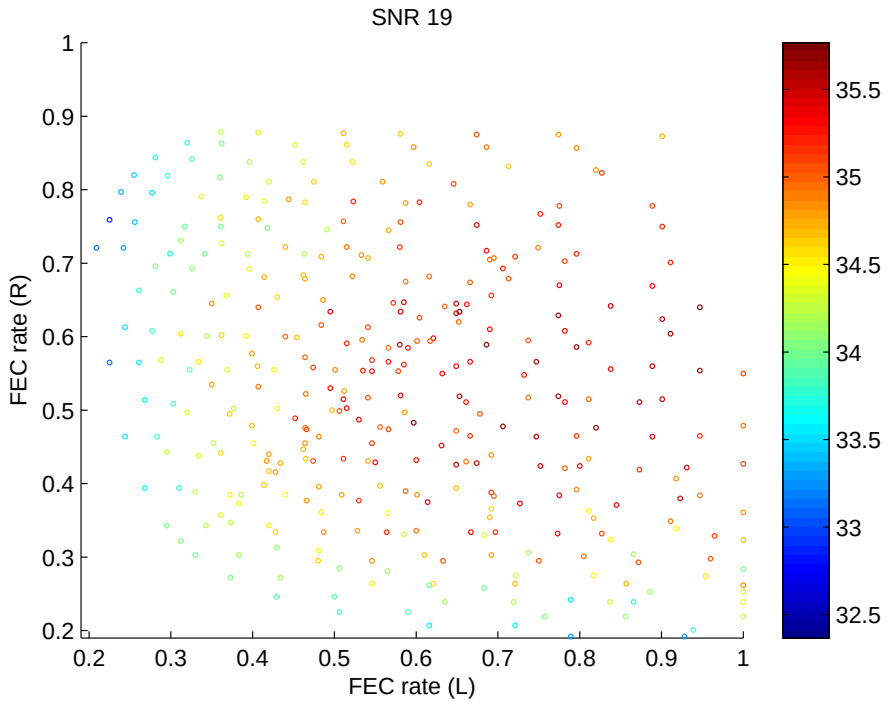


Figure 5.32: Modeling results for SNR 19

5.8 Conclusions

For an error resilient 3D video broadcast system over DVB-H, we have analyzed the encoding and error protection of the system. In order to optimize these parameters for different channel conditions, we performed extensive simulations. It is clear that increasing error protection is essential for cases where channel SNR is below 20. If channel SNR is 20 or more, protecting bitstreams with 0.75-1 FEC ratio with EEP or close to EEP will be sufficient to have the optimum performance.

When the channel SNR decreases, we can clearly see using more FEC protection is required and also protecting left bitstream more is important. For channel SNR 19, FEC protection for left view should be from 0.5 upto 0.8 and FEC protection for right view should be from 0.7 upto 1 to have optimum performance. For channel SNR 18, FEC protection for left view should be from 0.4 upto 0.6 and FEC protection for right view should be from 0.6 upto 0.8 to have optimum performance.

In our system, we used IPP and simplified prediction structure due to its simplicity and also

structure more robust to error losses. That is why, the results for higher channel rates, EEP and close to EEP is preferred. However with prediction structures, where right view is more dependent on the left view, protecting left bitstream more will be required. This can be true for different content as well. For more stationary sequences interview prediction will dominate and protection of left bitstream will be more important. However performing extensive tests is not easy and time consuming. In order to decrease the testing process, we propose a methodology to first model the end-to-end distortion characteristics for a given channel loss distribution. As seen from the results, this modeling provides a similar insight with a close match of the winning schemes and requires only a small percentage of the simulation results.

CHAPTER 6

Conclusion

This thesis proposes new techniques for error resilient coding and streaming for multi-view video. We are dealing with coding, decoding, error protection and we proposed several techniques for improving MVV streaming/broadcasting systems.

3D Coding schemes depend on the decoding environment, transmission environment and subjective quality. Different coding techniques and standards are available depending on the representation format.

For stereoscopic videos, a novel asymmetric coding technique is proposed using downsampling one of the views spatially or temporally based on the well-known theory that the human visual system can perceive high frequencies in 3D from the higher quality view. Proposed method is compared, with respect to coding efficiency, with several possible downsampling schemes. Subjective tests are also employed to verify the results. Selection of schemes is highly dependent on the content and content based encoding is also proposed by extending this work. Using the proposed approach stereoscopic videos can be coded at a rate upto 1.2 times that of monoscopic videos with little visual quality degradation. Also this technique is used in the implementation of the first multiple description coding of stereoscopic videos. Proposed asymmetric coding ideas can easily be extended for MVV using MVC standard.

For stereoscopic videos encoded for mobile devices, we analyzed possible stereoscopic encoding schemes since previous investigations are different from the requirements of mobile devices. Experiments with two encoders with several configurations are carried out to figure out coding efficiency of different tools. Decoding tests also give useful information about the possible processing performance when implemented on a mobile device platform. We recommend two schemes depending on the processing power and memory of the mobile device. For

low end devices H.264/AVC monoscopic codec with IPP+CABAC settings over interleaved stereoscopic content can be used and for higher end devices H.264/AVC MVC extension with simplified referencing structure is recommended.

For 3D decoding, multi-threading approaches are investigated and MVC standard compliant MVV decoding architecture enabling multi-threaded decoding is proposed. In order to achieve multi-threaded decoding, simplified structures are proposed depending on the number of cores in the system. Proposed approach brings negligible loss of encoding efficiency and minimum processing overhead compared with the standard solution or non-multi threaded approaches. A test-bed is implemented to test the proposed ideas in a real-time multi-threaded decoding system. Benchmark tests reveal the necessity for multi-threaded decoding for N-view multi-view display systems with HD resolutions. Another advantage of the proposed system is its simplicity for implementation.

In this thesis, two end-to-end system is implemented for 3D streaming and 3D broadcasting. Systems are used for offline simulations for error protection and optimization. Systems are also used as real-time demonstrators and supports backward compatibility for legacy clients.

We implemented an end-to-end stereoscopic video streaming system with content-adaptive stereoscopic video coding. It is implemented by modifications to available open source monocular streaming platforms. This system is improved with a standards-based, flexible, end-to-end MVV streaming architecture. This system supports several different display types and different representations. Multi-threaded decoding, multi-view error concealment and error protection added to the system as well. Another advantage of the implemented system is adaptive streaming of MVV using MVC standard. Users receive only the required parts of the 8-view encoded bitstream depending on their display capability and user preferences.

If there are clients using different types of display and capabilities, we advise to use MVC encoding using simplified prediction structure. This enables adaptation of encoded bitstream online and better usage of network. However if the display types of the users are all the same or mostly the same, using full prediction structures will increase the compression efficiency and used network bandwidth. Nevertheless, variety of multi-view displays with changing number of views, encourage us to use our propose approach.

Error-resilient stereoscopic video streaming system using rateless Raptor codes for error pro-

tection is implemented. We suggest a methodology for modeling the end-to-end RD characteristic of this system. We modeled RD curve of video encoder, performance of channel codec and distortion caused by packet losses. Using this model we optimized the encoding parameters of the video codec and code rate of the Raptor coder. The simulation results clearly demonstrate the significant quality gain against the non-optimized schemes.

We implemented first DVB-H based 3D broadcast system proposed in the literature using current standards. We integrated error resilient tools into the system. By using both stereo video and video plus depth, we tested extensively to show the importance and characteristics of several error resilient tools. According to simulations results we concluded slice mode outperforms full frame slice mode in most of the cases. MVC encoding outperforms simulcast encoding. EEP (and UEP closer to EEP) schemes have a better quality than other FEC modes since used prediction structure minimizes the dependencies between views. In this work, only a couple of UEP schemes are tested.

We also compared the performance of the system for different slice sizes. We showed that for low motion sequences, overall performance is better using full frame size slices and for high motion sequences it is better to use smaller slice sizes like 500 or 750.

In order to optimize encoding and error protection rates, we employed more extensive tests using MVC encoded video. Similar to modeling for Internet based streaming system, we modeled the system using models of RD curve of video encoder, channel codec and distortion caused by packet losses. Using end-to-end RD model, parameters are optimized. We have shown that increasing error protection is essential for cases where channel SNR is below 20. If channel SNR is 20 or more, protecting bitstreams with 0.75-1 FEC ratio with EEP or close to EEP will be sufficient to have the optimum performance. When the channel SNR decreases, using more FEC protection is required and also protecting left bitstream more is important. For channel SNR 19, FEC protection for left view should be from 0.5 upto 0.8 and FEC protection for right view should be from 0.7 upto 1 to have optimum performance. For channel SNR 18, FEC protection for left view should be from 0.4 upto 0.6 and FEC protection for right view should be from 0.6 upto 0.8 to have optimum performance.

These results are valid for the given prediction structure. For prediction structures where interview prediction is enabled for non-anchor pictures and also for stationary sequences, interview prediction will be dominated. In such cases, protection of left bitstream will be

more important. However performing extensive tests is not easy and time consuming. In order to decrease the testing process, we propose a methodology to first model the end-to-end distortion characteristics for a given channel loss distribution. As seen from the results, this modeling provides a similar insight with a close match of the winning schemes and requires only a small percentage of the simulation results.

As future work, different prediction structures and using Hierarchical B-pictures can be examined. Encoding modeling and performance in lossy networks of these coding schemes might be different from the current schemes used in this thesis.

REFERENCES

[1] Project, E.F.: Mobile3dtv: Mobile 3dtv content delivery over dvb-h system (2009)

[2] Smolic, A., Mueller, K., Merkle, P., Fehn, C., Kauff, P., Eisert, P., Wiegand, T.: 3D Video and Free Viewpoint Video–Technologies, Applications and MPEG Standards. (In: Proceedings of the IEEE International Conference on Multimedia and Expo (ICME'06)) 2161–2164

[3] Fehn, C.: A 3D-TV system based on video plus depth information. In: ASILOMAR CONFERENCE ON SIGNALS SYSTEMS AND COMPUTERS. Volume 2., IEEE; 1998 (2003) 1529–1533

[4] Merkle, P., Smolic, A., Mueller, K., Wiegand, T.: Multi-view video plus depth representation and coding. In: Proceedings of ICIP. (2007) 201–204

[5] Shade, J., Gortler, S., He, L., Szeliski, R.: Layered depth images. In: Proceedings of the 25th annual conference on Computer graphics and interactive techniques, ACM New York, NY, USA (1998) 231–242

[6] Girod, B., Chang, C., Ramanathan, P., Zhu, X.: Light field compression using disparity-compensated lifting. In: Proc. of the IEEE Intl. Conf. on Acoustics, Speech and Signal Processing 2003. Volume 4., (Citeseer) 761–764

[7] Onural, L., Sikora, T., Ostermann, J., Smolic, A., Civanlar, M., Watson, J.: An assessment of 3DTV technologies. In: Proc. of NAB Broadcast Engineering Conf. (2006) 456–467

[8] Onural, L., Gotchev, A., Ozaktas, H., Stoykova, E.: A survey of signal processing problems and tools in holographic three-dimensional television. IEEE Trans. Circuits Syst. Video Technol **17** (2007) 1631–1646

[9] Iddan, G., Yahav, G.: 3D imaging in the studio (and elsewhere...). In: Proc. SPIE. Volume 4298. (2001) 48–55

[10] Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International journal of computer vision **47** (2002) 7–42

[11] Vetro, A., Pandit, P., Kimata, H., Smolic, A.: Joint draft 8.0 on multiview video coding. Joint Video Team (JVT) of ISO/IEC MPEG ITU-T VCEG ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6 (2007)

[12] : ISO/IEC JTC1/SC29/WG11, ISO/IEC 23002-3 Representation of auxiliary video and supplemental information (2007) Doc. N8259.

[13] Kajiki, Y., Yoshikawa, H., Honda, T.: Ocular accommodation by super multi-view stereogram and 45-view stereoscopic display. In: Proc. Int'l Display Workshops 11. (1996) 489–492

[14] Aksay, A., Bilen, C., Kurutepe, E., Ozcelebi, T., Bozdagi Akar, G., Civanlar, R., Tekalp, M.: Temporal and spatial scaling for stereoscopic video compression. In: Proc. EUSIPCO'06, Sept. 4-8, Florence, Italy (2006)

[15] Aksay, A., Pehlivan, S., Kurutepe, E., Bilen, C., Ozcelebi, T., Akar, G., Civanlar, M., Tekalp, A.: End-to-end stereoscopic video streaming with content-adaptive rate and format control. Signal Processing: Image Communication **22** (2007) 157–168

[16] Gurler, C.G., Aksay, A., Akar, G., Tekalp, A.: Architectures for Multi-Threaded MVC-Compliant Multi-View Video Decoding and Benchmark Tests. Signal Processing: Image Communication, Special Issue on Breakthrough Architectures for Image and Video Systems (2010) accepted.

[17] Tan, A., Aksay, A., Akar, G., Arikan, E.: Rate-distortion optimization for stereoscopic video streaming with unequal error protection. EURASIP Journal on Applied Signal Processing **2009** (2009)

[18] Bici, M., D., B., M., D.A., Aksay, A., Akar, G.: Error Resilient 3D Video Transmission over DVB-H. (IEEE Selected Topics in Signal Processing) submitted.

[19] Aksay, A., Akar, G.: Optimization of Encoding and Error Protection Parameters for 3D Video Broadcast over DVB-H. (ACM Multimedia System Journal) submitted.

[20] : ITU-T Rec. H.264 — ISO/IEC 14496-10 AVC, "Advanced Video Coding for Generic Audiovisual Services" (2009)

[21] Wiegand, T., Sullivan, G., Bjontegaard, G., Luthra, A.: Overview of the H. 264/AVC video coding standard. IEEE Transactions on circuits and systems for video technology **13** (2003) 560–576

[22] Smolic, A., Kimata, H.: Report on 3dav exploration. ISO/IEC JTC1/SC29/WG11 Doc N5878 (2003)

[23] Vetro, A., Matusik, W., Pfister, H., Xin, J.: Coding approaches for end-to-end 3D TV systems. In: Picture Coding Symposium, Citeseer (2004)

[24] : Survey of algorithms used for multi-view video coding (mvc). ISO/IEC JTC1/SC29/WG11 Doc N6909 (2005)

[25] Bilen, C., Aksay, A., Bozdagi Akar, G.: A multi-view video codec based on H.264. In: Proc. IEEE Conf. Image Proc. (ICIP), Oct. 8-11, Atlanta, USA (2006)

[26] Schwarz, H., Marpe, D., Wiegand, T.: Analysis of hierarchical B pictures and MCTF. In: Proc. ICME, Citeseer (2006) 1929–1932

[27] Merkle, P., Smolic, A., Mueller, K., Wiegand, T.: Efficient prediction structures for multiview video coding. IEEE Transactions on circuits and systems for video technology **17** (2007) 1461–1473

[28] Fehn, C., Kauff, P., De Beeck, M., Ernst, F., Ijsselsteijn, W., Pollefeys, M., Van Gool, L., Ofek, E., Sexton, I.: An evolutionary and optimised approach on 3D-TV. In: Proc. of IBC, Citeseer (2002)

[29] Merkle, P., Wang, Y., Mueller, K., Smolic, A., Wiegand, T.: Video plus depth compression for mobile 3D services. In: Proc. IEEE 3DTV Conference, Potsdam, Germany. (2009)

[30] Fehn, C.: Depth-image-based rendering(DIBR), compression, and transmission for a new approach on 3D-TV. In: Proceedings of SPIE. Volume 5291. (2004) 93–104

[31] Zhang, L., Tam, W.: Stereoscopic image generation based on depth images for 3 D TV. IEEE transactions on Broadcasting **51** (2005) 191–199

[32] Kurutepe, E., Aksay, A., Bilen, C., Gürler, C., Sikora, T., Akar, G., Tekalp, A.: A standards-based, flexible, end-to-end multi-view video streaming architecture. In: Packet Video 2007. (2007) 302–307

[33] : ISO/IEC JTC1/SC29/WG11 Vision on 3D Video (2009) Doc. N10357.

[34] Julesz, B.: Foundations of cyclopean perception. The University of Chicago Press (1971)

[35] Woo, W., Ortega, A.: Optimal blockwise dependent quantization for stereo image coding. IEEE Transactions on Circuits and Systems for Video Technology **9** (1999) 861–867

[36] Dinstein, I., Kim, M., Henik, A., Tzelgov, J.: Compression of stereo images using subsampling and transform coding. Optical engineering(Bellingham. Print) **30** (1991) 1359–1364

[37] Stelmach, L., Tam, W., Meegan, D.: Stereo image quality: effects of spatio-temporal resolution. In: Proceedings of SPIE. Volume 3639. (1999) 4

[38] Reichel, J., Schwarz, H., Wien, M.: Scalable video coding - working draft 3. JVT-P201 (2005)

[39] Segall, C.: Study of upsampling/down-sampling for spatial scalability. JVT-Q083, Nice, FR, PL **14** (2005) 21

[40] BT, I.: 1438, Subjective assessment of stereoscopic television pictures, Recommendation ITU-R BT. 1438, ITU Telecom. Sector of ITU (2000)

[41] Schertz, A.: Source coding of stereoscopic television pictures. In: Image Processing and its Applications, 1992., International Conference on. (1992) 462–464

[42] Norkin, A., Aksay, A., Bilen, C., Akar, G., Gotchev, A., Astola, J.: Schemes for multiple description coding of stereoscopic video. Lecture Notes in Computer Science **4105** (2006) 730

[43] Willner, K., Ugur, K., Salmimaa, M., Hallapuro, A., Lainema, J.: Mobile 3D Video Using MVC and N800 Internet Tablet. In: 3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video, 2008. (2008) 69–72

[44] Cho, S., Hur, N., Kim, J., Yun, K., Lee, S.: Carriage of 3D audio-visual services by T-DMB. Electronics and Telecommunications Research Institute, Republic of Korea, in Proc ICME (2006)

[45] Flack, J., Harrold, J., Woodgate, G.: A prototype 3D mobile phone equipped with a next-generation autostereoscopic display. Proceedings of the SPIE Stereoscopic Displays and Virtual Reality Systems XIV, San Jose, CA, USA (2007)

[46] Kwon, J., Kim, M., Choi, C.: Multiview Video Service Framework for 3D Mobile Devices. In: Proceedings of the 2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing-Volume 00, IEEE Computer Society (2008) 1231–1234

[47] : Updated call for proposals on multi-view video coding. ISO/IEC JTC1/SC29/WG11 Doc. N7567 (2005)

[48] Project, E.F.: (3dtv network of excellence, "public software and data repository,")

[49] Knorr, S., Sikora, T.: An image-based rendering (IBR) approach for realistic stereo view synthesis of TV broadcast based on structure from motion. In: IEEE Int. Conf. on Image Processing (ICIP), San Antonio, USA, Citeseer (2007)

[50] Coordination, H.S.: (H.264/avc reference software jm 14.2)

[51] Pandit, P., Vetro, A., Chen, Y.: Wd 2 reference software for mvc. ITU-T JVTAB207 (2008)

[52] Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: From error visibility to structural similarity. IEEE transactions on image processing **13** (2004) 600–612

[53] Saponara, S., Blanch, C., Denolf, K., Bormans, J.: Data transfer and storage complexity analysis of the AVC/JVT codec on a tool-by-tool basis. Joint Video Team (JVT), Doc. JVT-Dl38, Klagenfurt, Austria (2002)

[54] CNET: (Intel 80-core processor)

[55] Gitorious: (ffmpeg-mt in ffmpeg)

[56] Sihn, K., Baik, H., Kim, J., Bae, S., Song, H.: Novel approaches to parallel H. 264 decoder on symmetric multicore systems. In: Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing-Volume 00, IEEE Computer Society (2009) 2017–2020

[57] Meenderinck, C., Azevedo, A., Juurlink, B., Alvarez Mesa, M., Ramirez, A.: Parallel scalability of video decoders. Journal of Signal Processing Systems **57** (2009) 173–194

[58] Gurler, C., Aksay, A., Akar, G., Tekalp, A.: Multi-Threaded Architectures and Benchmark Tests For Real-Time Multi-View Video Decoding. In: Proceedings of the 2009 IEEE ICME. (2009)

[59] Ugur, K., Liu, H., Lainema, J., Gabbouj, M., Li, H.: Parallel encoding-decoding operation for multi-view video coding with high coding efficiency. (In: Proceedings of the Conference on True Vision, Capture, Transmission, and Display of 3D Video (3DTV'07)) 1–4

[60] Inition: (Sharp actius al3du)

[61] GmbH, N.: (Newsight complete 3d autostereoscopic hardwre and software solutions: Displays)

[62] Su, Y., Vetro, A., Smolic, A.: Common test conditions for multiview video coding. ITU-T JVTU211 (2007)

[63] Bjontegaard, G.: Calculation of average PSNR differences between RD-curves. Doc. VCEG- M (2001)

[64] Chang, S., Zhong, D., Kumar, R.: Real-time content-based adaptive streaming of sports videos. In: Proceedings of the IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL'01), IEEE Computer Society (2001) 139

[65] Mohan, R., Smith, J., Li, C.: Adapting multimedia internet content for universal access. IEEE Transactions on Multimedia **1** (1999) 104–114

[66] Ozcelebi, T., Tekalp, A., Civanlar, M.: Delay-distortion optimization for content-adaptive video streaming. IEEE Transactions on Multimedia **9** (2007) 826–836

[67] Van Beek, P., Smith, J., Ebrahimi, T., Suzuki, T., Askelof, J.: Metadata-driven multimedia access. IEEE Signal Processing Magazine **20** (2003) 40–52

[68] G., G.: An open-standards based server-client model for robust streaming of 3DTV over the Internet. M.S. Thesis, Koç University (2008)

[69] Bilen, C., Aksay, A., , Akar, G.: Two Novel Methods for Full Frame Loss Concealment in Stereo Video. In: Packet Video 2007. (2007)

[70] Lin, L., Ortega, A.: Bit-rate control using piecewise approximated rate-distortioncharacteristics. Circuits and Systems for Video Technology, IEEE Transactions on **8** (1998) 446–459

[71] Ronda, J., Eckert, M., Jaureguizar, F., Garcia, N.: Rate control and bit allocation for MPEG-4. Circuits and Systems for Video Technology, IEEE Transactions on **9** (1999) 1243–1258

[72] Ribas-Corbera, J., Lei, S.: Rate control in DCT video coding for low-delay communications. Circuits and Systems for Video Technology, IEEE Transactions on **9** (1999) 172–185

[73] Sermadevi, Y., Hemami, S.: Linear programming optimization for video coding under multiple constraints. In: Proceedings of Data Compression Conference. (2003) 53–62

[74] Chakareski, J., Apostolopoulos, J., Girod, B.: Low-complexity rate-distortion optimized video streaming. Image Processing, IEEE International Conference on **3** (2004)

[75] Yang, E., Yu, X.: Rate Distortion Optimization for H. 264 Interframe Coding: A General Framework and Algorithms. Image Processing, IEEE Transactions on **16** (2007) 1774–1784

[76] Chou, P.: Rate-distortion optimized streaming of packetized media. Multimedia, IEEE Transactions on **8** (2006) 390–404

[77] Girod, B.: Rate-Distortion Analysis and Streaming of SP and SI Frames. Circuits and Systems for Video Technology, IEEE Transactions on **16** (2006) 733–743

[78] Conklin, G., Greenbaum, G., Lillevold, K., Lippman, A., Reznik, Y., Inc, R., Seattle, W.: Video coding for streaming media delivery on the Internet. Circuits and Systems for Video Technology, IEEE Transactions on **11** (2001) 269–281

[79] Girod, B., Stuhlmuller, K., Link, M., Horn, U.: Packet loss resilient internet video streaming. In: Proc. SPIE Visual Commun. Image Processing. (1999)

[80] Cai, H., Zeng, B., Shen, G., Xiong, Z., Li, S.: Error-resilient unequal error protection of fine granularity scalable video bitstreams. Applied Signal Processing, EURASIP Journal on (2006)

[81] Pei, Y., Modestino, J.: H.263+ packet video over wireless IP networks using rate-compatible punctured turbo (rcpt) codes with joint source-channel coding. In: Proc. of the IEEE ICIP. (2002)

[82] Byers, J., Luby, M., Mitzenmacher, M., Rege, A.: A digital fountain approach to reliable distribution of bulk data. In: Proceedings of ACM Sigcomm. (1998)

[83] Luby, M.: LT codes. In: Proc. of the 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS). (2002) 271–282

[84] Maymounkov, P.: Online codes. Research Report TR2002-833, New York University, Nov (2002)

[85] Shokrollahi, A.: Raptor Codes. Information Theory, IEEE Tans. on **52** (2006) 2551–2567

[86] Wagner, J., Chakareski, J., Frossard, P.: Streaming of scalable video from multiple servers using rateless codes. In: Proc. IEEE Conf. on Multimedia and Expo (ICME), Toronto, Canada (2006)

[87] Luby, M., Gasiba, T., Stockhammer, T., Watson, M.: Reliable Multimedia Download Delivery in Cellular Broadcast Networks. Broadcasting, IEEE Transactions on **53** (2007) 235–246

[88] Luby, M., Watson, M., Gasiba, T., Stockhammer, T., Xu, W.: Raptor codes for reliable download delivery in wireless broadcast systems. In: Proc. of the IEEE CCNC. (2006)

[89] Fernando, J., Loo, W., Arachchi, K., Yip, H., Malcolm, P.: Joint source and channel coding for h.264 compliant stereoscopic video transmission. In: Canadian Conf. on Electrical and Computer Engineering. (2005)

[90] Vetro, A., Pandit, A., Kimata, H., Smolic, A.: Joint Draft 4.0 on Multiview Video Coding. Joint Video Team **Doc. JVT-X209** (2007)

[91] Varsa, V., Hannuksela, M., Wang, Y.: Non-normative error concealment algorithms. ITU-T VCEG **62** (2001)

[92] Stuhlmuller, K., Farber, N., Link, M., Girod, B.: Analysis of video transmission over lossy channels. Selected Areas in Communications, IEEE Journal on **18** (2000) 1012–1032

[93] Mor, J.: The Levenberg-Marquardt algorithm: Implementation and theory. Lecture Notes in Mathematics **630** (1977) 105–116

[94] Ozbek, N., Tekalp, A., Tunali, E.: Rate Allocation Between Views in Scalable Stereo Video Coding using an Objective Stereo Video Quality Measure. Acoustics, Speech and Signal Processing ICASSP, IEEE International Conference on (2007)

[95] Gill, P., Murray, W., Wright, M.: Practical optimization. London: Academic Press (1981)

[96] Tan, A., Aksay, A., Bilen, C., Akar, G., Arikan, E.: Error resilient layered stereoscopic video streaming. In: 3DTV-Conference, Kos, Greece (2007)

[97] Furht, B., Ahson, S.: Handbook of Mobile Broadcasting: DVB-H, DMB, ISDB-T, and MediaFLO. Auerbach Publications (2008)

[98] Project, E.F.: 3dphone project (2008)

[99] Faria, G., Henriksson, J., Stare, E., Talmola, P.: DVB-H: Digital broadcast services to handheld devices. Proceedings of the IEEE **94** (2006) 194–209

[100] : ETSI, Digital Video Broadcasting (DVB): DVB-H implementation guidelines (2009) TR 102 377 V1.3.1.

[101] : (ETSI, Digital Video Broadcasting (DVB): Framing Structure, Channel Coding and Modulation for Digital Terrestrial Television) EN 300 744 V1.6.1.

[102] A. Aksay, A., Bici, M.O., Bugdayci, D., Tikanmaki, A., Gotchev, A., Akar, G.B.: A Study on the Effect of MPE-FEC for 3D Video Broadcasting over DVB-H. In: MobiMedia'09. (2009)

[103] Vetro, A., Pandit, P., Kimata, H., Smolic, A., Wang, Y.: Joint Draft 8.0 on Multiview Video Coding. Joint Video Team, Doc. JVT-AB204 (2008)

[104] Services, A.A.M.: (Fatcaps: A free, linux-based open-source dvb-h ip-encapsulator)

[105] Oksanen, M., Tikanmaki, A., Gotchev, A., Defee, I.: Delivery of 3D Video over DVB-H: Building the Channel. In: NEM-Summit'08. (2008)

[106] Kumar, S., Xu, L., Mandal, M., Panchanathan, S.: Overview of error resiliency schemes in H. 264/AVC standard. Press, Journal of Visual Communications and Image Representations (2005)

[107] : (Joint video team of itu-t vceg and iso/iec mpeg. wd reference software for mvc (jmvc) v.5.0.5)

[108] Failli, E.: Digital land mobile radio. Final report of COST 207 (1989)

[109] Gao, X., Zhuo, L., Wang, S., Shen, L.: A H. 264 Based Joint Source Channel Coding Scheme over Wireless Channels. In: Proceedings of the 2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing-Volume 00, IEEE Computer Society (2008) 683–686

APPENDIX A

CODING IMPLEMENTATIONS

Several open source codecs are used in this thesis and modified for the improved performance.

A.1 Encoder Modifications

- Implementation of Multi-View Video Codec using H.264/AVC Monoscopic Video Encoder (JM 10.1 Reference Software)
- Implementation of Asymmetric Stereo Video Codec using H.264/AVC Monoscopic Video Encoder (JM 10.1 Reference Software)
- Integration of Slice Encoding Mode for MVC Extension of H.264 Video Encoder (JMVC 5.0.5 Reference Software)
- Integration of Asymmetric Encoding Mode for MVC Extension of H.264 Video Encoder (JMVC 5.0.5 Reference Software)

A.2 Decoder Modifications

- Implementation of Real-Time Multi-View Video Decoder using FFMPEG Decoder [compliant with JMVM 3.0.2 Reference Software]
- Integration of Slice Encoding Mode for MVC Extension of H.264 Video Encoder (JMVC 5.0.5 Reference Software)

CURRICULUM VITAE

Anıl Aksay was born in Ankara, Turkey, in 1978. He received the B.S. and M.S degree in Electrical and Electronics Engineering, from Middle East Technical University, Ankara, Turkey in 1999 and 2001, respectively. He worked as a researcher in 3DTV project, Network of Excellence funded by the European Commission 6th Framework Programme. He is currently working as a researcher in MOBILE3DTV project, funded by the European Commission 7th Framework Programme. His research interests include image, video and 3D coding, streaming, transmission.

His publications are as follows:

Book Chapters

1. A. Norkin, M. O. Bici, **A. Aksay**, C. Bilen, A. Gotchev, G. Bozdagi Akar, K. Egiazarian, and J.Astola, "Multiple description coding and its relevance to 3DTV" in Haldun M. Ozaktas and Levent Onural (Eds.), "Three-Dimensional Television: Capture, Transmission, and Display.", Springer, Heidelberg, 2007. In press.

Journal Papers (SCI, SCI-EXP)

1. **A. Aksay**, G. Bozdagi Akar, "Optimization of Encoding and Error Protection Parameters for 3D Video Broadcast over DVB-H", ACM Multimedia System Journal, Special Issue on Wireless Multimedia Transmission Technology and Application (submitted)

2. M. O. Bici, D. Bugdayci, A. M. Demirtas, **A. Aksay**, G. Bozdagi Akar, "Error Resilient 3D Video Transmission over DVB-H", IEEE Journal of Selected Topics in Signal Processing, Special Issue on Recent Advances in Video Processing for Consument Displays (submitted)

3. C. G. Gurler, **A. Aksay**, G. Bozdagi Akar, A. M. Tekalp, "Architectures for Multi-Threaded MVC-Compliant Multi-View Video Decoding and Benchmark Tests", Signal

Processing, Image Communication, Special Issue on Breakthrough Architectures for Image and Video Systems (accepted)

4. A. S. Tan, **A. Aksay**, G. Bozdagi Akar, E. Arikan, "Rate-Distortion Optimization for Stereoscopic Video Streaming with Unequal Error Protection," EURASIP Journal on Advances in Signal Processing, Special Issue 3DTV: Capture, Transmission and Display of 3D Video, vol. 2009, Article ID 632545, 15 pages, 2009. doi:10.1155/2009/632545 (Jan 2009).

5. **A. Aksay**, S. Pehlivan, E. Kurutepe, C. Bilen, T. Ozcelebi, G. Bozdagi Akar, R. Civanlar, M. Tekalp, "End-to-end Stereoscopic Video Streaming with Content-Adaptive Rate and Format Control", Signal Processing:Image Communication Special Issue on Three-Dimensional Video and Television, Vol 22/2 pp 157-168 (Feb 2007).

6. B. U. Toreyin, A. Enis Çetin, **A. Aksay**, M. B. Akhan, "Moving Object Detection in Wavelet Compressed Video", Signal Processing:Image Communication, EURASIP, Elsevier, vol. 20, pp. 255-264 (Mar 2005). Listed in the TOP25 articles within the journal: Jan.-Mar. 2005 (7th),Apr.-June 2005 (17th), July-Sept. 2005 (11th), and Oct.-Dec. 2005 (18th) and Jan.-Mar. 2006 (16th).

International Conference Papers

1. **A. Aksay**, M. O. Bici, D. Bugdayci, A. Tikanmaki, A. Gotchev, G. B. Akar, "A Study on the Effect of MPE-FEC for 3D Video Broadcasting over DVB-H," Mobimedia 2009, London, UK, Sept'09.

2. C. G. Gurler, **A. Aksay**, G. Bozdagi Akar, A. M. Tekalp, "Multi-Threaded Architectures and Benchmark Tests for Real-Time Multi-View Video Decoding", IEEE ICME 2009, Cancun, Mexico, July 2009.

3. **A. Aksay**, G. Bozdagi Akar, "Evaluation of Stereo Video Coding Schemes for Mobile Devices", 3DTV-CON 2009, Potsdam, Germany, May 2009.

4. M. O. Bici, **A. Aksay**, A. Tikanmaki, A. Gotchev, G. Bozdagi Akar, "Stereo Video Broadcasting Simulation for DVB-H", NEM-Summit'08, St Malo, France, Oct. 2008.

5. E. Kurutepe, **A. Aksay**, C. Bilen, C. G. Gurler, T. Sikora, G. Bozdagi Akar, A. M. Tekalp, "A Standards-Based, Flexible, End-to-End Multi-View Video Streaming Architecture", PV'07, Lausanne, Switzerland, Nov. 2007.

6. A. S. Tan, **A. Aksay**, C. Bilen, G. Bozdagi Akar, E.Arikan, "Rate-Distortion Optimized Layered Stereoscopic Video Streaming with Raptor Codes", PV'07, Lausanne, Switzerland, Nov. 2007.

7. C. Bilen, **A. Aksay**, G. Bozdagi Akar, "Two Novel Methods for Full Frame Loss Concealment in Stereo Video", PCS'07 , Lisboa, Portugal, Nov. 2007.

8. **A. Aksay**, A. Temizel, A. Enis Cetin, "Camera Tamper Detection Using Wavelet Analysis For Video Surveillance", IEEE AVSS'07, London, UK, Sept. 2007.

9. A. S. Tan, **A. Aksay**, C. Bilen, G. Bozdagi Akar, E.Arikan, "Error Resilient Layered Stereoscopic Video Streaming", 3DTV CON'07, Kos, Greece, May 2007.

10. C. Bilen, **A. Aksay**, G. Bozdagi Akar, "A Multi-View Video Codec based on H.264", IEEE ICIP 2006, Atlanta, GA, USA, Oct. 2006.

11. A. Norkin, **A. Aksay**, C. Bilen, G. Bozdagi Akar, A. Gotchev, J. Astola, "Schemes for multiple description coding of stereoscopic video", MRCS 2006, Istanbul, Turkey, Sept. 2006. Lecture Notes in Computer Science, vol. 4105, pp. 730-737, Springer-Verlag Heidelberg.

12. **A. Aksay**, C. Bilen, E. Kurutepe, T. Ozcelebi, G. Bozdagi Akar, M. R. Civanlar, A. M. Tekalp "Temporal and Spatial Scaling For Stereoscopic Video Compression", IEEE EUSIPCO 2006, Florence, Italy, Sept. 2006.

13. S. Pehlivan, **A. Aksay**, C. Bilen, G. Bozdagi Akar, R. Civanlar, "End-to-End Stereoscopic Video Streaming System", IEEE ICME 2006, Toronto, Canada, July 2006.

14. A. Boev, A. Gotchev, K. Egiazarian, **A. Aksay**, G. Bozdagi Akar, "Towards compound stereo-video quality metric: a specific encoder-based framework", IEEE SSIAI 2006, Denver, Colorado, USA, March 2006.

15. **A. Aksay**, M. Oguz Bici, G. Bozdagi Akar, "Evaluation of Disparity Map Characteristics For Stereo Image Coding", IEEE International Conference on Image Processing, ICIP-2005, Italy, Sept. 2005.

16. **A. Aksay**, C. Bilen, G. Bozdagi Akar, "Subjective Evaluation of Effects of Spectral and Spatial Redundancy Reduction on Stereo Images", 13th European Signal Processing Conference, EUSIPCO-2005, Turkey, Sept. 2005.

17. Ü.Ünal, **A. Aksay**, G. B. Akar, "An Implementation of a Wireless Streaming System", COST 276 Workshop, Ankara, Dec. 2004.

18. B. U. Toreyin, A. Enis Çetin, **A. Aksay**, M. B. Akhan, "Moving Region Detection in Compressed Video", 19th Int. Symposium on Computer and Information Sciences, Kemer-Antalya, Oct 2004. Lecture Notes in Computer Science, vol. 3280, pp. 381-390, Springer-Verlag Heidelberg.

19. **A. Aksay**, G. Bozdagi, M. B. Akhan, "Wavelet Based Image Sequence Compression," IEEE Balkan Conference on Signal Processing (BCSP'2000), Istanbul, Turkey, Jun 2000.

20. **A. Aksay**, G. Bozdagi, M. B. Akhan, A. Temizel, P. B. Duhamel, "Motion Wavelet Compression," IEE Colloqium Time-Scale and Time-Frequency Analysis and Applications, London, UK, Feb 2000.

21. **A. Aksay**, A. Temizel, M. Ozdal, G. Bozdagi, H. Palaz, "Real Time Motion Estimation Using TMS320C80," ICSPAT'99, Orlando, Nov. 1999.