Error Resilience and Concealment in Multiview Video over Wireless Networks

A thesis Submitted for the degree of

Doctor of Philosophy

by

Abdulkareem Bebeji Ibrahim

Supervised by Prof. Abdul H. Sadka

Electronic and Computer Engineering

School of Engineering and Design

Brunel University London

December 2014

ABSTRACT

Multiview video is capable of presenting a full and accurate depth perception of a scene. The concept of multiview video is becoming more useful especially in 3D display systems by enhancing the viewing of high resolution stereoscopic images from arbitrary viewpoints without the use of any special glasses. Like monoscopic video, the multiview video is faced with different challenges such as: reliable compression, storage and bandwidth due to the increased number of views as well as the high sensitivity to transmission errors. All these may lead to a detrimental effect on the reconstructed views. The work in this thesis investigates the problems and challenges of transmission losses in a multiview video bitstream over error prone wireless networks. Based on the network simulation results, the proposed technique is capable of addressing the problem of transmission losses. In practical wireless networks, transmission errors are inevitable and pose a serious challenge to the coded video data. The aim of this research effort is to examine the effect of these errors in a multiview video bitstream when transmitted over a lossy channel. Moreover, this research work aims to develop a novel scheme that can make the multiview coded videos more robust to transmission errors by minimizing the error effects and improving the perceptual quality. Multi-layer data partitioning as an error resilient technique is developed in JMVC 8.5 reference software in order to make the multiview video bitstream more robust during transmission. In addition to that, we propose a simple decoding scheme that can support the decoding of the multi-layer data partitioning bitstream over channels with high error rate. The proposed technique is benchmarked with the already existing H.264/AVC data partitioning technique. The work in this thesis also employs the use of group of pictures as a coding parameter to investigate and reduce the effects of transmission errors in multiview video transmitted over a very high error rate channel. The experiments are carried out with different error loss rates in order to evaluate the performance of these techniques in terms of perceptual quality when transmitted over a simulated erroneous channel. Errors are introduced using the Sirannon network simulator. The error performance of each technique is evaluated and analysed both objectively and subjectively after reconstruction. The results of the research investigation and simulation are presented and analysed in chapter six of the thesis.

Acknowledgements

First and foremost, I would like to give all praise to Almighty Allah (God) for blessing me with health, wisdom, and guidance for the successful completion of the research work. Special gratitude and appreciation goes to my supervisors for this achievement for their relentless support, stimulation and motivation. It was indeed a great honour and pleasure to work with the two well recognized experts in the field of video compression and multimedia communications in person of Professor A.H Sadka and Dr. Nikolaos Boulgouris. Professor A.H Sadka been my primary supervisor has contributed immensely in so many ways throughout the research period. It is worth mentioning to speak about his patience to research work, professional advice, suggestions and recommendations, constructive criticism and finally, I will never forget the free coffee drinks! He is truly a role model, and I can see the relationship between us going beyond where it started from. Dr Nikolas Boulgouris been the secondary supervisor has contributed through advice and comments at various stages of the research. I would like to thank the Petroleum Technology Development Fund (PTDF) of Nigeria for the financial commitments and support. I also wish to thank my employer, the Director General of NASRDA, Dr. S.O Muhammad for his acceptance to further my study abroad. Special thanks to Alh Tijjani Galadima of PTDF for his help and support especially with my administrative challenges regarding PTDF. A special thanks to both my father and mother in law for their relentless motivation and moral support. I would also like to thank my brothers and sisters, family friends and relatives and all CMCR colleagues for their support in one way or the other throughout the research period.

Dedication

I dedicate this thesis to my beloved parents Alh. Ibrahim Abdulkareem and Hajiya Binta Ibrahim Tukur for their wonderful love, care and prayers. Also, this thesis is dedicated to my loving wife Engr. Hadiza Sani Haliru and our two lovely kids Khadeeja Abdulkareem Bebeji and Abdullah Abdulkareem Bebeji. They mean everything to me, and I will forever love you all.

List of Abbreviations

2D	Two-dimensional
2DTV	Two-dimensional Television
3D	Three-dimensional
4k	4K Resolution
8k	8K Resolution
ACK	Positive Acknowledgement
ARQ	Automatic Repeat Request
ATM	Asynchronous Transfer Mode
ATTEST	Advanced Three-dimensional Television System
AVC	Advanced Video Coding
BMA	Block Matching Algorithm
CABAC	Context Adaptive Binary Arithmetic Coding
CAVLC	Context Adaptive Variable Length Coding
CIF	Common Intermediate Format
CMCR	Centre for Media Communication Research
CRC	Cyclic Redundancy Check
DCT	Discrete Cosine Transform
DIBR	Depth Image Based Rendering
DP	Data Partitioning
DSP	Digital Signal Processing
DTV	Digital Television
DV	Disparity Vector
DVB	Digital Video Broadcasting
DVD	Digital Video Disc
DVB-H	Digital Video Broadcasting - Handheld
DVB-T	Digital Video Broadcasting – Terrestrial
EC	Error Concealment
ER	Error Resilient

FEC	Forward Error Correction
FMO	Flexible Macroblock Ordering
FVT	Free Viewpoint Television
FVV	Free Viewpoint Video
GOB	Group of Block
GOP	Group of Pictures
GOV	Group of Views
HD	High definition
HDTV	High Definition Television
HEVC	High Efficiency Video Coding
HTTP	Hypertext Transfer Protocol
HVS	Human Visual System
IDR	Instantaneous Decoder Refresh
IEEE	Institute of Electrical and Electronics Engineers
IVF	Interview Flag
IP	Internet Protocol
IPTV	Internet Protocol Television
ISO	International Organization for Standardization
IEC	International Electro technical Commission
ISSN	Integrated Special Services Network
ITU	International Telecommunication Union
ITU-R	International Telecommunication Union-Radio communication
ITU-T	International Telecommunication Union-Telecommunication
JMVC	Joint Model for Multiview Video Coding
JPEG	Joint Photographic Experts Group
JSVM	Joint Scalable Video Model
Kbps	Kilobits Per Second
LCD	Liquid Crystal Display
LT	Luby Transform
MB	Macroblock
MCP	Motion Compensated Prediction

MBA_MAP	Macroblock Allocation Map
MDC	Multiple Description Coding
MERL	Mitsubishi Electric Research Laboratories
ML	Multi-Layer
MOS	Mean Opinion Score
MPEG	Moving Picture Experts Group
MPEG-2 TS	MPEG-2 Transport System
MTU	Maximum Transmission Unit
MV	Motion vector
MVC	Multiview Video Coding
MVV	Multiview Video
NACK	Negative Acknowledgement
NAL	Network Abstraction Layer
NALU	Network Abstraction Layer Unit
NASRDA	National Space Research and Development Agency
OTT	over the top
P2P	Peer-to-peer
PLR	Packet Loss Rate
PPS	Picture Parameter Set
PSNR	Peak Signal to Noise Ratio
PTDF	Petroleum Technology Development Fund
QOS	Quality of Service
QCIF	Quarter Common Intermediate Format
QOE	Quality of Experience
QP	Quantization Parameter
RGB	Red Green Blue
RTP	Real Time Transport Protocol
SAD	Sum of Absolute Difference
SD	Standard Definition
SDTV	Standard Definition Television

SEI	Supplemental Enhancement Information
SNR	Signal to Noise Ratio
SPS	Sequence Parameter Set
SSD	Sum of Square Difference
SQCIF	Sub-Quarter Common Intermediate Format
SVC	Scalable Video Coding
TCP	Transport Control Protocol
TR	Temporal Reference
UDP	User Datagram Protocol
Ultra-HD	Ultra High definition
UEP	Unequal Error Protection
UMTS	Universal Mobile Telecommunications Systems
VCEG	Video Coding Expert Group
VCL	Video Coding Layer
VLC	Variable Length Coding
VoD	Video on Demand

List of Symbols

Symbol	**Meaning**
$\mathbf{D_0}$	Central Distortion
$\mathbf{D_1}$	Distortion for Channel 1
$\mathbf{D_2}$	Distortion for Channel 2
$\mathbf{R_1}$	Bitrate for Description 1
$\mathbf{R_2}$	Bitrate for Description 2
$\Delta\mathbf{R}$	Extra Overhead Bitrate
J	Lagrangian Cost Function
D	Sum of Absolute Difference
λ	Lagrange Parameter
R	Coding Rate
C	Complexity Mode
d	Distortion error of luma block
$\mathbf{Y_{ref}}$	Boundary luma pixel value for the MC MB
$\mathbf{MV_{dir}}$	Motion Vector of the MC MB
$\mathbf{Y_{rec}}$	Boundary luma Pixel for reconstructed Frame
N	Number of average pixels
G	Good State
B	Bad State
$\mathbf{P_{00}}$	Probability of bad to bad
$\mathbf{P_{01}}$	Probability of bad to good
$\mathbf{P_{10}}$	Probability of good to bad
$\mathbf{P_{11}}$	Probability of good to good
$\mathbf{A_0}$	Header information of frame 0 view 0 partition
$\mathbf{A_1}$	Header & motion information of frame 1 view 1 partition
$\mathbf{A_2}$	Header & motion information of frame 2 view 2 partition
$\mathbf{B_0}$	Intra coded residual information of frame 0
$\mathbf{B_1}$	Intra coded residual information of frame 1
$\mathbf{B_2}$	Intra coded residual information of frame 2
$\mathbf{C_0}$	Empty partition
$\mathbf{C_1}$	Inter coded residual information of frame 1

$\mathbf{C_2}$	Intra coded residual information of frame 2
$\mathbf{E_{x,\,y}}$	Residual error
$\mathbf{I_{x,\,y}}$	Pixel value
$\mathbf{P_{x,\,y}}$	Predicted value
$\mathbf{x,\,y}$	coordinates of the variables

Table of Contents

List of Figures

List of Tables

1. Chapter One: Introduction

1.2. The Context

Multi- view video transmission is a fast growing multimedia technology that is active both in the industry and research community. The content delivery system for multiview video is capable of streaming multiple simultaneous views of the same scene at the same time instant. The scene could span depending upon the number of cameras allowing the end user to see a larger view of the scene. This system requires a setup of cameras, encoder, decoder, streaming video server and a high speed data processing mechanism. Such services have been in the past constrained by limited resources such as computational complexity and network capacity. A contributing factor to widespread use of these services is the user's quality of experience (QoE), acceptance and awareness. This concept is a major technological revolution in terms of display and many other applications. The demand for multiview video is increasing in the area of multimedia technology. Multiview video technology can be used for coverage of sports events, broadcasting and in medical field and so on.

1.3. Problem Statement

In today's age, the growing demand for 2D video content and 3D services over the internet has made experts predict that by 2018; more than two-thirds of the world's mobile broadband traffic will be video content [1]. The main challenge in 3D multi view video communication over wireless networks is to present an acceptable quality of Experience (QoE) to end users. However, wireless channel is still a challenging problem due to the limitation in bandwidth and the presence of channel errors. These problems briefly described which include bandwidth variation and transmission errors. Bandwidth limitation is one of the most important factors in multimedia communications generally. Video streams are transmitted over networks with time varying conditions and resource limitations [2]. However, unreliability, bandwidth fluctuations and high bit error rates of wireless channels can cause severe degradation to video quality [3]. Packet loss and transmission error pose a serious challenge to the transmission of compressed video across networks resulting in packet losses due to congestion; thus the receiver may not be able to receive all of the compressed video data because of the losses, and consequently, the video quality will deteriorate [4].

1.4. Aims and Objectives

The main aim of the research is to investigate and evaluate through experiments and simulations the effects of transmission losses in multiview video bitstream when transmitted over a simulated error prone network and, also to develop a model that will efficiently improve the perceptual quality of the reconstructed multiview video. Specifically, the research objectives are stated as follows:

- To identify and study with understanding, all the error resilience techniques in both 2D and 3D video communication.
- To development a network model that will verify and validate the research work.
- Design novel methodologies and algorithms that will suitably improve the effects of channel errors for 3D multiview video coding.
- To have an in depth knowledge of the state of the art developments in the field.

1.5. Motivation

The application for the delivery of multimedia content such as video streaming, video calls, and IPTV over the internet is rapidly growing and becoming very common to the general public for both fixed and mobile services. However, in wireless video communication, the two major problems when transmitting multimedia content are bandwidth limitations and transmission errors. Bandwidth constraint has been dealt with in the last two decades by a number of coding algorithms. In particular, the state of the art video compression standard H.264/AVC (Advanced Video Coding) and its newly released extension High Efficiency Video Coding (HEVC) can provide better compression and quality. The H.264/AVC video coding standard adopts variable length codes (VLCs) as entropy codes for the achievement of high coding efficiency. The concept and design of VLC is the main root cause of error propagation due to its sensitivity to channel errors. A single bit error can render the whole bitstream undecodable and useless. Recently, 3D video representation and communication is also evolving and gaining research interest. The transmission of 3D video content over error prone channels pose more serious challenge because of the increased coding dependencies that makes it highly and more vulnerable to transmission errors. When addressing the problem of transmission error for a particular compressed sequence, the use of an effective error control strategy is necessary and important, this could be either at the source or channel level.

There are a number of error control techniques at the source level commonly known as error resilient techniques that are available for 2D video in H.264/AVC, which can be extended and utilised for 3D video. Data partitioning in particular is one error resilient standard that has not been readily exploited in 3D multiview video coding based on research finding and survey.

1.6. Research approach

The design and development of error resilience techniques for 3D multiview video coding aims at minimizing the effects of channel error in 3D video communication over a wireless network. The following approaches were considered.

1 The coded multiview bitstream is parsed and partitioned into three different partitions by the H264/AVC algorithm that is implemented into the JMVC reference software.

2 The H.264/AVC data partitioning algorithm is modified to be able to create another layer of partitioning within the multiview bitstream for a higher level of error robustness in a networked environment.

3 Valid error pattern is generated into the two different data partitioning algorithms by the use of a network simulator for evaluation purposes.

4 The corrupted multiview bitstream of the two algorithms is decoded with the modified decoder that is tolerant to errors and is capable of decoding the partitioned multiview bitstreams.

5 The performance of the two techniques is evaluated in terms of different error rates both objectively and subjectively.

1.7. Thesis main achievements

The achieved objectives are outlined below:

1. Design and development of the Multi-layer data partitioning technique for Multiview Video Coding in the JMVC reference software.
2. Implementation of H.264/AVC data partitioning technique in the JMVC 8.5 reference software.
3. Modification of the frame copy error concealment technique in the multiview video JMVC codec.
4. Decoder optimization for high performance handling of the multi-layer MVC bitstream at high error rates.

1.8. Thesis Outline

The thesis is organized into seven chapters, Chapter one describes the thesis layout by introducing the research topic, problems and challenges in the research field, aims and objectives of the research work, research motivation, methodology and the research contributions.

Chapter 2 introduces the state of the art video coding standard and the concept behind the H.264/AVC and 2D video coding technology. In addition, the chapter also presents and describes in details the multiview video coding principle, which is by default an extension to H.264/AVC video coding. The just released High Efficiency Video Coding (HEVC) is also presented in this chapter. This chapter addresses and focuses on the aspects that are directly relevant to the research work in the thesis.

Chapter 3 provides an overview of the 3D video communication system from content capturing at the source to display for the end users. Different components of the 3D video communication such as 3D video compression, 3D format and representation, 3D-HEVC and 3D display systems have been described. Current development and challenges related to each component are reported in this chapter as well as the future challenges that may affect 3D video communication system at large.

The 3D video quality assessment is also discussed in this chapter it includes the objective and subjective approach. Recent and standardized methodology as approved by the video coding expert group is reported and the challenges affecting 3D quality measurement in general.

Chapter 4 discusses the challenges in video transmission and a review of the standard error resilient techniques in H264/AVC is presented. Some recent developments by extension in ER and EC into MVC are also included and discussed. The chapter also briefly discussed Error control technique and also gives a review of error concealment and MVC error resilient decoder.

Chapter 5 describes in detail the experimental procedures and network simulation test bed. Key coding parameters such as Group of Pictures (GOP) and Quantization Parameter (QP) and their effects in the experimental study are analysed. Optimal and Appropriate settings for each of the coding parameter used is also explained. The chapter introduces and discusses all the tools and software employed in the course of the research work and experiment. The conditions necessary for the design of error resilience and concealment algorithms in multiview video coding (MVC) are highlighted. The results of some experimental work involving a comparative analysis between 2D and 3D compression are also reported.

Chapter 6 give details of the implementation process of H.264/AVC data partitioning technique in the JMVC 8.5 reference software. Following examination and evaluation of the H.264 DP technique, the main contribution of this chapter is the development of multi-layer data partitioning technique in MVC. The two techniques are crossed examined and evaluated in terms of error robustness against different channel losses and the perceptual quality of the reconstructed views. Furthermore, the bitrate consumption and analysis of the two algorithms are compared and reported based on an experimental study. The chapter further introduces a simple decoding scheme for erroneous MVC bitstream. The performance of the scheme is analysed based on the reconstructed views in terms of perceptual quality and bit rate for different GOP. Several simulations are carried out with the different multiview video test sequences. Both objective and subjective results are presented for various scenarios of the bitrate performance when no data partitioning is used and when both H.264 and multi-layer data partitioning are used.

Specifically, this chapter also considers two worst case scenarios of transmitting or streaming multiview video over a simulated channel with 20% error rate. Subjective evaluation of the MVV is carried out on each view in order to investigate in detail the effects of channel losses across views and how GOP size can be used to mitigate this in the reconstructed views.

A detailed and comparative analysis is given on the effect of varying GOP size for different error rates and suitable and optimum GOP size is recommended. The recommendation is based on our experimental results obtained which can be used for streaming or transmitting multiview video over a high loss rate channel with a bandwidth constraint.

Chapter 7 provides some conclusions based on the research study and experimental work undertaken. The chapter also discusses some recommendations for improvement as future work. Some research work and study beyond the scope of this thesis are presented in the appendix section.

1.9. Conclusions

This chapter discusses in brief the research scope and the thesis structure in general in various sections. The chapter begins with research introduction, challenges, aims and objectives, research main achievements and list of research publications. Lastly, the layout of the thesis is presented.

2. Chapter Two: H.264 Video Coding System

2.1. Introduction

Rapid development in digital communication and video coding systems has transformed multimedia communications in almost every aspect of life. This includes the integration of different kinds of applications and services into many and various platform devices for the delivery of multimedia services almost anywhere and anytime. The high demand multimedia services in today's age has pushed the boundary of video coding for codecs with better flexibility and higher compression gain to make them more applicable to different services and network conditions. In order to meet the industry requirement of standardizing existing video techniques, video coding standards were developed by two international organizations, ITU-T and ISO/IEC. The family of ISO/IEC MPEG standards includes MPEG-1, MPEG-2, MPEG-4, and MPEG-4 part 10 (AVC). ITUT-T H.264x series standards consist of H.261, H.263, and H.264. The evolution of video coding standards reflects the technological progress (Fig. 2.1) toward improving the coding efficiency of video compression technologies [5]. For instance, the current H.264/AVC standard jointly developed by ITU-T video coding experts group (VCEG) and ISO/IEC Moving Picture Experts Group (MPEG) [6].

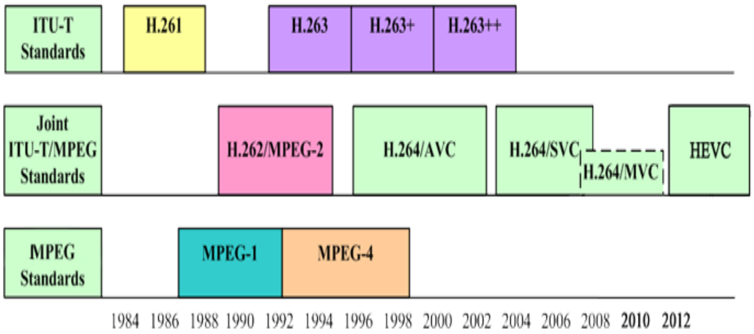


Figure 2.1: Advancement in video coding technology [7]

The development of H.264/AVC has demonstrated a better performance of up to 50% in terms of coding efficiency over a wide range of bit rates and different video resolutions when compared to other previous video coding standards [8]. The H.264/AVC video encoder is equipped with a unique transmission tool that can facilitate the transmission of video coded data across various network channels [6]. This feature of the codec has made it more useful for a wide range of applications such as Video-on-Demand, digital media storage, TV broadcasting, High Definition TV (HDTV), mobile TV, Multimedia streaming and conversational applications and systems. For these purposes, relevant industries are employing H.264/AVC coding system in commercial applications [9][10]. The H.264/AVC standard supports several error resilience techniques in order to combat transmission or channel errors which cause severe effects on the perceptual quality of the reconstructed video sequences. The error control techniques in H.264 video coding will be discussed in detail later in chapter 4. The rest of this chapter provides an introduction to the video coding concept of H.264/AVC as a standard.

2.2. Principles of H.264/AVC

Generally, visual information and video data require basically two major factors for considerable utilization, which are a huge amount of bandwidth and storage. A video sequence can be reconstructed through the use of lossless compression algorithms which can assure perfect reconstruction of the original video data. This compression algorithm can typically reduce data rate by a factor of two which is not sufficient for mobile wireless video communication [11]. On the other hand, lossy compression can achieve higher compression gain by reducing the correlation within the video sequence without affecting the perceptual quality of video significantly. Two types of redundancies are exploited in H.264/AVC coding system: First is the Psycho-visual redundancy whereby the strategy here is to discard video information that is apparent to the human visual system (HVS). The second type is the Spatio-temporal redundancy technique that exploits the similarities between neighbouring pixels within a picture or across different pictures.

2.3. Colour Space Model

An accurate representation of colour in video frames requires at least three numbers per pixel position. Video colour space in H.264/AVC separates a colour representation into three different components known as YCbCr. The Y component is called luminance, and it represents brightness. The two other components, Cb and Cr are known as the chrominance blue and chrominance red respectively [12]. Video sequences in YCbCr format is important and more preferred over RGB colour space model, this is true because the chrominance components Cb and Cr can be represented as a lower resolution video than the luminance component as the case may be. The main reason behind this concept is because the HVS has a lower sensitivity to colours than luminance [11] and this can allow a good colour image quality to be represented with quite a small amount of chrominance values. In the YCbCr 4:2:0 sampling, each of the chrominance components Cb and Cr is considered to have half the horizontal and vertical resolution of the luminance component Y. This colour representation is well utilized in consumer applications since the sampling format reduces a considerable amount of data rate and storage space required by a factor of two when compared to RGB or Y Cb Cr 4:4:4 sampling format with less or minimum reduction in the video quality.

2.4. Video Formats

The H.264/AVC compression algorithm is capable of compressing a wide variety of video frame formats. In fact, it is a common practice in H.264/AVC to capture or convert to one of the available set of intermediate formats before compression and transmission [13]. The common intermediate Format (CIF) is the basic frame format, in which each frame has a resolution of 352 x 288 pixels. 4CIF is appropriate for standard definition TV (SDTV), CIF and QCIF are popular for video conferencing applications, while QCIF and SQCIF are more appropriate for mobile multimedia applications where display resolution and bitrate are limited.

2.5. The Technical Overview of H.264/AVC

The MPEG-4/AVC H.264 video coding is currently one of the major video coding standards which is designed to be simple with high compression performance and network friendly nature that will allow video bitstream to adapt with the different types of network [14]. As part of the standardization objective, the video coding tool consists of the Video Coding Layer (VCL), which is responsible for the compression of the source video into coded information, and the Network Abstraction Layer (NAL). The NAL is designed to format the coded video data by adding header information in a way that is suitable for delivery over an IP network (usually by transport layers) or storage media [15].

2.5.1. Video Coding Layer

The Video Coding Layer (VCL) of H.264/AVC is similar in many ways to the previous standards like MPEG-1, MPEG-2, MPEG-4, H.261 and H.263. VCL is designed to be network independent which consists of the core compression engine with different syntactical levels known as the block, MB, and the slice [12]. In practice, a standard compliant H.264/AVC codec should include functional components as depicted in Fig. 2.2, which adopts the conventional hybrid block based (temporal and spatial prediction) video coding. It is important to note that the VCL contains several coding tools that enhance error resilience in the compressed video stream and will be discussed in chapter four of this thesis. Usually, a sample video frame F_n is partitioned into different regions of MBs each representing a 16x16 sample regions of non-overlapping MBs. For each of the MBs of the current frame, a prediction P is formed either in Intra or Inter mode.

Intra coded MBs are predicted from spatial neighbouring samples of MBs that were previously encoded. The inter coded MBs are predicted by way of motion compensation from previously decoded MBs from a reference frame(s) F_{n-1} which can be selected either from the past or future frames already decoded. The residual block R_n which is the difference between the original and predicted sample MBs is transformed into a domain of de-correlated video data suitable for compression. The resulting transform coefficients are approximated using scalar quantization to generate X quantized coefficients. These quantized coefficients are then reordered, entropy encoded and transmitted along with entropy coded prediction and control information after being encapsulated in the Network Abstraction Layer (NAL) for storage or suitable transmission across a network [9].

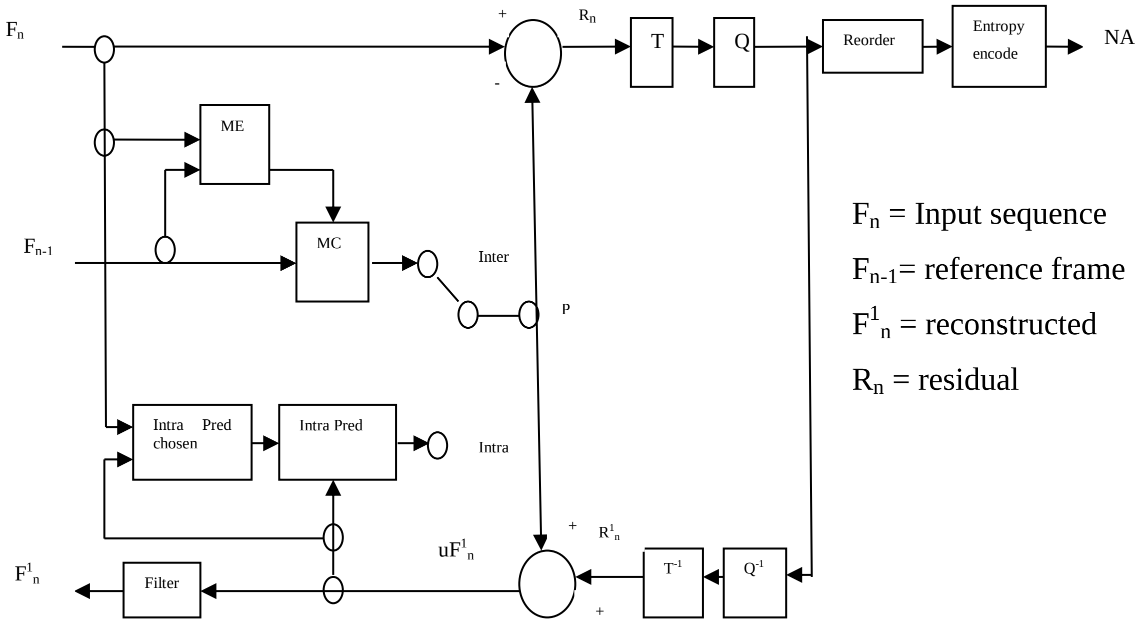


Figure 2.2: H.264/AVC Encoding Process

Furthermore, H.264/AVC encoder consists of an internal decoder platform for the generation of exact prediction of subsequent encoded frames at the decoder. This concept ensures encoder – decoder synchronization. The coefficients X are re- scaled and inverse transformed to generate the residual block R^1_n which is added to the prediction signal uF^1_n to reconstruct the block. Furthermore, de-block filtering is used to remove and smooth out edge discontinuities. This reconstructed block is buffered for the prediction of subsequent encoded pictures. H.264/AVC employs the basic coding mechanism implemented by previous video coding standards. For the purpose of performance improvement, a number of enhancements and refinements were introduced in the H.264/AVC codec. These enhancements include:

2.5.1.1. Intra Frame Prediction

Two types of Intra prediction modes are supported for luminance prediction in H.264/AVC; namely The Intra 4x4 and intra 16x16 predictions. The former is suitable for coding regions with complicated texture information with each of the 4x4 luminance block separately predicted. The latter is suitable for coding of smooth regions with less motion and prediction is performed on the entire 16x16 luminance block. For chrominance samples, Intra prediction is carried out in a similar fashion on the Intra 16x16 and is always performed on the 8x8 blocks.

The H.264/AVC codec has nine directional Intra prediction modes for each 4x4 block, four directional intra prediction modes for the 16x16 block and four directional prediction modes for the chrominance prediction. The encoder is left with the decision to select the prediction mode for each block of MB that minimizes the difference between the prediction and the original block.

2.5.1.2. Inter Frame Prediction

This is also known as motion compensated prediction. Prediction is achieved from the image signal of the already transmitted reference images. Each MB can be divided into smaller partitions of luminance block of 16x16, 16x8, 8x16, and 8x8 samples [16].

In general, larger block size is more suitable for homogenous regions and smaller block size is more suitable for encoding of regions with complex texture information. This technique is known as Variable Block Size Motion Compensated Prediction. H.264/AVC adopts what is known as quarter pixel accuracy to represent motion vectors and another enhancement in the video codec is the ability of the H.264/AVC video codec to reference several preceding images to achieve motion compensation prediction. This technique is known as motion compensated prediction with multiple reference frames and this technique exploits long term dependencies in a video sequence. H.264/MPEG-4 AVC also employs the partitioning of pictures into further subdivision known as slice, which can be further subdivided into macroblocks. In this concept of slicing, each slice is independent of the other in the picture [17]. Each sample of the macroblock is either spatially or temporally predicted, and the residual signal generated is presented for transform coding. The H.264/AVC standard supports slice coding that enables the coding of macroblocks at slice level.

- I-Slice uses intra frame coding to spatially predict each macroblock from other surrounding macroblocks within the same slice.
- P-Slice supports both intra and inter-frame predictive coding by using one prediction signal for each predicted region.
- B-slice supports intra frame coding, inter frame coding, and also inter frame bi-predictive coding by using two prediction signals that are combined with a weighted average to form the region prediction [17].

For I-slices, the standard provides numerous directional spatial intra frame prediction modes, in which the prediction signal is generated from the decoded intra macroblocks with a slice. For the luminance component, the intra frame prediction can be applied to individual 4x4 or 8x8 luminance blocks or to the full 16x16 luminance array for the macroblocks.

For P and B slices, the standard also permits variable block size motion compensated prediction with multiple reference picture. The macroblock type signals the partitioning of a macroblock into blocks of 16x16, 16x8, 8x16, or 8x8 luminance samples. Once a macroblock type specifies partitioning into four 8x8 blocks, each of these sub-macroblocks can be further split into 8x4, 4x8, or 4x4 blocks [17].

2.5.1.3. Transform and Quantization

H.264/AVC specifies transform and quantization processes that are designed to offer high efficient coding of the video information, to remove mismatch or drift between the encoder and decoder and to enhance low complexity implementations [13].

All operations involved in the transform process are achieved through integer arithmetic only requiring addition and shifts. The programmable stream architecture provides a powerful mechanism to achieve high performance in media and signal processing [18].

Quantization is adopted in the H.264 standard to precisely represent sample value or group of sample values in order to reduce the amount of video data that is needed to encode the representation. This concept is analogous to rounding off figures. However, the rounding precision is controlled by a step size that specifies the smallest representable value increment [19]. Transform and quantization process are both computationally intensive components in the design of the H.264 video coding tool. The standard adopts block based motion prediction, so the residual difference between the current frame and the predicted frame is organized into a block of video data, and each block is independent of others , exposing a great deal of data parallelism. Technically, 4x4 or 8x8 adaptive transform block sizes adopted are an integer orthogonal computation that allows for bit exact implementation for all H.264 compliant codecs [13]. The Smaller block size leads to a significant reduction in ringing artefacts and also has the additional advantage of removing the need for multiplication [20].

2.5.1.4. Entropy Coding

Entropy coding in H.264/AVC standard is generally based on the fixed tables of variable-length-codes (VLCs) designed to focus on the residual data coding in default mode [21]. The residual coding is achieved by first mapping a block of transform coefficients into one-dimensional list using a pre-defined scanning pattern. The list of transform coefficient levels is then coded by using the combination of run-length and variable length coding [22]. Two types of entropy coding methods are supported in H.264/AVC standard. They are context adaptive variable length coding (CAVLC) and context adaptive binary arithmetic coding (CABAC). The context adaptive variable length coding (CAVLC) is a part of entropy coding that is used to encode the residual blocks in zigzag order. They are designed to take advantage of the many characteristics of quantized 4x4 blocks such as using run length coding technique to compactly represent strings of zeros and so on [23]. The CABAC is an extension of the binary arithmetic coding (BAC). It is an arithmetic coding system that is used to encode/decode syntax elements in order to achieve higher compression performance through adaptive probability of arithmetic estimates based on local statistics [24]. CABAC has been adopted as a normative part of the H.264/AVC standard to provide an alternative method of entropy coding. Compared to CABAC, CAVLC offers a reduced cost of computation and implementation at the expense of lower compression gain. In media applications, TV signals in standard or high definition, CABAC offers higher bit rate savings of 10-20% compared to CAVLC at the same objective video quality [22].

2.6. Profiles and Levels

The H.264/AVC addresses technical issues in a wide range of applications such as bit rates, sequence/frame resolutions, perceptual quality and network services [8]. However, every application has its own different requirements. In an effort to maximize interoperability with less complexity, H.264/AVC as part of its specification defines profiles and levels.

A profile includes a subset of the entire functions in the bitstream while a level as specified in the recommendation imposes constraints on the values of the syntax elements in bitstream such as bit rate, storage and resolution.

Three major profiles have been defined in H.264/AVC video coding.

- Baseline Profile: This is considered to be the simplest profile that targets applications with low delay and computational complexity. This profile is suitable for videoconferencing and mobile applications.
- Main Profile: This profile is inclusive of the baseline tools and was aimed for broadcast and storage applications. It provides the best quality at the expense of higher complexity (mainly due to B-slices and CABAC) and delay.
- Extended Profile: This profile supports all the tools in both the baseline and the main profiles with the exception of CABAC. It is also suitable for video streaming applications and further comprises additional error resilience tools.

Each profile is designed to target a specific class of applications for optimum performance and to define what feature sets the encoder may utilize and to limit the decoder implementation complexity [19].

2.6.1. Network Abstraction Layer

An important feature in the recent H.264/AVC codec is its ability to insert video related information within the network abstraction layer units (NALUs). This concept enhances the transmission of H.264/AVC bitstream over a variety of network channels [6]. This is also applicable for a wide range of media applications such as TV broadcasting, mobile TV, video-on-demand, digital media storage, high definition TV, multimedia streaming and conversational applications. The Network Abstraction Layer efficiently represents the coded video data into an organized format for delivery across the network. Technically, the coded video data is encapsulated into NAL units as shown in Fig. 2.3, also referred to as packets. Every NAL unit consists of 1 byte header information and integer number of bytes representing the video data in the payload. In H.264/AVC, the NAL unit header specifies information about the NAL unit type and the level of importance of the NAL unit payload while decoding [25]. The ITU-T specifies a generic format for use in both packet oriented and byte oriented bitstream transport system. These two formats are identical except that the latter is preceded by a start code prefix [14]. Fig.2.4 illustrates how the video codec communicates through NAL unit which holds video related information that can further be encapsulated into several other transport formats such as MPEG-2 TS, Real Time Transport Protocol (RTP), MPEG-4 file format and H.32X conversion services [12].

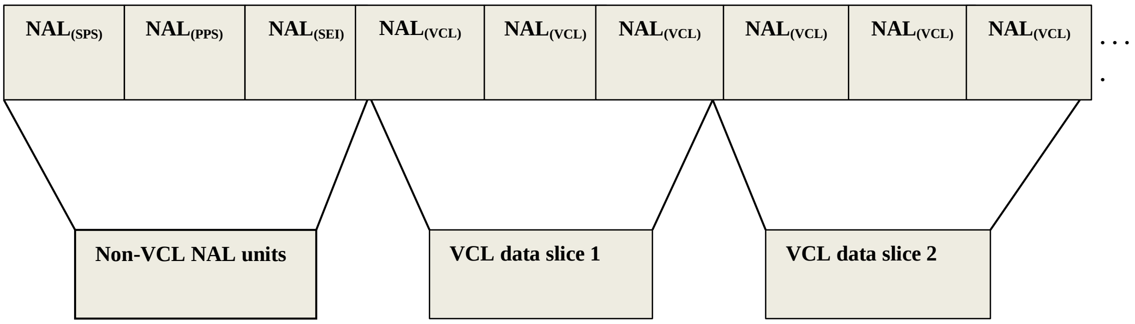


Figure 2.3: Packet oriented bitstream format

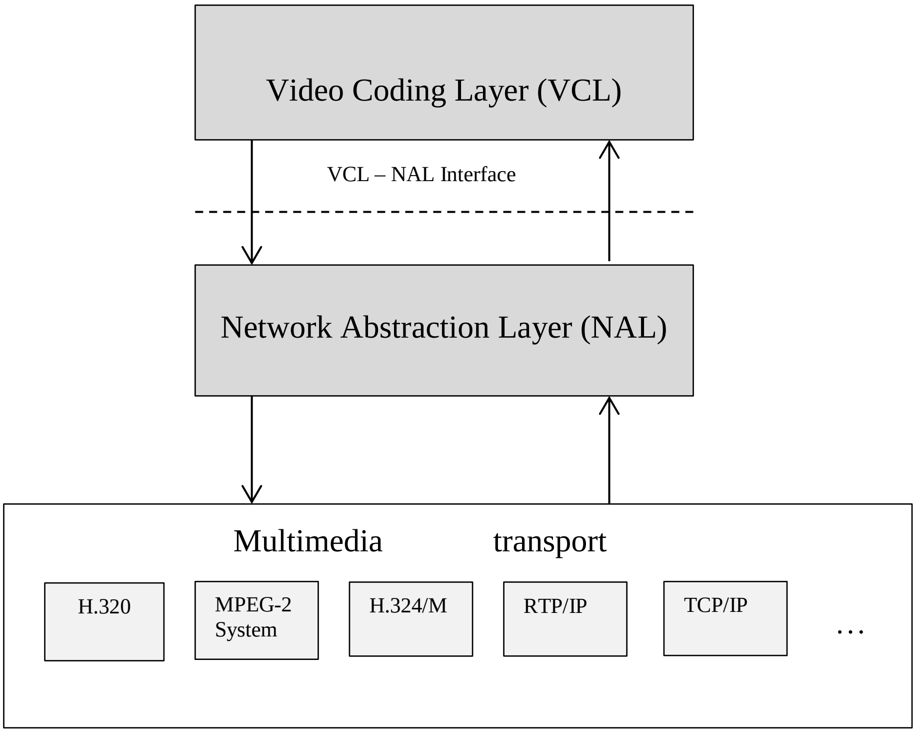


Figure 2.4: H.264/AVC standards in transport environment

NAL units are further classified into VCL NAL units and non-VCL NAL units. The VCL NAL units contain coded information of the picture while the non-VCL NAL unit contains related additional information such as parameter sets and supplemental enhancement information (SEI) messages [17]. The concept of VCL NAL unit types and non-VCL NAL unit types are defined and documented in [26] and [27]. An important concept of parameter sets is that it decouples the transmission of frequently changing information in the video coded data in order to avoid the transmission of previous information within the VCL NAL units. This concept provides a more efficient transmission. There are two types of parameter sets: The Sequence parameter sets (SPS) is applicable across a series of consecutive coded video frames and the Picture parameter set (PPS) carries information that is related to a single or more pictures in a coded video data. As part of the flexible nature of H.264/AVC video coding, it allows for these related parameter sets to be transmitted well ahead of the VCL NAL units [28]. However, as a measure to achieve robustness against loss, multiple NAL units of these parameter sets can be transmitted or alternatively, the parameter sets could be sent in a more reliable transport mechanism such as in a feedback system.

2.7. MVC Extension of H.264/AVC

Emerging 3D techniques such as free viewpoint and 3DTV are recent and new types of visual media applications that expand the user’s scope of experience beyond what is experienced with the 2D video [29][30]. The 3DTV and FVV offers a depth impression of the observed scenery that can allow an interactive selection of viewpoint and direction within a defined viewing range [31]. Multiview video coding is a common element between these systems that uses multiple views from slightly different angles in order to capture the same scene at the same time. Because of the increased number of cameras capturing the same scene, large amount of video data is generated for either storage or transmission. However, this challenge necessitates the different compression techniques for the multiview video that can encode the video sequence without losing the visual quality significantly. The multiview video coding [32] is an extension of the Advanced Video Coding (AVC) recent standard [25] that provides efficient coding of multiview video. The MVC system architecture can be demonstrated in Fig. 2.5. A number of temporally synchronized video sequences are encoded by an MVC encoder to produce a single bitstream for transmission or storage.

The decoder receives the MVC bitstream, and decodes the bitstream into N components/views for viewing. Each view of the MVC bitstream is identified by an arbitrary view ID number and is not specific to any ordering between views [33].

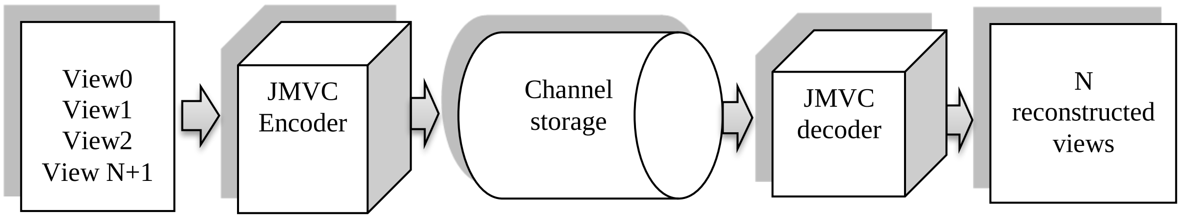


Figure 2.5: MVC system architecture

MVC provides superior network robustness and compression performance for delivering 3D video content by taking advantage of the interview dependencies of the different views. Also, the backward compatibility with the H.264/AVC codecs makes it widely interoperable in environments having both 2D and 3D capable devices [34]. Like any other video coding standard, the key requirement is to achieve high compression gain. The main goal of MVC is to offer a significant increase in the compression efficiency as compared to encoding of each view individually. Compression efficiency in this context is a measurement between bitrate and quality of video. Also, as a general requirement and design consideration for video coding standards, it is important to minimize the use of the following resources which include memory, processing power, and error robustness [19]. Some specific requirements that are attributed to MVC include random access which is a feature that ensures any picture can be accessed at any time. This can be achieved by inserting an intra-coded picture that does not need to be predicted from other pictures. Also, view scalability is a requirement for MVC in order to access a portion of the bitstream to produce a limited number of N original views [31]. Another required feature is backward compatibility which can allow one view from the MVC bitstream to be in conformance to a standard H.264/AVC codec.

The video quality consistency amongst the views is addressed. It should be possible to adjust the encoding process to achieve an approximate constant quality across all views. Like the H.264/AVC codec, relevant camera parameters should be transmitted with the bitstream in order to enhance interpolation between views at the decoder.

2.7.1. Temporal and Interview correlation

In video coding concept, the main difference between monoscopic video coding and multiview view video coding is the additional views that are capturing the same scene. While the coding efficiency of any standard advanced video codec depends on the quality of the prediction signal, MVC on the other hand can achieve coding gain through efficient interview prediction. The MVV sequences generate a huge volume of data that will require large storage and a high bandwidth for transmission. This problem necessitates the development of various compression schemes for MVC that are discussed in detail in chapter three. MVC achieves higher coding efficiency by utilizing the spatial redundancy between neighbouring views in addition to the temporal redundancy between successive frames in the MVV bitstream. By default, the MVC compression algorithm searches the reference frames of successive pictures within a search window both in the current view and from other neighbouring views for prediction. Based on this principle, the advanced video coding extension reference software was developed, which is known to be the Joint Multiview Video Model (JMVM) and standardized by the Joint Video Team (JVT). The JMVM standard utilizes hierarchical B-frames across all views within the prediction structure to achieve higher coding gain at the expense of delay. The coding technique of the standard also makes use of variable size prediction scheme of H.264 to exploit the redundancies within subsequent frames both in time and space domain. The prediction scheme in MVC consists of conventional variable size motion estimation (ME) and the added disparity estimation (DE) technique. These two important schemes are computationally intensive in the MVC system [35]. There are a number of proposed fast algorithms for MVC in order to reduce the complexity of prediction in both temporal and view direction [36] [37]. Most of these algorithms utilizes camera geometry and by finding a way of reducing the search range of both the ME and DE in the MVV sequences. In addition to that, some of the algorithms achieve faster ME and DE processing time by reducing the number of reference frames used for prediction at the expense of improved coding efficiency.

2.7.2. Motion and Disparity estimation

The MVC achieves high coding efficiency by employing the use of both motion and disparity estimation in the coding process of the multiview video sequences. The motion estimation process in MVC is very similar to that of H.264/AVC, only that MVC extended the motion estimation algorithm by incorporating disparity estimation in order to utilize the redundancies in view direction. The concept of ME determines how objects in a scene moves and tries to compute vectors that can represent the estimated motion of the object. Motion compensation makes use of the estimated motion of an object in the scene to achieve the video compression. Efficient ME can minimize the energy in the motion-compensated residual picture and also improve the coding efficiency. Disparity estimation on the other hand, can improve interview prediction which is similar to ME. Although the statistical properties of DVs can differ from MVs, the geometric properties and constraints are always put into consideration during prediction and coding [38]. Similar to MC, redundancy can be reduced in view direction of MVC by way of compensating the target image from the reference image by using the disparity vectors. The DE algorithms in general attempts to match the pixel values in one frame of a view with their corresponding pixel values of another frame in a different view [39]. The difference between views in MVC depends on the disparity effects and while MCP uses reference frames from the same view, DCP uses reference frames from other views to achieve coding efficiency.

2.8. Extending H.264/MPEG-4 for Multiview

The major recent extension of H.264/AVC is the MVC design [40]. Annex H of the H.264/AVC standard specifies in details a number of added basic H.264 syntax to support MVC [13] including:

- Sequence parameter set: This syntax specifies views and anchor/key picture references.
- Reference picture list: structured to enhance interview prediction.
- NAL unit order: The modification here is to utilize prefix NAL unit, which contains additional information about the base view. A compliant H.264/AVC decoder may decide to discard Prefix NAL unit and can go on to decode only the base view.
- Picture numbering and reference indices: Also modified to support the decoding of multiview videos.

2.9. MVC prediction structure

The key concept of MVC is the interview prediction, which is employed in order to fully utilize both the spatial and temporal redundancy to achieve high compression gain. Because all the cameras are basically capturing the same scene from strategic position, substantial interview redundancy is present [17]. Fig. 2.6 illustrates the prediction structure for MVC. The first view in the bitstream also known as the base view is encoded without interview dependencies but with normal temporal motion compensation technique as in 2D video coding. Similarly, it is decoded independently to be backward compatible with the standard H.264/AVC.

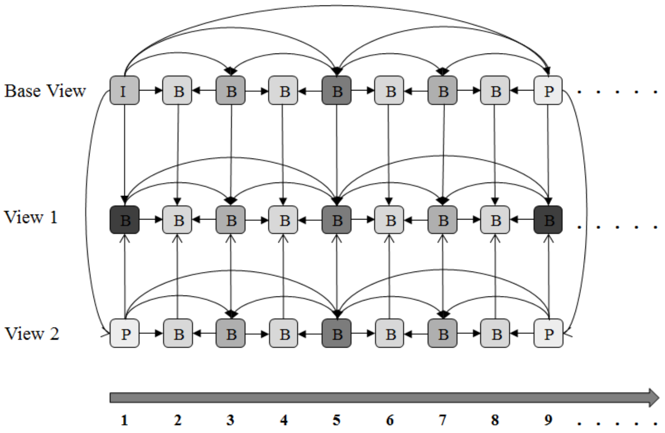


Figure 2.6: MVC prediction structure

All other views also referred to as non-base views depend on the base view as reference for increased compression gain. Interview prediction between the views is similar to temporal prediction in a regular 2D video but utilizes disparity vectors with the interview reference frames. Although high compression can be achieved by the prediction structure, one of the main disadvantages of MVC design structure is that because of the interview dependency, a single bit error in the multiview video bitstream can propagate to subsequent views. This may render the bitstream invalid for decoding or result in severe quality degradation in the reconstructed views. A possible solution to this problem is achieved by employing some techniques presented in chapter six.

2.10. Multiview Video Bitstream

A part of design principle for MVC is that the compressed MVV bitstream must include the base view bitstream which is encoded independently from other views in a format compatible with decoders for single view profile of the standard. As described in sub-section 2.6.1, coded video data in H.264/AVC is organised into NAL units. MVC utilizes this NAL unit type structure to achieve backward compatibility for MVV. This can be achieved by encapsulating the video data associated with the base view in a NAL unit that have previously been defined for 2D video, while the video data associated with non-base view is encapsulated in an extension NAL unit type that can be used for both Scalable Video Coding (SVC) and MVC [17] . Usually, the use of flag is specified to indicate whether a NAL unit is associated with an SVC bitstream or MVC bitstream.

2.11. MVC NAL units

One of the main differences between MVC coded data and 2D bitstream includes the encapsulation of their contents into NAL units and the header structuring. In MVC, the NAL unit has a 4 byte NAL unit header (Fig. 2.7) to support additional information which includes anchor picture_id, view_id, priority_id, temporal_id [41] of the non-base view. The base view of MVC structure is independently coded and compliant with the requirement of H.264/AVC. To contain the coded picture information of the non-base view in the bitstream, a new NAL unit type known as coded slice of MVC extension is introduced [34]. Another unique type of NAL unit introduced into MVC is the prefix NAL unit which includes descriptive and useful information of the coded picture in the H.264/AVC [42]. The prefix NAL unit precedes an associated H.264/AVC VCL NAL unit and holds its essential features in multiview context [34]. Conceptually, the prefix NAL unit holds information about base view VCL NAL unit such as NAL unit type 1 and NAL unit type 5 that precede each prefix NAL unit. Types 1 and 5 are respectively coded slice of a non IDR and IDR picture.

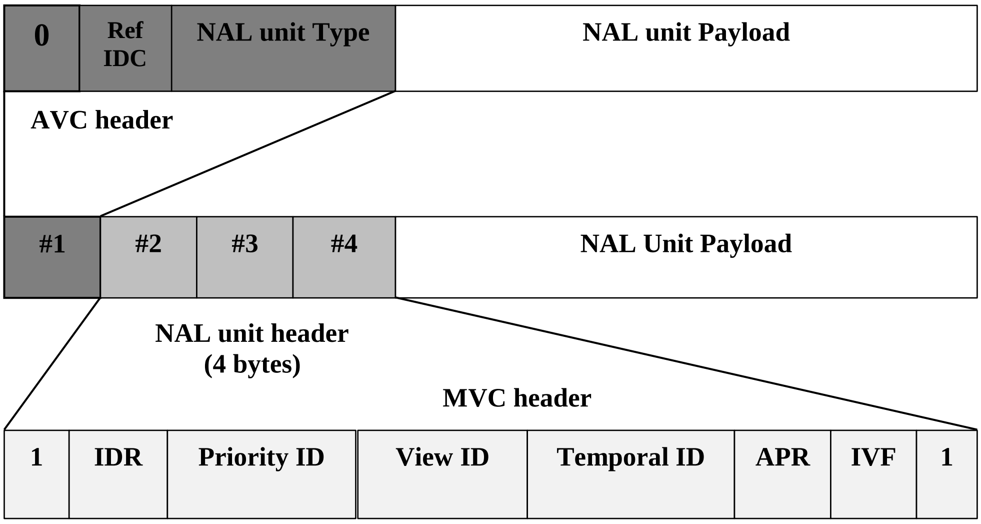


Figure 2.7: MVC NALU header interface [41]

2.12. MVC Decoding Process

Additional high level syntax is required to decode an MVC bitstream; this is mainly signalled to the decoder through the MVC extension of the sequence parameter set (SPS) that is defined by the H.264/MPEG-4 AVC. There are three essential pieces of information that are contained in the SPS extension [17]:

- View identification
- View dependency
- Level index for operation points

In the view identification part high level syntax, the total number of views as well as the listing of identifiers is indicated. The view identifiers are important especially for associating a specific view to a particular index, while the view order index is mainly signalled by the view identifiers. The view order index is critical to the bitstream decoding process because it is responsible for the order in which the views can be decoded.

The view dependency high level syntax consists of a set of information that can precisely show the number of interview reference frames for each of the two reference frame lists that are used in the prediction process, as well as the views that may be used for the prediction of a particular view. Also, separate view dependency information for anchor and non-anchor frames is provided in order to enhance flexibility in the prediction process while ensuring not overloading the decoders with dependency information that may change periodically.

In the case of non-anchor frames, the view dependency only indicates some specific set of views that may be used for inter prediction. The level index for operation points is the SPS extension that is responsible for signalling level information and information about the operating points in the MVC bitstream. It basically specifies the resource requirement of an MVC decoder that conforms to a particular level. In an MVC bitstream, an operating point corresponds to a specific temporal subset and a set of views which can allow the standard to signal multiple level values with each level being associated with a particular operating point. The syntax is an indication of the number of views that are targeted for output and the number of selected views that are necessary for the decoding of specific operating points [43].

2.13. High Efficiency Video Coding (HEVC)

The overall amount of video data rate that is to be delivered across the internet will continue to grow exponentially, driven by everyday increase in users and services and the demand for increasing high quality resolution of video data from SD to HD and beyond. It is becoming difficult and more challenging in the current transmission network to carry these quality requirements and services to the end users, especially video over broadband services that have now become a major phenomenon. The expected emergence of ultra-high HD resolution such as 4K x 2K and beyond in the near future and the increased demand of 3D services such as 3DTV and FVV and many more will be fully supported by next generation display [44]. Therefore, it has become necessary to develop a new video coding compression standard that is capable of efficiently meeting up with the current challenges. HEVC is the new generation of video compression technology with higher compression capability than the existing AVC High profile standard, and has got a potential to support a broad range of current and future applications [45].

2.14. HEVC Standardization

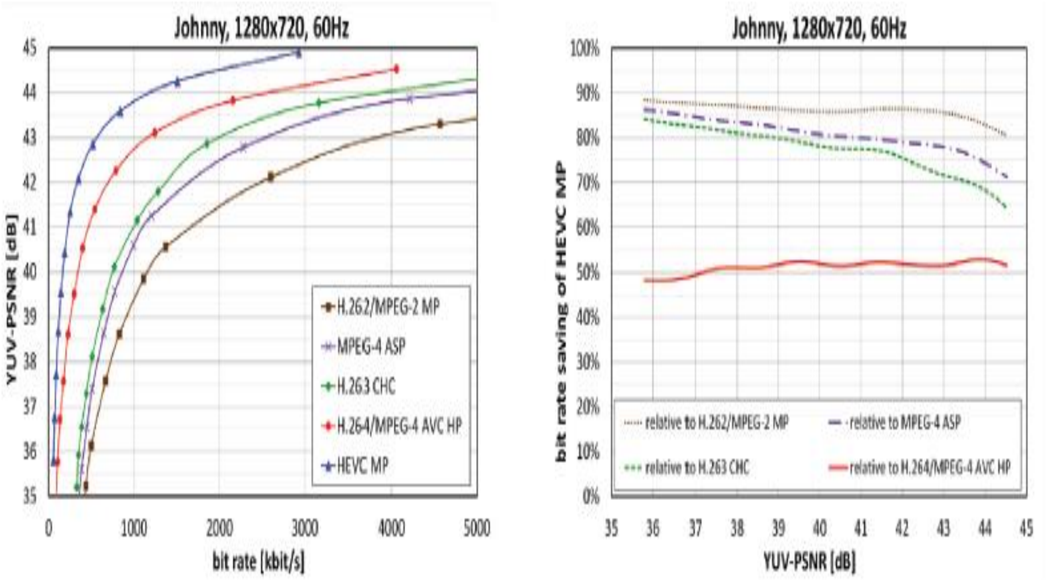
The High efficiency video coding (HEVC) standard is the most recent joint video project of the ITU-T video coding Experts Group (VCEG) and the ISO/IEC Moving Experts Group (MPEG) standardization organizations have been working under a joint collaboration known as the Joint Collaborative Team on Video Coding (JCT-VC) [46].

HEVC offers a much higher and efficient compression gain than its predecessor H.264/MPEG-4 AVC, and is particularly suitable for streaming high resolution videos with a bandwidth saving of around 50 percent.

Basically, HEVC enables a network to stream twice the number of standard TV channels, HEVC can also provide up to four times the capacity on the same network [47]. The HEVC has been designed to address essentially all existing application s of the H.264/MPEG-4 AVC standard and to mainly focus on these two issues: higher video resolution and increased use of parallel processing architectures. The syntax of HEVC is generic and should be applicable to other different applications [46]. Some applications of HEVC include its suitable for the compression of all kinds of video. For this reason three profiles have been defined: Main, Main 10 and Main Still Picture. The main is the all-purpose profile with a depth of 8 bits per pixel that can support 4:2:0 which is the most common uncompressed video format used by consumer devices from their mobile phones to HDTVs. Furthermore, the main 10 has an extended bit depth to 10 bits per pixel, which is also suitable for consumer applications, like the UHDTV, which requires a very high quality. The increase in bit depth is to support wide compression dynamic range without experiencing visual artefacts that are sometimes common with 8 bits. The third profile, Main Still Picture is a subset of Main that is designed to support still images at a depth of 8 bits per pixel. The recent deployment of HEVC released in 2013 is for mobile support and Over the Top (OTT) applications. With this deployment, software implementations capable of decoding HEVC without hardware acceleration can easily be downloaded on smartphones, tablets, and PCs, enabling mobile TV, video streaming and download services on existing devices [47]. As it is common to all past ITU-T and ISO/IEC video coding standards, in HEVC, only the bitstream structure and syntax is standardized, as well as constraints on the bitstream and its mapping in order to generate decoded pictures. The mapping is achieved by defining the semantic meaning of the syntax elements and a decoding process so that any conformance decoder can produce the same bitstream that is in conformity to the constraints of the standards. The restriction of the scope of the standard allows maximum freedom to optimize implementations in a way appropriate to specific applications like balancing compression quality, implementation cost, and other considerations [46].

2.15. Benefits and Complexity of HEVC

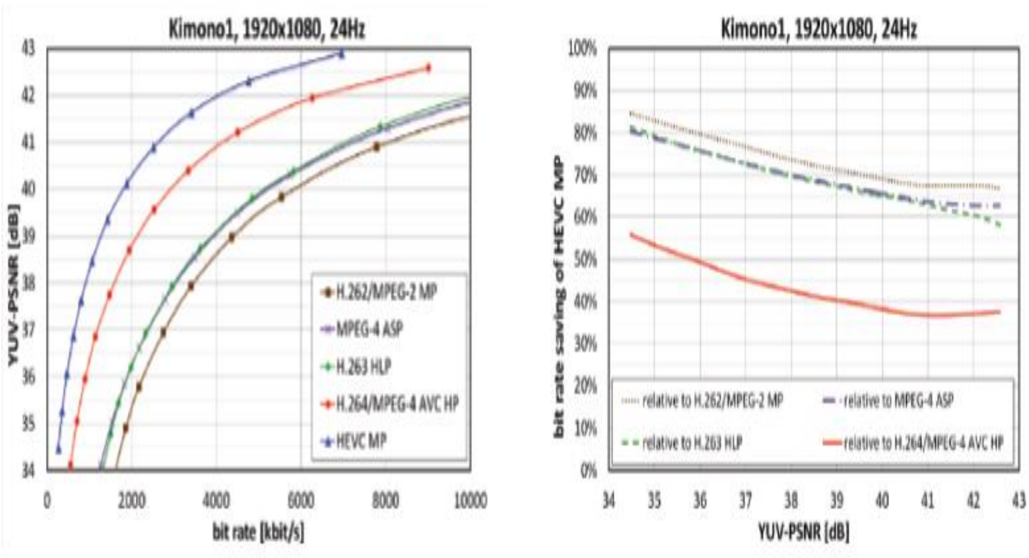
HEVC’s key benefit is the bandwidth efficiency, targeting a 50% reduction in bitrate compared to the current MPEG-4 AVC standard at a comparable video quality. However, for video applications that do not require bandwidth, HEVC can be used to significantly improve video quality at the same bit rate as AVC [48]. The HEVC standard is also developed in order to support and accommodate new and existing applications that include 4K, 3840 x 2160 pixels and 8K, 7680 x 4320 pixels) [49]. While HEVC has demonstrated numerous benefits in different applications, on the other hand, these remarkable achievements are at the cost of high computational complexity of the encoding and decoding processes. Compared with its predecessor, HEVC is 20 to 100 percent more complicated for decoding a bitstream and up to 400 percent more complex when encoding a video sequence (based on preliminary testing) [50]. The HEVC has emerged a new and state of the art video coding standard which is developed by the JCT-VC group to replace the current H.264/AVC video coding standard.. Researchers in different institutions and related industries are coming up with different performance and observations in HEVC codec. The authors in [51] have reported and presented some interesting results about coding efficiency in various video coding standards. In their report, the HEVC (HM-8.0 reference codec) main profile is compared in terms of coding efficiency and bitrate savings with various codecs. These codecs include the JSVM software for H.264/MPEG-4 AVC high profile, Fraunhofer HHI implementation of MPEG-4 visual Advanced Simple Profile (ASP), the H.263 codec of the University of British Columbia Signal Processing and Multimedia Group High Latency Profile, and the MPEG Software Simulation Group for H.262/MPEG-2 main profile. The experiment was conducted for both entertainment and interactive applications with all encoders having the same mode decision, mode estimation, and quantization settings. The bitrate savings of 35.4% in HEVC is recorded compared to H.264/MPEG-4 AVC HP, 63.7% compared to MPEG-4 ASP, 65.1% compared to H.264 HLP, and 70.8% compared to H.262/MPEG-2 MP.



(a) (b)

Figure 2.8: RD curves and bitrate saving plots for interactive applications [51]

Fig. 2.8(a) and Fig. 2.8(b) illustrate the results obtained for the interactive video applications, such as video conferencing. Fig. 2.8(a) depicts the RD curve for the Johnny sequence with a high resolution of 1280 x 720 at a frame rate of 60Hz. The video quality is plotted as a function of the average bitrate. Fig. 2.8(b) is a plot that illustrates the bit rate savings of HEVC MP relative to H.262/MPEG-2 MP, H.263 CHC, MPEG-4 ASP, and H.264/MPEG-4 AVC HP as the function of the PSNR. These results indicate that the HEVC standard clearly outperforms its predecessors in terms of coding efficiency



(a) (b)

Figure 2.9: RD curves and bitrate saving plots for entertainment applications [51]

It can be observed from Fig. 2.9(a), HEVC provides significant gains in terms of coding efficiency relative to the older video coding standards. It can also be noticed that the coding efficiency gains for lower bitrate range i.e. conversation application are higher than the average results reported for the entertainment application for their report. The work in [52] presents a subjective evaluation of HEVC main profile compared to the AVC high profile. The test compared visual quality for twenty video sequences with resolutions ranging from 480p to Ultra HD that were encoded at various bitrates or quality levels. The analysis of their subjective test results shows that HEVC test points at half or less than half the bit rate of the AVC reference were found to achieve comparable quality in 86% of the cases. The estimation of the bit rate savings from their report confirmed that the HEVC main profile achieves the same subjective quality as AVC high profile while requiring on average approximately 59% fewer bits. Fig. 2.10 shows the average BD-rate savings of HEVC in comparison to AVC.

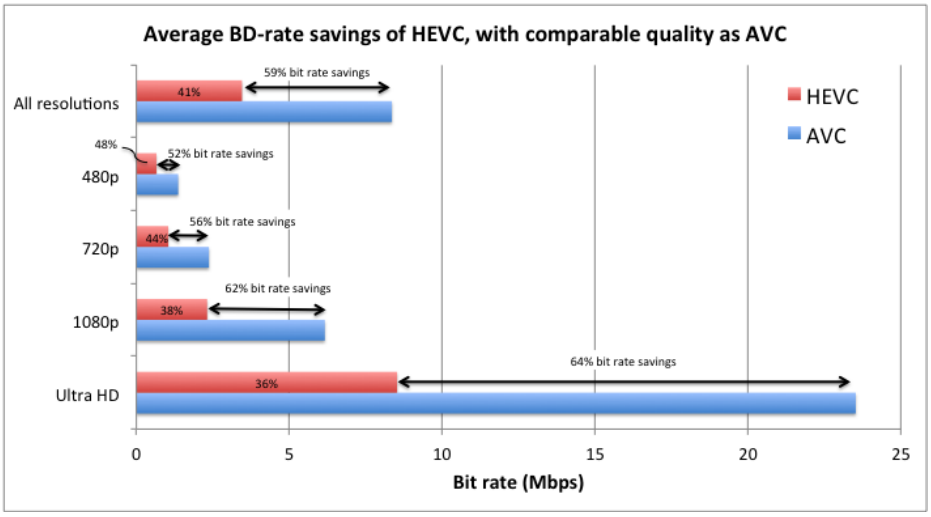
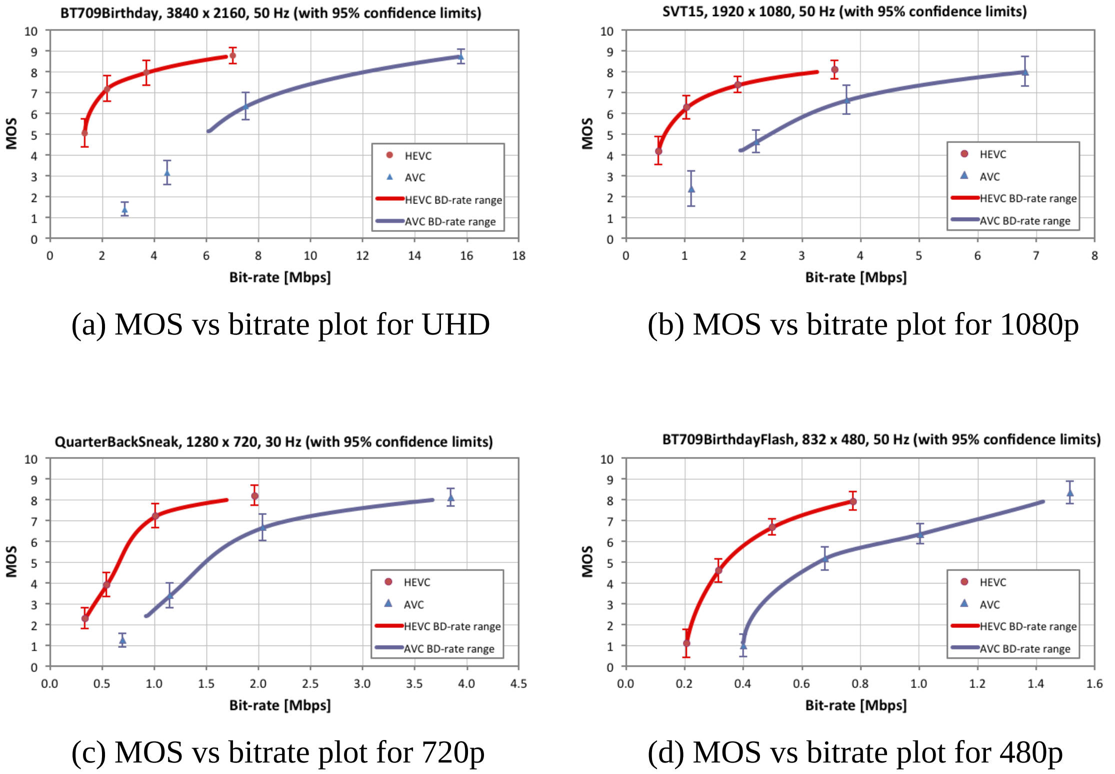


Figure 2.10: Average bitrate savings (BD-Rate) of HEVC compared to AVC [52]



(a) MOS vs bitrate plot for UHD

(b) MOS vs bitrate plot for 1080p

(c) MOS vs bitrate plot for 720p

(d) MOS vs bitrate plot for 480p

Figure 2.11: MOS vs bitrate plots for different sequences [52]

The analysis in Fig. 2.11 shows the MOS vs. bit rate plots indicating that the HEVC can significantly reduce the amount of bitrate relative to the AVC for different test sequences, resolutions, and frame rates.

2.16. HEVC Extension

For more advanced applications of HEVC standard, such as 3D content production for heterogeneous devices and network, the Joint Collaborative Team on 3D Video Coding Extension Development (JCT-3V) is established. This is solely for the development of new 3D standards including extensions of HEVC [53]. The 3D-HEVC [54] standard goes beyond traditional stereoscopic and multiview representations of video and extends to include the use of depth information and view synthesis. More advanced 3D capabilities with much higher resolution and visual quality aim toward future consumer electronics and content that can be utilized for theatre, home, and mobile applications. The 3D-HEVC is based on the Multiview-plus-Depth (MVD) format. This extension can support the coding of multiple views and associated depth information. This is achieved by introducing new advanced coding tools to the HEVC design, which improve the encoding capability for both the video views and the depth data [55]. Some of these new key coding tools in HEVC design include larger block sizes, hierarchical block coding, higher number of intra prediction modes, advanced motion vector prediction (inter frame), sample adaptive offset filtering and waveform parallel processing. Scalability is a key attribute to any video coding standard that allows trimming and resizing of video streams to suit different network conditions and receiver capabilities; scalable extensions to HEVC has just been released in July 2014. Range extensions that support many colour formats as well as increased bit depths are another field of active research. In additions to these extensions, further developments are expected to be available in the current HEVC framework, such as higher compression gain at a lesser complexity. It is likely that the full potential of HEVC is yet to unfold; when this video coding standard becomes fully established, it will no doubt relieve the current traffic load in networks, advance video based services and enable new innovations.

2.17. HEVC and Future challenges

The HEVC is designed and finalized under three different stages, but indeed it will still take time for the codec to support consumer electronics or end users. In addition, the industry has to study how well to optimize HEVC compressed video content in terms of quality and to be bandwidth efficient, while maintaining constant visual quality. Although HEVC promises 50% efficiency, we are not there yet as early tests revealed compression efficiencies to be between 15% - 35% [56].

Currently, the HEVC codec takes about 10 times longer than H.264/AVC at the same frame rate due to extreme encoder computation and complexity; this will definitely pose a challenge for encoding and transcoding applications especially at the early stage [57].

2.18. Conclusions

In this chapter, the video coding concept and the advancement in the video coding tools have been discussed. An overview of the key components of the standard H.264/AVC codec and the extended versions of the H.264/AVC standard, which include the JMVC and the HEVC standards, are also presented. The chapter elaborated in more details the necessary development and new features that have been added to the multiview coding system and the benefits associated with the concept. In addition to that, a general overview of the new state of the art HEVC standard is presented, which included the standardization, extensions, benefits and future challenges. The chapter also highlighted some experimental results that indicate a superior coding performance of the newly released state of the art HEVC coding tool against the current H.264/AVC codec. Several studies have reported a significant improvement in terms of coding efficiency and picture quality. To this extent, objective and subjective results that have been reported confirm that the main aim of developing the HEVC standard in order to deliver the same visual quality as H.264/MPEG-4 AVC high profile at half the bitrate has been accomplished. The next chapter discusses the concept of 3D video coding and communication pipeline.

3. Chapter Three: 3D Video Systems and Communication

3.1. Introduction

Digital media has influenced and changed modern society over the last 2 decades significantly. Vast amount of media is produced, processed, stored, and transmitted in digital formats with digital equipment. Applications, terminals and content are merging faster than ever. We can watch TV with our mobile phones, surf the web with the TV set, and modern home PCs are powerful multimedia workstations capable of more or less everything [30]. An important factor for this success story is the availability of international standards for digital media formats. They provide interoperability between different systems while still allowing for competition among equipment and service providers. ISO MPEG is one of the international standardization bodies that play an important role in the digital media market.

Recent research and convergence of technologies from computer graphics, computer vision, multimedia and related fields enabled also the development of new types of media, such as 3D video and its applications that expand the user's sensation far beyond what is offered by traditional media. The concept of 3D video is commonly understood as a type of multimedia visual application that provides depth perception of the observed scenery. However, this is achieved by the use of special hardware 3D display systems that ensure each specific view is projected into each eye of the viewer [58].

3.2. 3D Video Fundamentals

In order to better appreciate 3D imaging and video technology, a basic understanding of HVS is required. Basically, HVS [59, 60] consists of two elements, the two eyes and the brain. As the two human eyes of an individual are separated by about 6-8 cm, the 3-D depth perception is realized by two slightly different images projected to the left and right eye retinas (binocular parallax) and then the brain fuses the two images to give the depth perception. Basically, each of the eye comprises of a retina that receives information and transfers it to a region of the brain known as lateral geniculate body and finally to the visual cortex via the optic nerve. Pictures produced at each of the retinas are viewed up-side-down and as the visual information is processed by the visual cortex; one single upright image is generated.

3.3. 3D Content Creation

The 3D video content can be generated by numerous processes with different types of cameras. The stereo or depth camera can be positioned in such a way that they can capture the video and associated disparity information simultaneously. For multiview camera setup, each camera captures multiple images simultaneously from different angles. Matching process is required to generate the disparity map for each pair of cameras so that the 3D perception is estimated from the disparity map [61]. The depth map information can be derived from a number of ways, which include linear perspective of 3D scene, and occlusion of objections [61].

3.4. 3D Video Compression

Many 3D compression formats have been proposed and developed for more than a decade. Like the conventional 2D video coding, most of the 3D video compression techniques are developed on the basis of the H.264 advanced video codecs. In general, video compression considers a trade-off between an adequate level of system complexity that will give high coding efficiency and affordable communication bandwidth through which the content can be delivered. All the 3D compression formats currently available have different pros and cons with regard to their functionality, efficiency, complexity. However, they generally share the following properties:

- Utilization of the existing 2D broadcast infrastructure
- Require less or no change to device components
- Backward compatibility
- Support wide range of display devices and allow for future extension
- High quality

3.5. Conventional Stereo Video

Stereoscopic systems are widely known for their simplicity in terms of 3D video data representation. A stereoscopic video can be used to provide 3D perception through a pair of video, which include a left and right view. When pair of 2D video is generated, the 3D impression can be experienced usually with the help of hardware devices when both the left and right views are viewed by the corresponding left and right eyes of the viewer.

3.5.1. Simulcast Video Coding

Simulcast is a common video representation format which is very common and widely established especially from the H.264 family of coding standards. In this concept, each view is independently encoded as in Fig. 3.1, transmitted, and decoded without having to exploit the redundancies between the views [62]. The technique is straight forward and less complex in terms of encoding and decoding of the views that are being used. The inability of the technique to exploit the correlations between views literally generates huge volumes of data, especially when the number of views increases. This fact affects the technique by making it inefficient for both storage and transmission. Simulcast can be made more efficient to utilize if the stereo multiplexing format described in the next section is applied to it.

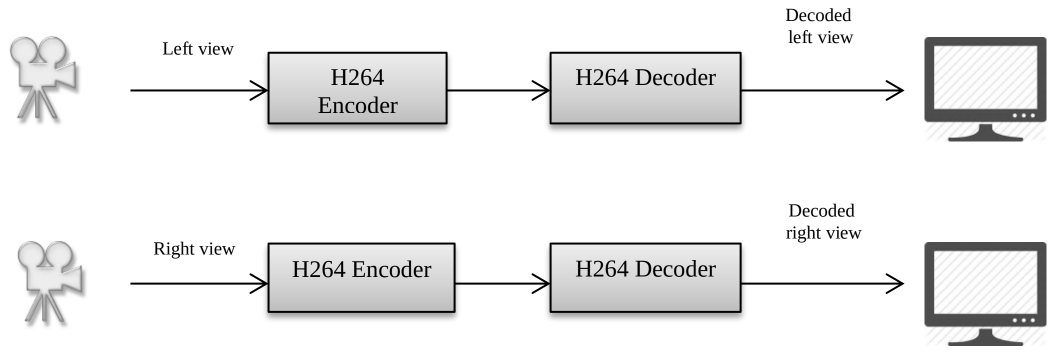


Figure 3.1: Simulcast video coding technique

3.5.2. Frame compatible Stereo Formats

The concept of the technique is that, the frames from the left and right views are subsampled into half resolution and then embedded into a single video frame. The multiplexed frames can then be compressed and transmitted. When the frames are received at the decoder, they can be de-multiplexed and reconstructed back into two views for viewing. This concept of subsampling can achieve reduction in video data to suit the 2D video broadcast infrastructure. The subsampling and multiplexing can be achieved by basically (a) time multiplexed format – the left and right frames are interleaved as alternating frames or fields (b) spatial multiplexing format – the left and right frames would appear in either side-by-side, as proposed by Sensio, RealD, and adopted by Samsung, Panasonic, Sony, Toshiba, and Direct TV [61] or over/under format, proposed by Comsat (c) checkerboard format.

Usually, in spatial multiplexing, the left and right frames are embedded in either horizontal or vertical dimension to fit in within the original size of the frame at the expense of reduced spatial resolution. The concept of stereo multiplexing is illustrated in one of the following ways in Fig. 3.2 and Fig. 3.3.

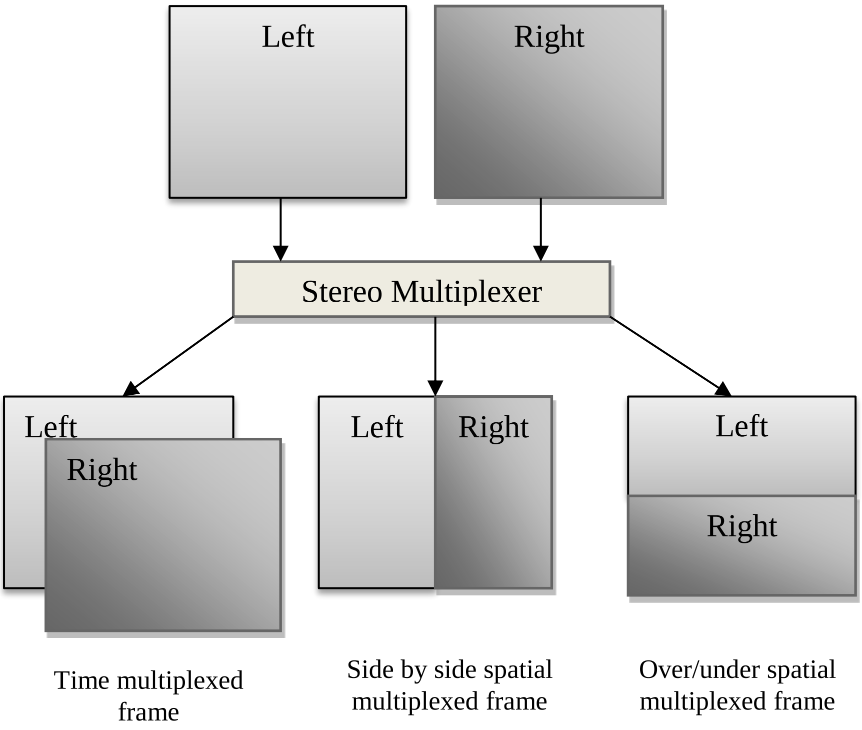


Figure 3.2: Time multiplexed, side by side and over/under frame compatibility

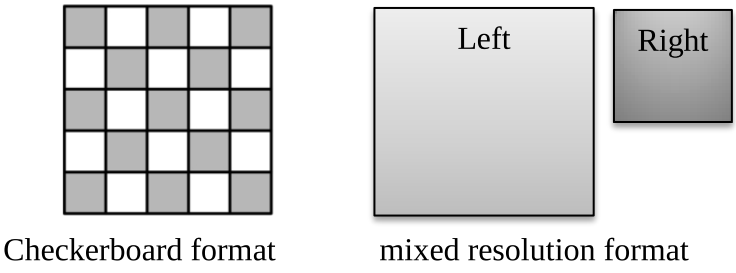


Figure 3.3: Checkerboard and mixed resolution formats

In addition to the above stereo multiplexing formats, another representation is derived, which is based on the binocular suppression theory. In mixed resolution, quality perception can be achieved when the resolution of one view is reduced by subsampling to a different resolution and then compressing the views independently [61].

3.5.3. Frame Compatible Coding with SEI Message:

In H.264/AVC standard, supplemental Enhancement Information can be transmitted with the frame compatible video data in order to signal useful video information (Fig 3.4) that includes frame packaging arrangement, sampling relationship between the two views and the view ordering to the decoder for processing [63]. When the video signal is received by the decoder, it can recognize the format defined and perform all the required processing such as scaling de-noising, or colour format conversion, based on the frame compatible format that is specified during the multiplexing process. Display devices can also benefit from the SEI message in order to be aware of the frame compatible formats of the video. This is achieved by transmitting the format information via supported interfaces such as the High-Definition Multimedia Interface (HDMI) [63].

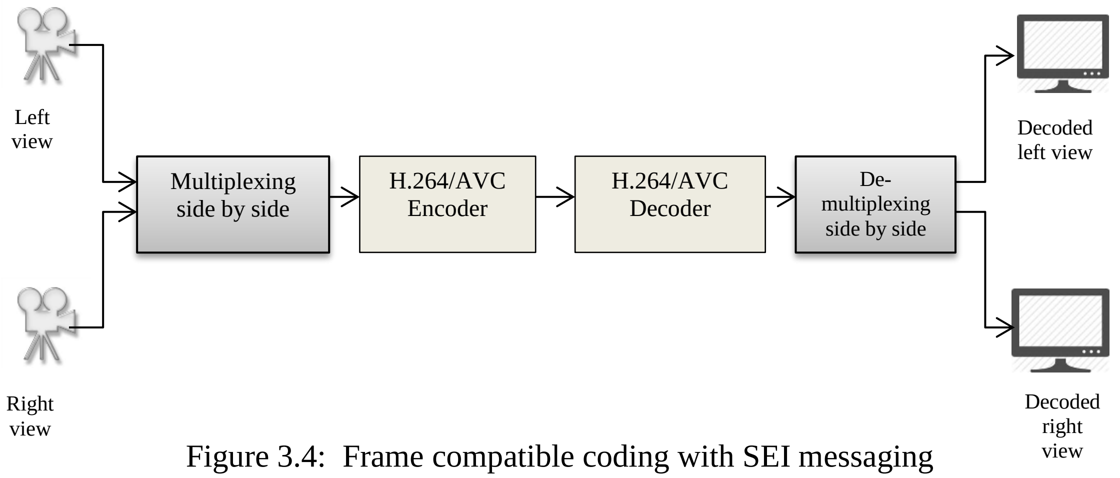
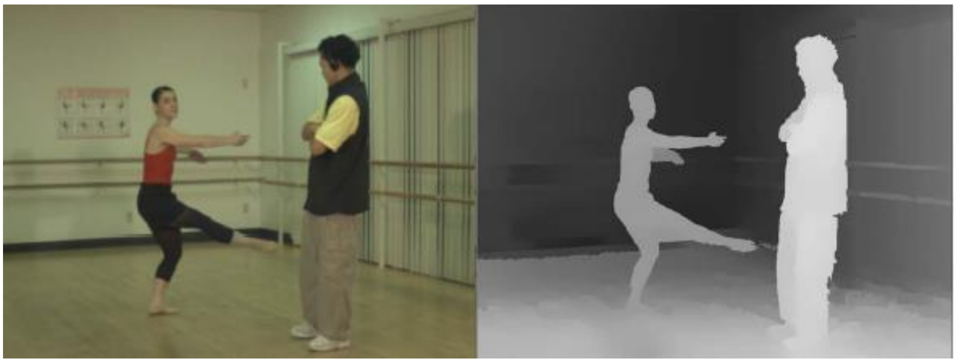


Figure 3.4: Frame compatible coding with SEI messaging

3.6. Depth information and coding

The conventional 2D colour video sequence captured by a CCD camera is also known as texture video. The depth video presents the depth information as grey scale, which is defined as the distance from camera to object in a scene [64]. The depth information has characteristics of both video signal and data on z-axis in world coordinate. The best way to capture the depth information is by employing a depth camera such as Z-Cam^{TM} and SR^{TM}. However, because of hardware performance limitations, the depth information is produced by either stereo or multiview matching method [65]. Fig 3.5 shows an example of a colour and depth image.



(a) Colour image (b) depth image

Figure 3.5: Colour and depth video of Ballet sequence [66]

The depth image does not have the chrominance components as can be seen in Fig. 3.5b, it only consists of luminance components because of the act that depth information is presented as a quantized image. The depth information is usually quantized to 8-bit image to be compatible with monoscopic video signals, and the maximum and minimum values on the z-axis in world coordinate are defined for view synthesis. The minimum and maximum values defined as depth range is restricted to a range in between two extreme Z_{near} and Z_{far} distances of the corresponding 3D point from the camera respectively [66]. Also, the closest point on the z-axis is associated with the value 255 and the farthest is associated with the value 0. One of the main characteristics of depth information is that it does not have textual information or shadow, and because it is not affected by light or illumination. These features of depth information are utilized in order to achieve a high coding gain in depth video coding. In general, textual information in a video makes it harder to encode because of the presence of a lot of high frequency components. Also in a standard textual coding where an object and a background are overlaid by shadow, finding a suitable motion vector can be a difficult task. However, the depth image and the colour image share a common feature which is the object boundary area. The boundary of the depth image has similar shape with boundary of the colour image, which can show similar movement to each other because they represent the same objects [64]. It is very essential to generate a high quality depth data for 3D video applications. As stated earlier on, depth estimation algorithms are mostly used in order to match corresponding signal components in two or more views using matching function with different area support and size [67]. The algorithms apply a matching criterion which include sum of absolute differences, cross correlation.

Furthermore, the depth estimation algorithms try to optimize the estimation, based on different approaches, such as graph cuts, belief propagation, and the plane sweeping technique in order to generate a high quality. Depth information in recent years has been studied extensively especially multiview video. While depth estimation algorithms are currently advancing, they can be prone to error due to problems of mismatch, especially for partially occluded image and video content that is only visible to one camera [67].

3.6.1. Video-plus depth

The video-plus-depth (V+D) representation provides an alternative to stereo video representation format to achieve 3D perception in applications such as 3DTV and FVV [68]. V+D is flexible and supports adjustments in the stereo rendering at the decoder. Also it creates virtual views in order to reduce the volume of video data to be transmitted or stored [63]. In this technique, a video signal, and a per-pixel depth map is transmitted to the user. From the video and depth information, a stereo pair can be rendered by 3D warping at the decoder. Depth information is also regarded as monochromatic, luminance-only video signal. In depth enhanced coding techniques such as V+D, usually the depth map is specified as a grey scale image. These grey scale images can be fed into the luminance channel of a video signal and the chrominance can be set to a constant value. The resulting standard video signal can be compressed by any state of the art video coding tool [66]. For V+D, view synthesis is necessary at the receiver for generating the second view of a stereo pair to be presented on stereoscopic displays. However, the extended concept has demonstrated some level of efficiency in 3D perception but at the cost of an increased complexity and computation. V+D format consists of a conventional 2D video and an associated per pixel depth information. The video and depth information can be rendered into a stereo pair through the view synthesis display [69]. Such applications and their algorithms can be very complicated and likely to fail due to an error. The concept of video plus depth can be illustrated in Fig. 3.6, where depth information is used as additional video information in order to generate and reconstruct the video sequence at the decoder and hence give a 3D video perception to the user.

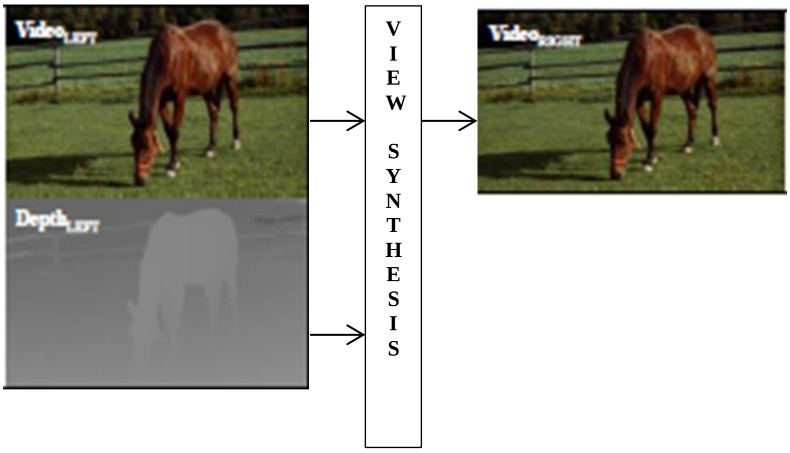


Figure 3.6: Video-plus-Depth format and its application [69].

In the V+D format, the decoder utilizes the depth map in order to generate the second view. V+D systems have the benefit of not using the full view to generating the second view rather; the depth map is utilized for efficient data transmission or storage. However, the depth map requires on average about 10 – 20% of the original video information for transmission or storage [70]. The major challenge with V+D to stereo rendering is the visual quality of the synthesized view. Rendering artefacts may lead to wrong and annoying 3D impression which usually result in a situation where the left and right view are not consistent [69]. The V+D format provides a very limited FVV functionality. If the head position of the user is tracked, the rendered stereo pair can be adjusted to the actual position. With the head motion, parallax viewing becomes possible in a very limited navigation range [71].

3.6.2. Multiview Video-plus-depth

Multiview video-plus-depth (MVD) is an extension of the video plus depth format [72]. While enhancing 3DTV, MVD representation is capable of rendering any intermediate view and free navigation in between the original cameras [71]. The extension of multiview video to MVD extends the navigation range significantly, which allows the virtual intermediate views to be rendered for anywhere in between the views, thus providing advanced FVV functionality. Compared to MVV format which synthesizes scenes by using image interpolation, the main advantage of MVD format is that the virtual views from the arbitrary viewpoints positions can be conveniently generated through DIBR technique for interactive applications [73].

Multiple cameras from slightly different angles capture a scene in the form of video that serves as input to the encoder. The MVD encoder is task with the function of deriving the depth information of each video and also extracting the 3D representation from the input videos. A coded representation of the video sequence is received by the MVD decoder for decoding and followed by multiview rendering. In MVD representation, the colour views and the corresponding depth maps should be coded with a high level of accuracy so that the decoder can synthesize the virtual view of a higher level quality. The virtual view is usually rendered by the depth image based rendering (DIBR) technique and its performance depends highly on the quality of the depth image [74]. In depth enhanced 3D video systems, efficient depth estimation and coding are crucial in order to achieve efficient and reliable 3D perception. The generated depth map in MVD systems is not displayed, but to be used in order to synthesize intermediate views. Rendering quality is very essential for more efficient 3D perception and viewing by a user. In general, depth information could be used at the receiver to generate additional views at the encoder to achieve more efficient coding with view synthesis prediction schemes [75]. It is a good requirement for 3D depth enhanced applications such as 3DTV and FVV to maintain the fidelity of depth data, because the performance and quality of the view synthesis highly depend on the accuracy of the geometric information provided by the depth. Therefore, it is important to consider a good trade-off between the quality of the depth information and the transmission rate of the channel. The MVD representation is designed to give high 3D visualization quality and high resolution at the cost of higher bitrate and increased complexity. The video data processing of this technique at both the sending and receiving ends is computationally intensive and error prone. Usually, the associated depth information of all the views is estimated which is followed by compression and transmission of the video signal. At the receiver, multiple virtual views are rendered from the received video data after decoding [68]. The technique is depicted in Fig. 3.7 with all of the processing chain involved in MVD.

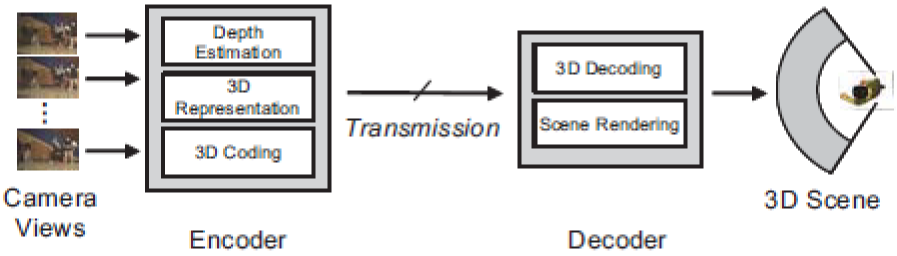


Figure 3.7: Multiview plus depth representation [66]

3.6.2.1. Virtual View Rendering

The main advantage of MVD representations in contrast to MVV is that due to the availability of depth information, rendering of 3D based applications like FVV can be realized. The combination of MVD with camera geometry gives the possibility to synthesize or render arbitrary intermediate views from a 3D representation of the scene. The process of virtual view rendering uses pairs of neighbouring original camera views to render arbitrary virtual views on a specified camera path between them. Usually, the relationship between points in 3D scene space and the values in the depth image is defined by the projection matrix and the quantization function. The projection matrix of a virtual camera is calculated from the two original cameras projection matrices by spherical linear interpolation (SLERP) and linear interpolation (LERP) [76]. Now the two original colour point clouds can be projected into the virtual camera view as illustrated in Fig. 3.8 top - left and right. Eventually, the two rendered colour images are merged together using the information from the rendered depth maps as well as texture weighting according to the position of the virtual camera relative to the original camera. This is shown in the centre image of Fig. 3.8.

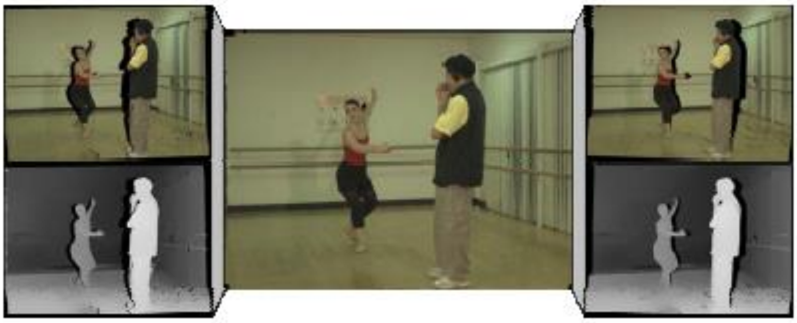


Figure 3.8: Rendering of virtual intermediate view in MVD [76]

3.6.3. Layered Depth Video

Layered depth video is an alternative to MVD that is derived from the concept of layered depth images (LDI) proposed in [77]. Layered depth video (LDV) is a representation that allows the rendering of video signals on a multiscopic 3D display. The technique consists of the original video with associated depth map and additional residual layers [78]. The additional residual layer in LDV includes the content of an image that is covered by foreground objects in the main layer as illustrated in Fig. 3.9. Certain types of LDV consist of one colour video with associated depth map as the main view together with one or more residual layers of colour and depth [66]. In addition to that, the residual layers also include information from other viewing directions that are not covered in the main view. The reference view of the camera is warped onto the other views and the pixels corresponding to the holes are extracted from the original views, which are inserted into the layers of the LDV. Thus, each pixel in the LDV represents a 3D vector also referred to as depth pixel [79]. The LDV concept is introduced in [78] and later developed in [80] [81].{{}} The problem with this method is that every pixel does not necessarily exist in every view, which results in holes occurring when the central view is projected. View synthesis reveals the parts of the scene that are occluded in the central view and make them visible in the side views by a process known as disocclusion. The solution to this problem is by pre-processing the depth video to allow the reduction of depth data discontinuities in a way that minimises the disocclusion. Introducing filter-induced distortion to the depth video may reduce the depth perception of the user. However, it is possible to remove disocclusion by considering more complex multidimensional data representations, such as the advanced LDV data representation. The technique allows the storage of additional depth and colour values for pixels that are occluded in the central view. This extra information provides the necessary information to fill in disoccluded regions in rendered views.

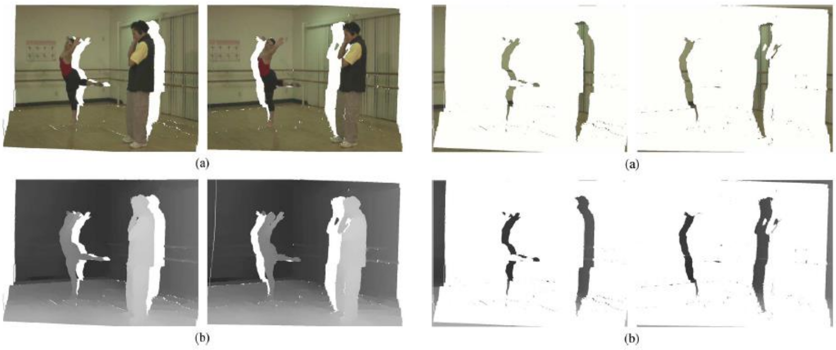


Figure 3.9: Layered depth video [81]

The left side of Fig. 3.9 represents the 3D warping of the central view into both side views. (a) Projected texture image (b) projected depth image. The right side of Fig. 3.9 represents residual data in both side views, (a) residual texture image (b) residual depth image. The 3D warping of the central view into both side views usually reveals the covered parts, which can also be transmitted along with the central view. The disoccluded regions are mainly concentrated along the depth discontinuities of foreground objects. Basically, the side views are reduced to residuals for the texture and depth image as shown in Fig. 3.9 (right-side, a, b), by subtracting the projected central view from a given side view. This technique has the advantage of reducing the data rate significantly. At the user side, the central view and residual data are extracted to reconstruct original side views Fig. 3.10, which gives a new viewing experience and a high degree of user interactivity.

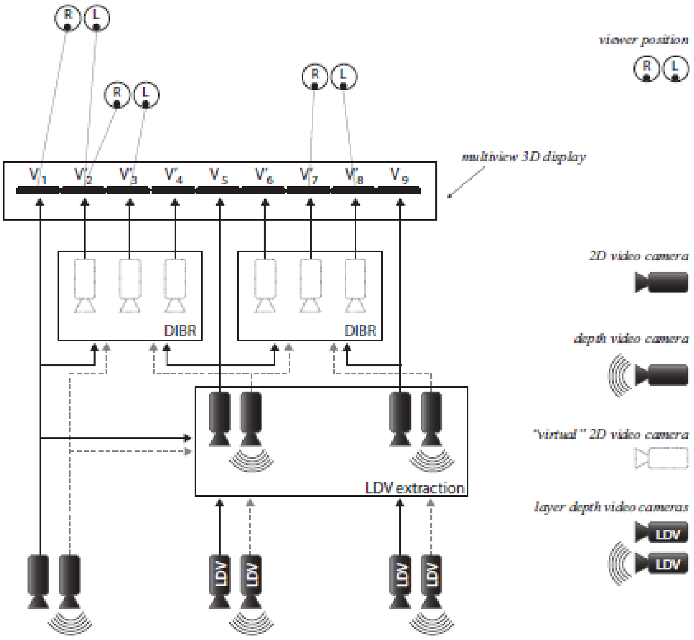


Figure 3.10: Multiview auto stereoscopic displays based on LDV content

In comparing between the LVD and MVD, it has also been reported in [66] that LVD may be more efficient in terms of performance than the MVD because less video data is transmitted. However, artefacts may be high because of additional error prone vision manipulation that is included, which operates on partially unreliable depth data.

3.7. 3D HEVC Extension

Recent standardization of HEVC standard has led to the development of HEVC extension for 3D video that can support the coding of multiview videos [82] and associated depth information. The HEVC extension for 3D video coding is developed based on implementation of new coding tools to the existing HEVC standard. The concept has achieved coding efficiency for both dependent video views and the associated depth information. The 3D-HEVC is the emerging standard for 3D video coding that is designed to encode 3D video content. The 3D-HEVC utilizes all the additional coding components to the HEVC standard to achieve efficient encoding of the texture videos with corresponding depth data. The concept of 3D-HEVC is similar to the Multiview plus Depth (MVP) representation format. However, the 3D-HEVC encoder as part of its features should be backward compatible and be able to encode all the texture videos without having to use their corresponding depth data. The depth information is represented by 8-bits sample that construct a monoscopic picture i.e. it uses shades of grey to represent the distance between the camera and the object. Furthermore, depth maps are different from texture data in the sense that they are characterized by sharp edges which represent object borders and large areas of constant regions that represent object areas [83]. Fig. 3.11 illustrates the coding block diagram of the HEVC extension for 3D video coding.

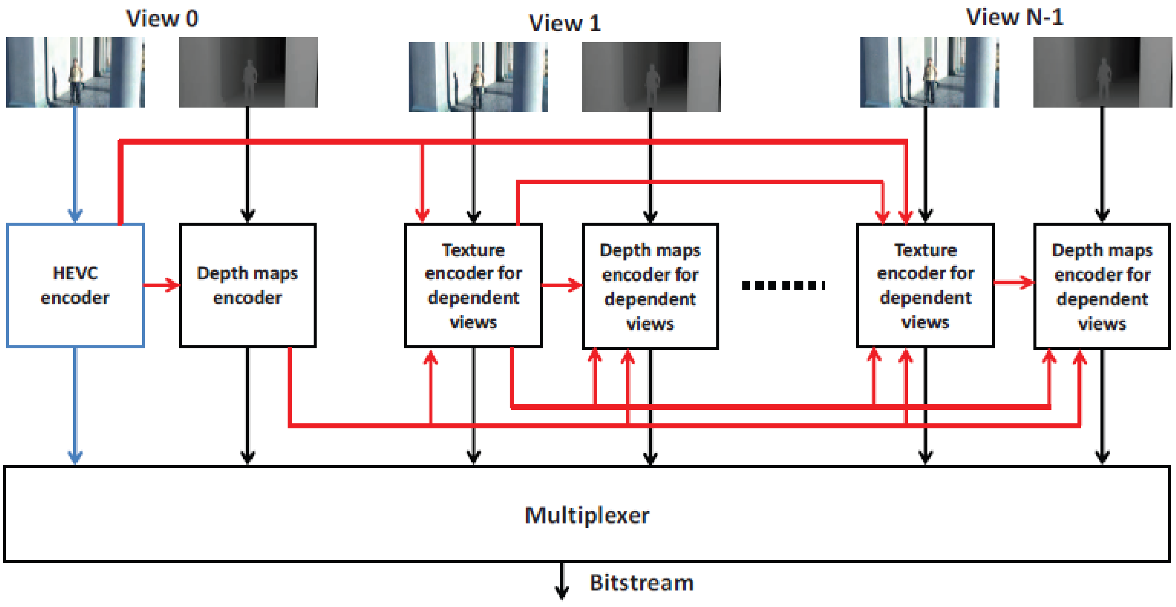

View 0
View 1
View N-1
HEVC encoder
Depth maps encoder
Texture encoder for dependent views
Depth maps encoder for dependent views
Texture encoder for dependent views
Depth maps encoder for dependent views
Multiplexer
Bitstream

Figure 3.11: 3D-HEVC video coding architecture [83]

The 3D-HEVC standard uses the same coding principle as HEVC by dividing the encoding block into Coding Three Units (CTU), which can be as large as 64x64 samples. The CTUs can be further divided into smaller units called Coding Units (CU) for use as a basic unit of intra and inter coding. From Fig. 3.11, the design utilizes inter-component dependencies between texture and depth data. Each texture picture of the view is associated with a depth map, which can be encoded access unit by access unit. One access unit may consist of all texture pictures and their associated depth maps captured at the same time instant. The coding order of access unit does not have to be the same with the display order. By default, the texture of a view is always encoded first before utilizing the depth data by exploiting the 2D HEVC algorithm. However, the depth map of the base view can be used to perform view synthesis prediction in the dependent view, which requires some additional tricks since the corresponding areas of the two views are not co-located. A desirable feature of this technique is that the stereo video can be easily extracted to support existing stereoscopic displays. When that happens, the dependency between the video data and the depth data may be limited [53]. The HEVC codec also supports the decoding of video data only. This can be achieved by configuring the inter-component prediction such that only video pictures are decoded independently of the depth data [84]. Also from the technique, the dependent views are coded with the same concept and tools as the independent view. However, additional tools have been integrated into the HEVC design, which utilises already coded data from other views in order to represent a dependent view efficiently.

These additional tools include disparity-compensated prediction, interview-motion parameter prediction and interview residual prediction. Conceptually, the disparity-compensated prediction is supported for the coding of depth maps of dependent views while the interview motion prediction and residual prediction can only be utilized for videos from dependent views. In general, multiview HEVC depth enhanced extension uses inter-component dependencies between textual and depth data which results to joint coding of texture and depth. However, there is a slight difference and restriction in this concept as the depth map of the dependent view is not allowed to be utilized when coding the textual data of the dependent view. The depth map of the base view can also be used to perform view synthesis prediction in the dependent view. Usually, further manipulation is involved, since the corresponding areas of the two views are not co-located [82].

The newest test model of 3D-HEVC can be found in [85]. Following the recent standardization of 3D-HEVC standard, several study and work has been reported. Mostly a comparison and performance evaluation between the 3D-HEVC standard and other standardized 3D coding techniques available in the literature. Because of the high coding efficiency and capability of HEVC standard, various existing 3D coding techniques such as MVV, MVD, and 3D holoscopic coding demonstrate a higher coding efficiency when utilized in the 3D-HEVC test model. The authors in [83] show a comparison in terms of bitrate saving between 3D-HEVC Test Model (HTM 6.0) and the Joint Multiview Video Coding (JMVC 8.5) reference software. Because the MVC standard is based on H.264/AVC standard, the depth map coding feature of 3D-HEVC codec was disabled for fair comparison. It is known that the standard JMVC codec does not support depth coding but uses only the texture data to encode all the views. In their report, different test sequences are encoded with both reference software at QP values of 25, 30, 35, and 40. The results of their experiment are presented in table 3.1, which shows the percentage of BD-rate reduction of 3d-HEVC standard compared with JMVC standard. It can be seen that the average bit rate saving of 51.8% is achieved, compared to JMVC with the same quality.

Table 3.1: BD-rate reduction for different test sequences

Sequences	BD-rate reduction
Balloons	43.9%
Kendo	54.0%
Newspaper_CC	49.0%
GT_Fly	51.2%
Poznan_Hall2	68.7%
Poznan_Street	46.2%
Undo_Dancer	49.5%
1024x768	49.0%
1920x1088	53.9%
Average	51.8%

Fig. 3.12 illustrates the graphical comparison between coding with 3D-HEVC tool and the MVC tool in terms of the bitrate performance for Kendo sequence. As it can be observed, the 3D-HEVC is able to significantly demonstrate better performance in terms of bitrate reduction and video quality compared to MVC standard. Similarly, all other test sequences from table 3.1 show that we can achieve a higher coding efficiency with the 3D-HEVC test model than using the MVC reference codec.

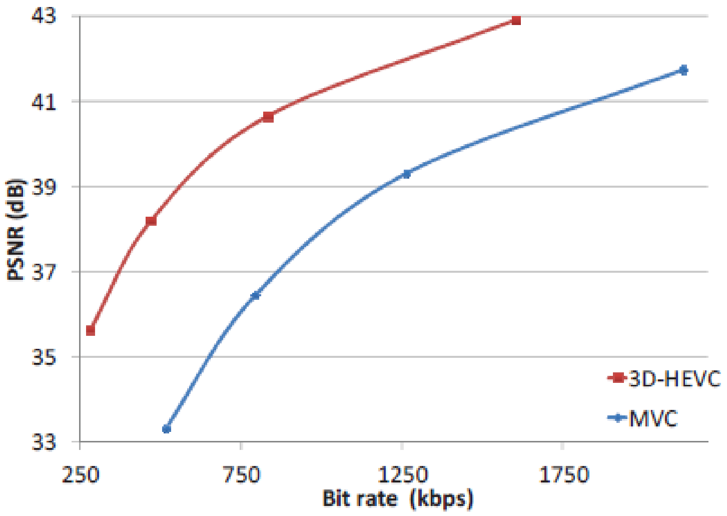
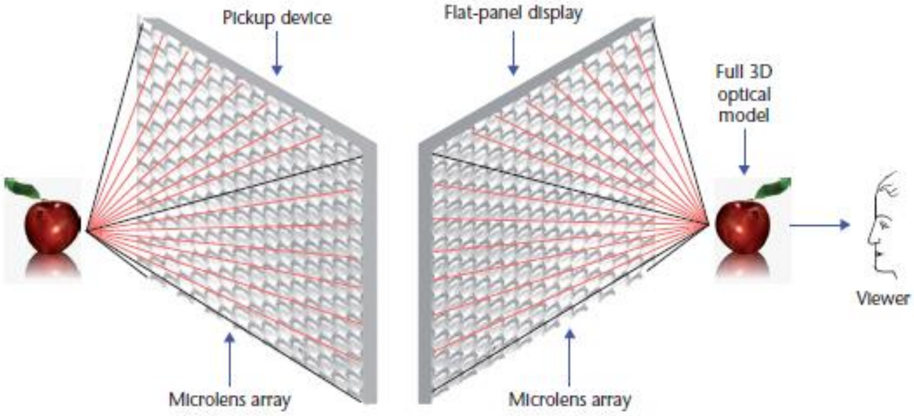


Figure 3.12: coding efficiency comparison for 3D-HEVC and MVC standards [83]

3.8. 3D Holoscopic Video Coding

The 3D Holoscopic technique is also known as integral imaging [86] that is based on an auto-stereoscopic light field technology. The concept is capable of recreating and transmitting the light intensity direction and information coming from a 3D object to the viewer's eyes. This concept enhances a more natural 3D sensation of a scene. Recent study in 3D Holoscopic technology is considered to provide better and more natural 3D perception. The technique allows better accurate convergence for an efficient 3D viewing and delivers more accurate depth information that minimizes the effects of eyestrain when compared with the actual stereoscopic and multiview technologies [87]. The 3D Holoscopic systems also allow continuous motion parallax throughout the viewing zone in both horizontal and vertical directions due to the optical structure of a micro-lens array.[88]. In order to provide 3D holoscopic content with convenient resolution to fit in the HD and higher resolution requirements, high definition content is required. Consequently, efficient compression tool is essential and required for reliable transmission or storage of the huge amount of data captured. Due to the small angular disparity between adjacent micro lenses, a significant cross correlation exists between neighbouring images. Therefore, this inherent cross correlation of 3D holoscopic images can be seen as a type of self-similarity and can be exploited in order to improve the coding efficiency [89]. Conceptually, the technique captures and displays 3D images with a single aperture camera and a regular flat screen, which is customized with suitable overlaid array of micro lenses for display. The holoscopic imaging technique consists of two main processes, namely: recording and replaying as shown in Fig. 3.13.



(a) (b)

Figure 3.13: 3D Holoscopic imaging technique [89]

To record a 3D holoscopic image or video, a regularly spaced array of small lens lets, closely packed together, in contact with a recording device is used as depicted in Fig. 3.13 (a). Each lens contains the intensity and directional information of the corresponding image of the 3D object in 2D form. Furthermore, each lens let on the recording device views the scene from a slightly different viewpoint to its neighbour; as a result, a scene is captured from many different angles, thus the parallax information is recorded [89]. On the other hand, a simple flat panel display unit is utilized to replay the holoscopic images by placing a micro-lens array on top of the already captured intensity images that is illuminated by white light diffused from the background. Fig. 3.13 (b) demonstrates how the object is formed in space through the intersection of light rays originating from each lens lets. With the holoscopic technology, the light field that represents the original object can be reconstructed around the display panel. However, the camera setup in Fig. 3.13 does not consider depth control that causes the reconstructed object to appear in its original location in space, thus allowing only 3D virtual images to be produced. One major limitation of this camera setup is that objects that are far from a micro-lens array will suffer from poor spatial sampling of sensor pixels [89]. This problem is solved by adopting the camera setup in Fig. 3.14 by incorporating objective and relay lenses. The objective lens supports depth control, which allows the image plane to be near the micro-lens array. Conceptually, the 3D holoscopic image's spatial sampling is determined by the number of available lenses. This ensures that higher resolution images can be obtained by reducing the size of the lens. In addition, live images are recorded in this type of setup by a regular block pixel pattern. Usually, the planar intensity distribution representing a 3D holoscopic image consists of a 2D array of *M* x *M* micro images, due to the structural arrangement of the micro-lens array that is utilized during the image capturing [89].

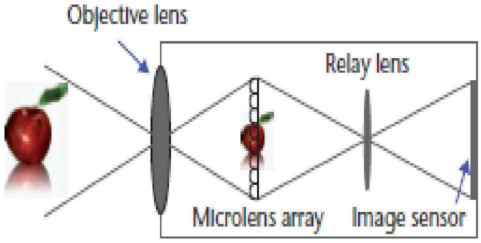


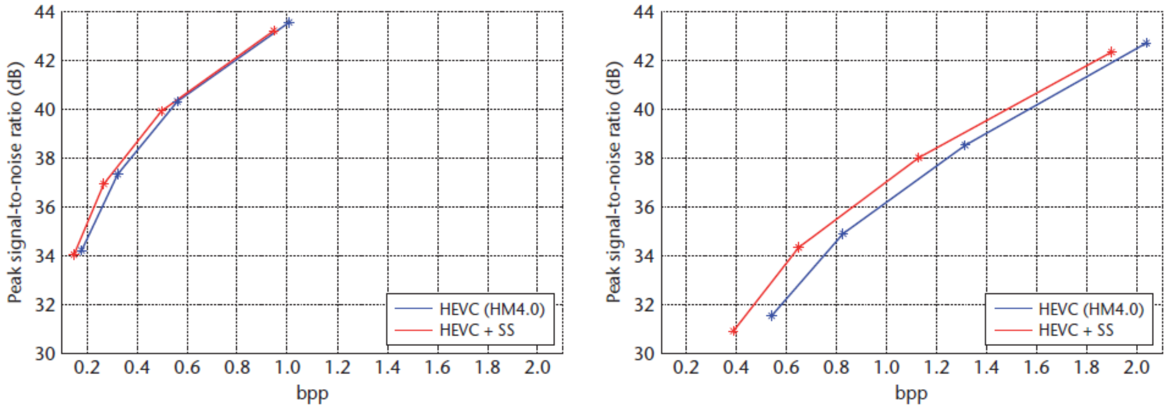
Figure 3.14: 3D holoscopic camera with objective and relay lenses [89]

In general, the 3D holoscopic imaging gives the users a true 3D viewing experience with less limitation than the stereo or multiview technologies. These limitations include eye strain, fatigue and, restricted view point. Just like any other digital video, 3D holoscopic video requires compression in order to make it suitable for use in storage or transmission applications. To this extent, the existing video coding standard such as the H.264/AVC is not very efficient for coding 3D holoscopic content because they are not designed to exploit the inherent spatial and temporal redundancies. The current video coding tools are often modified to encode the holoscopic video content, which is possible and achievable because of the huge correlations that can also be found in an integral video in terms of motion and disparity.

3.8.1. Concept of Self-Similarity Estimation and Compensation

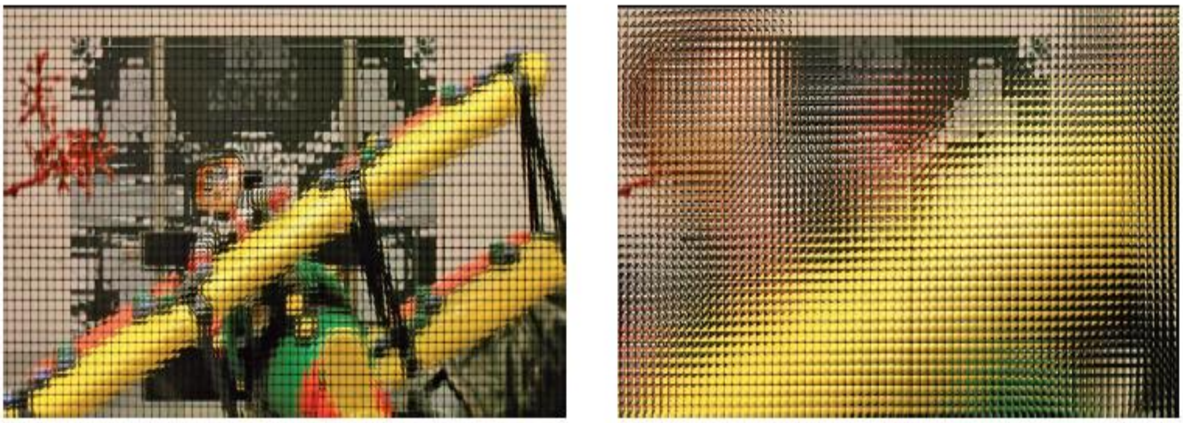
The self-similarity estimation process uses block matching criterion in order to find the best matching block for prediction within a given holoscopic image. However, in this scheme, the search region is restricted to the already coded and reconstructed area of the picture that is already encoded. The previously coded area in the holoscopic image forms the self-similarity reference, which is continuously updated as more area within the image gets encoded. This results in the chosen block to become the candidate predictor and the displacement between the two blocks is encoded as a vector. The vector generated is called self-similarity vector (SSV), which is similar to the motion vector in temporal predictive coding. Self-similarity prediction scheme is efficient in 3D holoscopic video content because knowledge of the precise structure of the underlying micro-lens array is not required, and consequently, the arrangement of the micro-images. On the other hand, in the self-similarity compensation block, the inverse quantized and inverse transformed prediction residual is added to the predictor to form the reconstructed 3D holoscopic video. The 3D holoscopic video information can also be stored like the H.264 video in the prediction memory for future predictions [90]. The concept of self-similarity estimation and compensation was first proposed in [91] in order to utilize the high self-similarity between neighbouring micro-images in a given holoscopic picture and to improve the coding performance of the H.264/AVC. The self-similarity spatial prediction mode for H.264/AVC is proposed in [91] which introduce a set of spatial prediction modes, in addition to the already existing intra prediction modes in H.264/AVC. Each additional mode defined as, INTRA_SS 16x16, INTRA_SS 16x8, INTRA_SS 8x16, and INTRA_SS 8x8 specifies a unique way to partition the MB in order to evaluate the self-similarity estimation and compensation [92].

For the INTRA_SS 8x8 mode, each 8x8 MB partition is further divided into 8x4, 4x8 or 4x4 sub-partitions for self-similarity compensation, which is specified in the compressed sequence with a sub-macroblock type syntax element. In addition to the new set of spatial prediction mode, a modified skipped MB mode which considers a 16x16 block size is also introduced as a candidate prediction mode. This mode is called the INTRA_SS SKIPPED in which the predicted self-similarity vector (PSSV) is directly selected without the use of self-similarity compensation. The PSSV is determined as the median vector of self-similarity vectors of three neighbouring blocks, in the same way as defined by the H/264/AVC standard for motion vector prediction [92]. Recent development and standardization of HEVC standard has demonstrated an improved coding efficiency in the H.264/AVC standard. However, HEVC does not currently utilize the inherent self-similarity that exists in the 3D holoscopic video content. Recent study in [89] [88] achieves high quality 3D holoscopic video with HEVC codec by incorporating the self-similarity algorithm into the HEVC coding tool. The SS algorithm is utilized to achieve inter prediction and to take advantage of the flexible inter prediction unit partition patterns in the 3D holoscopic video content. Furthermore, in [88] new spatial prediction modes are added to the existing HEVC's intra prediction modes to improve the coding performance of intra coded slices of the 3D holoscopic content. The authors in [89] demonstrated some quality improvement in the performance of HEVC for 3D holoscopic content by adapting the search range to the size of the prediction unit selected during self-similarity estimation. However, in their report, during self-similarity compensation, a single self-similarity vector is generated for each prediction unit and self-similarity skip prediction; only a SS vector is encoded and transmitted for each prediction unit. Their evaluations are based on the amount of self-similarity redundancy that exists per video frame. Fig. 3.15 and Fig. 3.16 represent the subjective and objective performance evaluation respectively.



(a) Frame with less SS redundancy (b) Frame with more SS redundancy

Figure 3.15: 3D Holoscopic quality improvement for plane and toy test sequence [89]



(a) Frame with less SS redundancy (b) Frame with more SS redundancy

Figure 3.16: 3D Holoscopic subjective view for plane and toy test sequence [89]

3.9. Error Resilience for 3D holoscopic video

Error control scheme in video transmission application is important and necessary for safe delivery of video content to the end users. Likewise, in order to ensure and guarantee the delivery of quality 3D holoscopic video over wireless network or the internet, efficient and reliable error control measures must be adopted in order to mitigate the effects of transmission losses in the channel. However, it is possible that the existing standard error control techniques such as error resilient and error concealment in H.264/AVC coding tool can be exploited in 3D holoscopic video content.

Perhaps, new techniques might have to be developed in the nearest future. This can be true, because the format and representation of the 3D holoscopic image or video generated from micro lenses give rise to an entirely different video content that is quite different from any video coding format and representation [93]. In this context, the development of an efficient and reliable error resilient technique for 3D holoscopic video should consider the video content and the characteristics of the transmission network. Error resilience for 3D holoscopic video content is an active research field, which is currently receiving attention from researchers. The 3D-SERVICIS is a research group that is established in June, 2014. The group is currently working on the scalable error resilient 3D holoscopic video coding for immersive systems and is expected to end in May; 2016. The main focus of the research includes the extension of scalable HEVC standard to 3D holoscopic content and to investigate the delivery channels of the 3D holoscopic content against errors/data losses. The project aimed at developing a new error control techniques that will be specific to 3D holoscopic content. Recently, the authors in [94] have also opened up a discussion on error resilience techniques in 3D holoscopic video coding. Conceptually, the study aimed at investigating the effects of transmission losses in a three layer display scalable 3D holoscopic video coding. In the report, the base layer represents a single view video, which can offer a 2D video of the 3D holoscopic content. Enhancement layer 1 and enhancement layer 2 is designed to support the stereo/multiview display and a full 3D holoscopic video content respectively. In this regard, the study proposes an error concealment scheme that is capable of estimating the missing data by exploiting the inherent redundancies that exist in the scalable 3D holoscopic video content. From analysis, the authors demonstrate that error concealment for H.264/AVC can be used to recover losses in 3D holoscopic video content. In some cases, the quality difference may be acceptable when compared to instances without losses.

3.10. 3D Video Display Systems

3D display system is simply a screen that can show 2D picture in 3 dimensions for more interesting and realistic viewing experience. The main approach to 3D display is based on stereoscopic viewing, exploiting the principle of binocular parallax [95] . Over the years a wide variety of techniques have been improved and new ones have evolved for both research and personal use. The adoption of 3-D displays is strongly driven by the ongoing digital multimedia revolution. While 3D imaging of previous decades relied on custom components and technologies far outside the mainstream, 3-D display devices in today’s age can take advantage of all-digital content handling chain that includes capture, processing, editing, and display [96]. Currently, 3D display systems provide new advantages to the end users; they are able to support an auto-stereoscopic, no-glass, 3D experience with significantly enhanced image quality [97]. Today, there are many 3D display system manufacturers with different brands already available in the market, most of these products are stereoscopic TVs with special glasses. Commonly available brands are Sony Bravia Smart 3D LED TV, Samsung 3D Plasma TV, LG Smart 3D LED TV and Panasonic Viera Smart 3D LED TV. Auto stereoscopic displays are not as common in the market as stereoscopic TVs and are very expensive to purchase, they come in brands of Toshiba, Sony, Alioscopy and Philips [98].

3.11. Stereoscopic displays with glasses

The 3D depth perception can be experienced through the use of filters or shutters to present to the viewer the left and right view information from a single display or screen. Glasses based systems are simple and have the advantage of being able to provide an entire audience with the 3D perception with an affordable cost [96]. These glasses are designed specifically to direct the left view to the left eye and vice-versa. They are classified as anaglyph; passive polarized or active shutter glasses [99].

3.11.1. Anaglyph

This is a simple and inexpensive approach to the 3D visualization problem and is applicable to common colour video equipment. Anaglyph 3D images can be generated when the images for the left and right eyes are combined together using a complementary colour coding algorithm [100]. Different colour pairs are used such as, amber blue and dark blue, red and green, red and blue, red and green, and red and cyan. The most commonly used colour pair is the red and cyan in which the red channel goes to the left eye and the cyan channel to the right eye.

Anaglyph images are usually created either by capturing the images with a binocular camera or by using the depth information of the object. The main disadvantage of this approach is the loss of colour information usually during view separation and the increased degree of cross talk which happens when a leakage occurs from one left view to right and vice-versa [100], which can result to eye fatigue.

3.11.2. Polarized glasses

Polarization multiplexing uses polarized light for image separation. The hardware configuration may consist of two monitors or projectors covered with linear or circular polarizing filters and are viewed with polarized glasses to maintain separate left and right eye views [101]. This type of display system allows viewers with polarised glasses to capture on each eye a light signal from one view only to produce a sensation of depth. One major setback in this approach is that images are distorted when the viewer's change their angle of view in linear polarization. However, the 3D perception in circular polarization is not affected by any change in angle by the user [102].

3.11.3. Active shutter glasses

The active shutter display is also known as time multiplexed technique. The left and right eye images are displayed alternatively on the screen at a high frame rate of about 120Hz in order to make the occlusion of the eyes unnoticeable. Viewers are required to wear battery powered active shutter glasses that are synchronized with the image displayed via an infrared emitter [100]. The disadvantages of this approach include the cost of the shutter glasses and the excess video bandwidth required compared to 2D.

3.11.4. Head mounted displays

The head mounted displays (HMD) consist of two LCD screens that are mounted in a glasses-like device and fixed relative to the wearer's eye position. Usually, the virtual world is portrayed by obtaining the users head orientation from a tracking system.

HMDs are binocular systems that can present the same image to both eyes, and offer a wide range of resolutions usually by trading off with field of view [103].

One of the notable advances of HMD is in the medical field where surgeon can have detailed computer generated information superimposed on the patient in real time [100].

3.11.5. Volumetric 3D display

The concept of volumetric display is still under development and investigation in the research community and industry. The volumetric 3D display can project 3D images directly into true 3D perception and does not require the use of special glasses for viewing. The range of applications is wide which include volumetric TV, display of dynamic scene, computer simulation and design, navigation, visualization of tomographic information in medicine, computer trainers and gamers, advertising and entertainment [104]. Volumetric displays generate volume filling three –dimensional image. Each volume element or voxel in a 3D scene is capable of emitting visible light from the region in which it appears [104]. This class of 3D display can provide a viewing angle of 360 degrees, and present imagery in true 3D space without the use of special glasses as with stereoscopic 3D displays [105].

3.11.6. Holographic 3D display

Holography is a sophisticated true-3D method. More research is needed to overcome technological difficulties associated with holographic displays [106]. Holographic displays can provide extremely realistic, high resolution and full colour images that actually float in the air, they are available in very large sizes. Moreover, 3D images that are generated by holographic displays are viewed directly on the recording material (photographic plate) from different angles without the use of special glasses [107]. This technology employs the principle of diffraction and propagation of waveforms. In holographic reproduction, light from an illumination source is diffracted by interference fringes on the holographic surface to reconstruct the light wave front of the original object [108].

3.11.7. Holoscopic 3D displays

In Holoscopic displays, the 3D scene is generated by collective pinhole/lens arrays that project view point-pixels in spatial direction where the intersection point of the two view point pixels creates a 3D pixel [109]. This concept is first proposed by G.M Lippmann in 1908. This concept allows users to experience the 3D depth perception with full motion parallax without the use of special glasses. Array of small spherical micro lenses known as fly’s eye lens array is used both in recording and displaying the 3D Holoscopic image. The design of the array is such that different images are visible depending on the viewing angle at the display [110]. Holoscopic imaging is also known as integral imaging concept and this unique property allows different parts of the image to be refocused after being taken and recorded [111].

3.11.8. Auto stereoscopic and Multiview 3D displays

Auto stereoscopic displays exploit the concept of stereoscopy to provide the 3D depth perception without requiring the viewer to use any form of special or viewing glasses. An auto stereoscopic system uses space division multiplexing to display 3D images by directing video signal to the left and right eyes through basically two types of multiplexing techniques, namely a parallax barrier or lenticular lenses [96]. The most reliable 3D display system currently available is the multiview 3D display which allows more views to be displayed in addition to motion parallax. Multiview 3D displays have the same principle as auto stereoscopic display in experiencing depth perception [98]. Multiview auto stereoscopic displays units are well advanced systems that deliver 3D perception without the need for eye wears. Technically, to ensure correct view separation to a viewer’s eyes, lenticular sheets or parallax barriers are placed strategically in front of the light emitters [112]. In parallax barrier, more than two images can be created in the viewing zone for the viewer and can give the viewer a sense of 3D perception without the use of special glasses. Motion parallax can as well be produced when there is angular shift in viewing [113]. The viewing zone in itself is determined by the position of sub pixels and the optical elements that control the direction of the light [114]. In parallax barrier systems, the use of LCD element is common in 3D displays because they offer good pixel position tolerances and high position stability, carefully controlled glass thickness and can combine successfully with other optical elements that are different [100].

3.11.8.1. Parallax Barriers

The concept of parallax barriers can be demonstrated as in Fig. 3.17. The function of the parallax barrier that is placed in front of the display is to direct the left and right views to the correct eye through occlusion and placing of stereo pair precisely in a grid. Some of the limitations of the technique include viewer restriction to 3D perception only within the viewing zone, out of which the viewer can perhaps view 2D image of the scene. Another problem with the technique is the visible dark areas caused by the barriers between pin holes that are placed to separate the view. In addition, the motion parallax produced by the barriers can cause motion sickness and eye fatigue when viewed out of the viewing region [98].

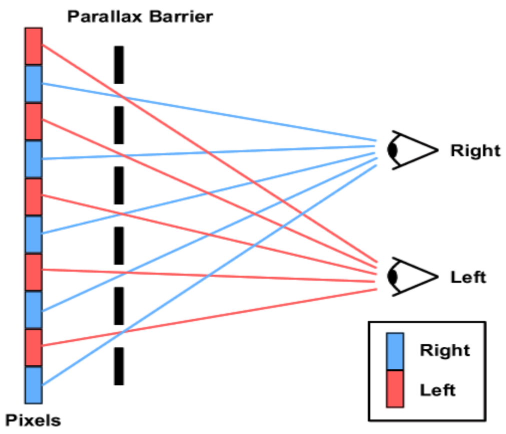


Figure 3.17: Auto stereoscopic parallax barrier display [115].

3.11.8.2. Lenticular lenses

Lenticular display system is very much similar to the parallax barrier which separates the left and right views through light occlusion while the lenticular lenses uses light diffraction to separate the views. The technique improves the brightness that the parallax barriers reduced to half through occlusion. The concept can be illustrated as in Fig. 3.18. Lenticular systems basically combine cylindrical lenses with flat panel displays to ensure that the diffused light from a pixel is only visible within a defined viewing angle in front of the display system [100]. Lenticular lens display has improved on the image quality, but the disadvantage includes overlapping of the LCD pixel sheet patterns with the lenticular lens pattern [116].

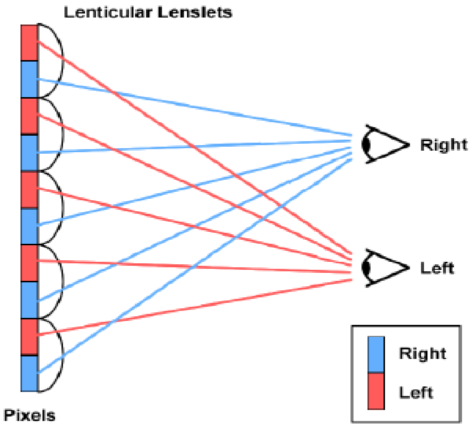


Figure 3.18: Auto stereoscopic lenticular lenses [115].

In practice, auto stereoscopic displays generally suffer from a number of limitations which include restriction in depth range as compared to glasses based stereo systems. Also, ghosting and cross talk reduces viewing comfort and resolution of each view can be limited [112]. Research is in progress towards the development of existing 3D display technology and more sophisticated designs are evolving to make 3D viewing experience more interesting and realistic. Also another available 3D display system is the magnetic 3D display [117]. It has a less complicated approach and cost less to produce than holographic, volumetric and integral imaging based displays [98].

3.12. 3D video content delivery

The growth of 3D video concept is becoming more available in homes and mobile devices through different media and platforms. This trend is expected to be widely adopted for different applications such as entertainment, medicine, industry, and so on. One important component of the 3D end-to-end system is the transport infrastructure which ensures a safe delivery of the 3D content to the end user. The delivery of 3D content/media through broadcast or on-demand to the end users with varying 3D display terminals (projector screen, TV, laptops, tablets and mobile devices) and bandwidths is one of the major challenges considered in order to deliver the 3D content to the homes and mobile devices [118]. There are basically two main platforms to deliver 3D video content: delivery over digital television (DTV) and Internet Protocol (IP) [41].

3.12.1. 3DTV Transmission

Three dimensions TV is an emerging technology that can allow users to view 3D TV live programs at the comfort of their homes. The concept is believed by the research and industry to be the next logical development towards a more natural and life-like visual home entertainment experience [119]. Advantage of the 3DTV broadcast is that the content can be delivered over existing HDTV infrastructures including viewers already with existing HDTV set top boxes. The concept can be used for cable, terrestrial and satellite broadcast and broadband channels. The 3DTV technology is still very active research field and there are a lots of projects proposals in 3DTV [120]. The most common and perhaps leading 3DTV developer is the DVB project. The transmission of 3DTV approach using Digital Video Broadcasting-Terrestrial (DVB-T) which is based on video-plus-depth representation was proposed by the European project, ATTEST [68].

In this concept, a 2D video stream is enriched with a depth map sequence. The video is coded and transmitted using the MPEG-2 standard through DVB while the depth map is encoded separately and transmitted as side information. At the receiver, the desired left and right views are reconstructed using depth map image rendering (DIBR). This concept provides easy adaptation with different 3D display systems, viewing conditions, and user preference [121]. There are other projects proposals on mobile 3DTV such as 3D-DMB and DVB-H. 3D-DMB is designed to deliver stereoscopic video and stereoscopic data content, which can provide users a 3D depth effect in a mobile environment. Mobile 3DTV over DVB-H is an extension of the terrestrial digital TV standard, DVB-T [122]. It is specifically developed to support the mobility and withstand problematic propagation in indoor and moving vehicle reception [121].

3.12.2. 3D Video on Demand

3D video on Demand ensures the delivery of video content over Internet Protocol (IP)-based networks, when requested by the user. One of the main problems of 3D video over the internet is the large amount of data to be delivered to the end user. The additional bitrate requirement of the additional view or depth information is still a challenge for 3D video content and this is currently an active research field which is important for the success of 3D video applications over the internet protocol (IP) networks [123]. 3D video can be delivered over different types of protocols that are available in today's internet which include RTP, UDP, and HTTP. Video on demand has become one of the most popular applications on the internet, and 3D video on demand is expected to be more popular in the near future. 3D video services and in particular those delivered through the IP and mobile channels face a number of challenges due to the need to properly handle a large volume of data and the possible limitations due to the condition of the transmission channel and the device [124]. Transmission of multiview video over IP is perhaps the most flexible and reliable solution for 3D content delivery, that can allow different transmission rates to different users based on the level of connection rate and display systems. In addition to that, MVV over IP can be considered to be stand-alone service or as a supplement to broadcast of stereoscopic video over DVB [125]. Currently in the United Kingdom (UK), Virgin Media TV customers, with a V HD set top box will be able to view and experience all 3D on demand contents from their homes over IP. An anticipated problem with 3D VoD is likely to arise, especially when more views are required to support multi view displays, this will clearly demand more network bandwidth.

3.13. Challenges for 3D Technology

The creation of 3D video content and its distribution is a multistage process that follows a lifecycle starting from content acquisition at the source, through production and packaging of the content, to distribution and finally optimal representation of the 3D content to the end users. Each component of the 3D technology is an active research field both in the research community and the industry. Problems and challenges regarding 3D video acquisition include temporal synchronization, geometrical calibration and colour balance between individual cameras. The captured 3D content for coding and display purposes needs to be converted from the production into the transport format by post processing. Formats such as video only formats will require minor adjustments in colour correction, subsampling or colour format conversion, while for depth enhanced format, complex algorithms [126] are necessary for depth rectification and enhancement [58]. These algorithms have high computational cost and are prone to error. A natural way to improve 3D multiview video content is to fully and efficiently utilize the correlations between frames and views. The depth enhanced representation formats are designed to further reduce the amount of multiview video data and transmission rate, view synthesis in itself is a very complicated task to achieve in addition to the presence of noise in the raw depth data [127]. Another issue in 3D video coding is that huge amounts of additional decoded pictures require to be buffered. When the number of views or auxiliary information becomes large, the required memory buffer may be prohibitive [34]. In general, the development of advanced 3D video coding algorithms needs to optimize the performance of both video and geometry distortions and in consideration of the level of quality of the virtual views rendered [58]. Delivering 3D video over networks is more challenging than monoscopic video. The 3D video content requires more bandwidth to deliver more views than the conventional 2D video. The encoded 3D video data consists of more dependency as a result of interview exploitation and synthesis prediction during source compression; thus, the existing 2D video transmission techniques cannot be employed directly to these advanced 3D representation formats [61]. In addition, the delivery medium is based on the best effort network and standardized guidelines are yet to be established. The 3D video content representation formats have been studied mainly for higher compression efficiency and resilience to transmission errors. However, there is a clear trend toward the development of 3D video delivery technologies for home users that can ensure higher quality and performance than the already available monoscopic techniques [128].

In the display aspect of 3D video, many different approaches are adopted to realise 3D free viewing experience. Auto stereoscopic 3D display system is one of many approaches that are cost effective and can deliver 3D depth perception. However, these auto stereoscopic 3D displays are not truly spatial displays because they exclude vertical parallax and depend upon the brain to fuse the two distinct images to create the 3D perception causing eye fatigue [129]. Also in addition to that, these display systems have a restricted viewing region outside which the 3D perception can be lost.

3.14. Quality assessment for 3D video

Currently, many 3D research projects exploit the 2D objective metric techniques available to measure and assess the quality of 3D video using objective quality measure and subjective quality measure. The most commonly used objective metric that assumes the correlation with the Human Visual System is the Peak Signal to Noise Ratio (equation 3.1), which compares the maximum possible signal energy to that of the noise energy [130].

$$\boldsymbol{PSNR} = \mathbf{10log_{10}}\left[\frac{\mathbf{255^2}}{\mathbf{MSE}}\right] \qquad (3.1)$$

MSE is defined as the Mean Squared Error between the original and the reconstructed video sequence. Usually in this modelling, the average value is computed from the PSNR values of all the frames in the video across all the views used. The different types of objective metrics are well detailed and documented in [132]. One major setback with objective metric and in particular the PSNR metric is that it has limited approximate relationship with the distortion or quality perceived by the Human Visual System [133]. For this reason, the second video quality evaluation method is the subjective test. The subjective quality metric considers the subject’s perception and opinion on a particular reconstructed video. In the subjective analysis, subjects are presented with the reconstructed single video sequence of each of the views for their assessment [134]. The approach for subjective and evaluation test is described in detail by ITU in [135] [136]. The human quality impression is usually presented on a scale of 1 to 5, where 1 represents worst and 5 represents best [131] as in table 5.2. This methodology is known as Mean Opinion Score (MOS).

Table 3.2: ITU-R quality and impairment scale

Scale	Quality	Impairment
5	Excellent	Imperceptible
4	Good	Perceptible, but not annoying
3	Fair	Slightly annoying
2	Poor	Annoying
1	Bad	Very annoying

Another quality metric approach is to calculate the MOS first, then calculate the percentage of frames with MOS worse than that of the original video. This method is reported to have the advantage of clearly showing the impairment caused by the erroneous channel [131]. This approach can be illustrated as in table. 5.3.

Table 3.3: Possible PSNR to MOS conversion [131]**.**

PSNR (dB)	MOS
>37	5 (Excellent)
31 – 37	4 (Good)
25 – 31	3 (Fair)
20 -25	2 (Poor)
< 20	1 (Bad)

The 3D immersive video quality evaluation is currently an open research area among researchers and developers because of its complex nature and also lack of reliable objective quality metric for 3D video [124]. However, 3D depth perception and quality of experience (QoE) can be analysed from the following features: Overall image quality, depth perception, naturalness, sense presence, and visual comfort. Also, the main sources of quality degradation in 3D QoE include display technology, viewing conditions, view synthesis, transmission and packet loss, and spatio-temporal video scalability [137].

Because 3D video perception is a multi-dimensional concept, the objective metrics for monoscopic video like the PSNR cannot assess information about the depth perception of 3D videos [138]. The limitations involved in measuring 3D video perception with PSNR have been outlined and demonstrated by Video Quality Experts Group (VQEG) [139]. Therefore, it has become necessary to devise a means and methodology to accurately evaluate the quality perception of 3D video. However, some few standards have defined subjective evaluation methodologies for 3D video, such as International Telecommunication Union Radio-Communication Standard Section (ITU-R) BT1438, ITU-R BT500-12, and ITU Telecommunication Standards Sector (ITU-T) P.910. These procedures are not good enough for measuring the 3D video perception and have some limitations such as inability to measure combined effects of perceptual attributes [124]. The work in [140] proposes a subjective quality evaluation on the effects of camera distance and JPEG coding. The study considers the impact of these factors and the effect on overall image quality, perceived depth, perceived sharpness, and perceived eye strain. The approach has a limited practical application because of some setbacks in handling compression artefacts. Generally, subjective quality evaluation is intensive and time consuming because it requires other resources such as human observers, controlled test environment and so on, and may be practically impossible to implement in a live scenario. On the other hand, 3D video objective quality evaluations are emerging to offer correct results that are comparable to the quality scores achieved by subjective tests [141]. It is a common practice to evaluate the performance of objective quality metric based on approximate values obtained from subjective quality assessments. Usually, subjective quality metric are designed to account for both depth map and texture related artefacts based on the extracted image features. Furthermore, it is important for the final score obtained to reflect the quality degradation in terms of the 2D image and depth perception and to have a reliable ground truth dataset to evaluate and verify the performance of the 3D objective metrics [124]. Currently, the recent development in auto stereoscopic displays has created an upward growth in the design and standardization of 3DTV content and quality measurement [142]. It is already known that 3DTV format considers the transmission of one colour video with an auxiliary per-pixel depth map in order to create the 3D depth perception.

However, a standard single view video quality metrics (VQM) can be used to measure the quality of the video plus depth content. This is achieved through quality measurement of the virtual views that are rendered from the distorted texture and depth sequences. Usually, the original reference sequence is required for the full reference VQM by rendering the virtual views from the original colour and depth maps. In this context, the authors in [142] present a study that can optimize the visual quality of 3D encoded video with depth representation format. In the analysis, the authors tested various bit rates for both the colour video and depth information and measured the quality by virtual view quality metrics as depicted in Fig 3.19.

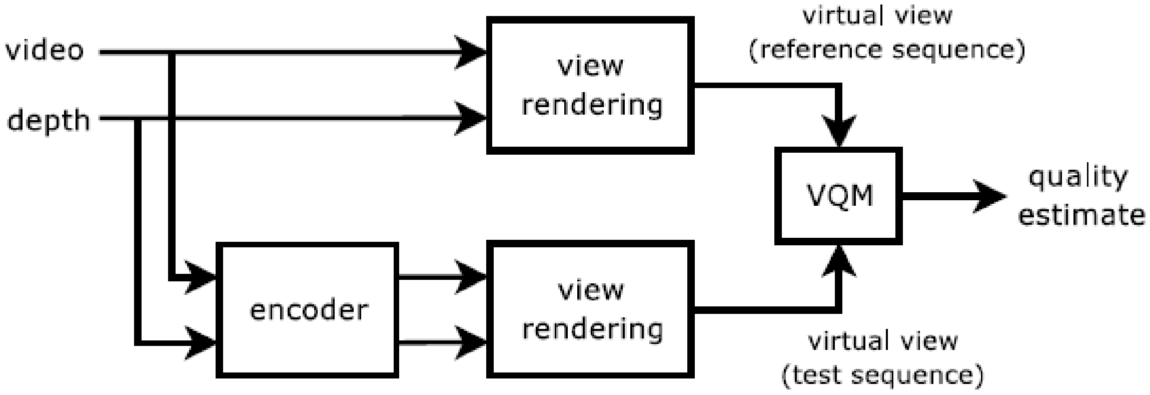


Figure 3.19: Quality metric for virtual view video [142]

In order to render the virtual views, the objects in colour image are shifted to different positions that can allow viewing when looking from a virtual camera that is equivalent to the real one. Since the new positions of the objects in the colour image are computed from the depth map, the video data from the depth and colour sequences are both fused together by way of virtual view synthesis. Hence, the measured quality in terms of Structural Similarity metric (SSIM) of the virtual view is assumed to represent the quality of the textual and depth sequences. Further analysis in the study considers small-scale subjective testing to evaluate the performance of the objective metric and the perceptual quality in the video plus depth content. It is demonstrated that the highest perceptual quality can be achieved when 15–20% of the overall bit budget is used to code the depth map sequence [142].

3.15. Conclusions

This chapter reviewed the 3D video coding concepts and formats from the basic techniques like the stereo video coding to the current and state of the art standards such as depth enhanced coding. The chapter also presented some results reported by recent study in 3D video coding and representation which indicate current development and improvement in terms of coding efficiency and picture quality among the various techniques. It can be noticed that, 3D video coding is still emerging with better perceptual quality at the expense of increased complexity in the 3D codec. However, the most common 3D representation and format is the depth enhanced coding technique. The depth enhanced coding system takes the advantage of depth information to create a 3D perception and efficiently reduce the amount of bitrate required for transmission against the multiview video systems. Other essential components of the 3D system that are dealt with in this chapter include 3D video content delivery, 3D video quality assessment and challenges involved in the development of the 3D video systems and technology.

4. Chapter Four: Error Resilience and Concealment for MVC Bitstream

4.1. Introduction

The delivery of compressed video stream to end users remains a challenge in video communication due to problems that include huge amount of data involved, diverse network characteristics, user terminal requirements, as well as user‘s context such as their preference and location. The current available bandwidth for wireless networks may not be able to accommodate the growing demands of delivering quality video to fixed or mobile users, especially for applications that require additional number of views [143]. Compressed multimedia data streams such as 3D video signal are even more sensitive and can be severely affected by bit errors when transmitted over error prone communication networks. This is because of the use of entropy coding technique with increased dependency used to generate the resulting bitstream [144]. Errors in the compressed video can cause loss in bitstream synchronization and severe error propagation. In order to minimize this effect, it is important and necessary to detect quickly that an error has occurred in the bitstream. However, a standard compliant video decoder can detect a syntactic error, such as illegal codeword, or a semantic error, such as reference frame loss [133]. Therefore, to mitigate transmission error or losses in any communication system, it is a basic requirement to consider the three main categories of video communication for an efficient error control strategy namely: error resilience, channel coding and error concealment. Some major standard error resilient tools [145] in H.264/AVC such as data partitioning, slice structuring, intra refresh, redundant slice and so on, shall be discussed later in this chapter. Channel coding such as error correction techniques can be deployed in the channel in order to protect the video bitstream from transmission errors and to correct any bit error that may have occurred. Redundant bits are also added to the bitstream to help detect and correct bit errors in this scheme. It is a common practice in the design of a reliable video communication to consider the trade-off between compression efficiency and perceived quality [146]. Thus, while compression schemes and source coding aim at removing various redundancies for higher compression efficiency, error resilience and channel coding introduce some redundancies in the video data to achieve high quality rate [147].

Therefore, it is necessary to maintain a balance between compression efficiency and video quality which may vary differently according to design applications. Error concealment, on the other hand, is a decoder based technique that utilizes spatial and temporal redundancies of the received video data in order to conceal the losses from the viewer [148]. The H.264/AVC error concealment in MVC shall be reviewed in details later in this chapter.

4.2. Challenges and Approaches

This section will be discussing the motivations behind the adoption of error resilient tools and the challenges they are faced with. There are several types of error resilient techniques employed by H.264 video coding standard; each one has a unique approach of trying to combat the effects of transmission error on a compressed video sequence. Error resilient techniques at the source coding level have proved significantly to solve the problem of transmission error in a bitstream. However, error resilient techniques have a common problem and challenge they face at source level especially when dealing with a wide range of error conditions such as random bit error, burst error, or packet loss error. In a video bitstream, a single bitstream error may entirely invalidate the video bitstream and render it useless because in reality most video coding techniques are not designed to be error robust. For instance, in H.264 video coding, the use of predictive and entropy coding techniques can provide high encoding efficiency to the video sequence but are not capable of withstanding errors. A more challenging aspect of a video bitstream especially for multiview video bitstream is that error in the bitstream is likely to propagate to subsequent frames and may be across the other views. This may also affect the visual quality of the reconstructed multiview video. A straightforward solution is to retransmit the affected video data by way of automatic repeat request (ARQ) technique. However, this may not be an appropriate and practical solution for real time applications such as conversational and streaming services because of their delay constraint; hence a more efficient technique is required [149]. In general, it is important for error resilient techniques to be designed in such a way that they can offer basic error robustness to the bitstream. It is also important that the design of ER techniques should consider a number of constraints such as implementation in the source encoder, in order not to comprise the coding efficiency, and should have a minimum computational cost and complexity. In addition, they should be flexible enough to adapt to the network environment [150].

4.3. Standard Error Resilience tools in H.264/AVC

The H.264/AVC standard specifies several error resilience schemes to minimize the effects of transmission errors on the perceptual quality of the reconstructed video sequences. However, these techniques assume a packet loss scenario where the receiver discards and conceals all the video information confined within a corrupted NALU packet. This implies that the type of error resilient techniques employed by the standard operates at the bitstream level, since not all the information contained within the corrupted NALU packet is un-utilizable [145]. Just like the previous video coding standards, the operation of the decoder is specified for error free bitstream as well as syntax and semantics of the bitstream in the H.264/AVC standard. In the following subsections, we shall review the standard error resilience tools in H.264/AVC including the data partitioning, slice structuring, redundant slice, intra refresh, reference frame selection, and switching pictures. Candidate error resilient tools for multiview video coding include data partitioning, slice structuring (flexible macroblock ordering and slice interleaving), redundant slice (multiple description coding) and intra refresh coding. However, none of these tools is implemented in the JMVC reference software for MVC extension [151]. Some research proposals are reported regarding error resilience for 3D multiview video coding. These techniques will be presented later in the chapter. Note that, most of the error resilience and concealment proposals in MVC are extension of the techniques in H.264/AVC.

4.3.1. Data Partitioning

It is a common practice in video coding systems to code all symbols of a macroblock together in a single bit string that forms a slice [152]. DP, nonetheless, creates more than one bit string (partitions) in every slice, and rearranges all symbols of a slice into a separate partition that have a close semantic relationship with each other. DP as an error resilient technique in H.264/AVC video coding has the least overhead as experimented in [153] and in addition, it is suitable for transmission over high error prone wireless networks. The concept of H.264/AVC is designed such that, it separates the Video Coding Layer (VCL) from the Network Abstraction Layer (NAL). The VCL specifies the core compression layer feature by efficiently representing the content of the video data while the NAL provides header information to support delivery over various types of networks [43]. This feature of the standard facilitates easier packetization and improved video delivery.

All data is contained in NAL units, each of which contains an integer number of bytes. H.264/AVC is designed in a way that when DP is enabled, each slice of the coded bitstream divides into three separate partitions with each of the partitions being from either type A, type B or type C partition. Type A partition consists of Header information, Quantization Parameters (QP), Macroblock (MB) type, reference indices, and motion vectors. This information is the most important, because without it, symbols of the other partitions cannot be used. The intra partition also called type B partition, consists of the Discrete Cosine Transform (DCT) intra coded coefficients, and the inter partition also known as type C partition, contains DCT coefficients of motion compensated inter frame coded MBs. Type C partition in many cases is the biggest partition of a coded slice and on the other side the least in importance. This can be illustrated in Fig. 4.1 as reported in [43]. The use of Type C requires the availability of Type A partition and not Type B partition to be decoded [152].

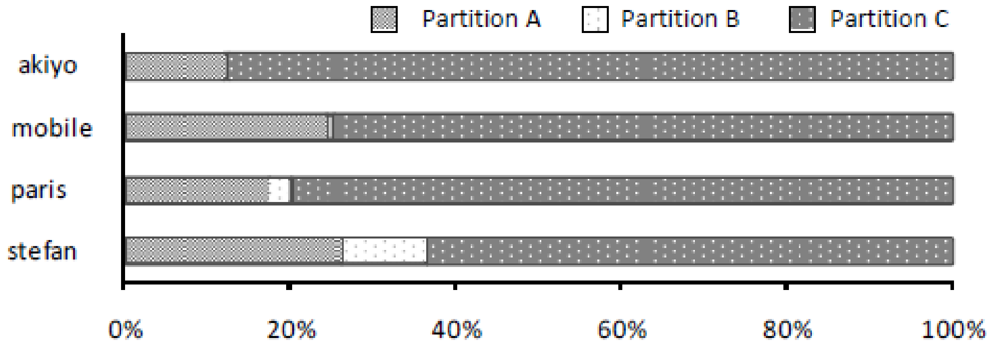


Figure 4.1: Percentage of data for partitions A, B and C in different test sequences [43]

Data partitioning in H.264/AVC can be illustrated as in Fig. 5.1 below, where a single slice is split into three NAL units.

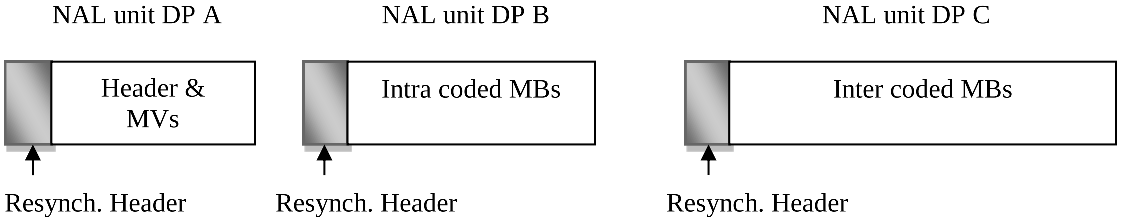


Figure 4.2: H.264/AVC Data Partitioning concept.

The three data partitions are packetized as individual and separate NAL units. This unique arrangement allows a video slice to be reconstructed even if the residual data is lost, provided that the header and motion information remains intact [154]. In the decoding process of a partitioned video slice, type A partition is independent of both type B and type C partitions, but not vice-versa. Type A partition with constraint intra prediction enabled makes the decoding of type B partition independent of type C partition. However, at the time of writing this thesis, no work is reported in making the decoding of type C partition also independent of type B partition [155]. For video streams, most modern formats have an optional data partitioning mode to improve error resilience by employing Forward Error Correction (FEC). FEC is also an error control technique against packet loss over packet switched networks [156]. Error correction technique can ideally be combined with data partitioning technique in order to provide high protection to the high priority video data such as the header and MV information. The concept is based on the application or physical layer in a technique commonly known as Unequal Error Protection (UEP) [157]. Strategies for improving quality performance in error prone environment include the application of FEC to Partition A and perhaps Partition B, or transporting the partition types over different channels, selecting the most reliable channel for partition A [13]. However, amongst the different standard error resilience techniques (at source coding level), data partitioning technique is found to be more efficient in terms of redundant bits required in order to achieve error robustness. The authors of [153] report some experimental results in terms of additional overhead for four common error resilience tools in H.264, which includes intra-coded macroblock refresh, data partitioning, flexible macroblock ordering and, slice structuring. However, Fig. 4.3 shows that data partitioning has the least overhead in terms of performance evaluation. The horizontal axis represents the mean bitstream rate arrived at by setting the quantization parameter (QP) to the given value while the vertical axis represents the mean overhead rate with that QP. As the quality decreases i.e. when QP goes higher, the advantage of data partitioning increases as the relative overhead of all schemes increases.

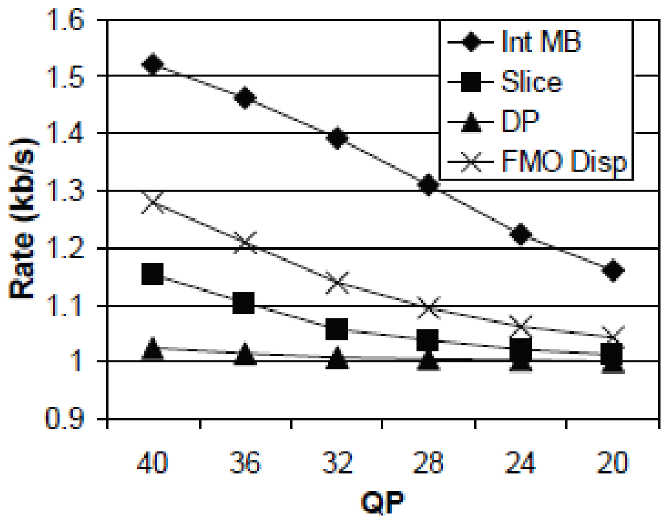


Figure 4.3: Bit rate performance for different ER schemes in H.264 [43]

When a partition is lost or dropped due to transmission losses or error, the reconstructed video is always affected differently. An experimental study in [43] proposes a scheme based on selective dropping of packets belonging to different partitions. After simple analysis of the NAL unit headers to identify the data partition each packet belongs to, the authors considers dropping video packets based on a defined criterion. That is, by dropping header information of type A only, dropping intra coded coefficients of type B only, dropping inter coded coefficients of type C only and dropping any packet at random. The result obtained in Fig. 4.4 shows that dropping packets from the data partition A can cause a severe degradation of more than 3dB in terms of PSNR measure when compared to random dropping of packets in the bitstream.

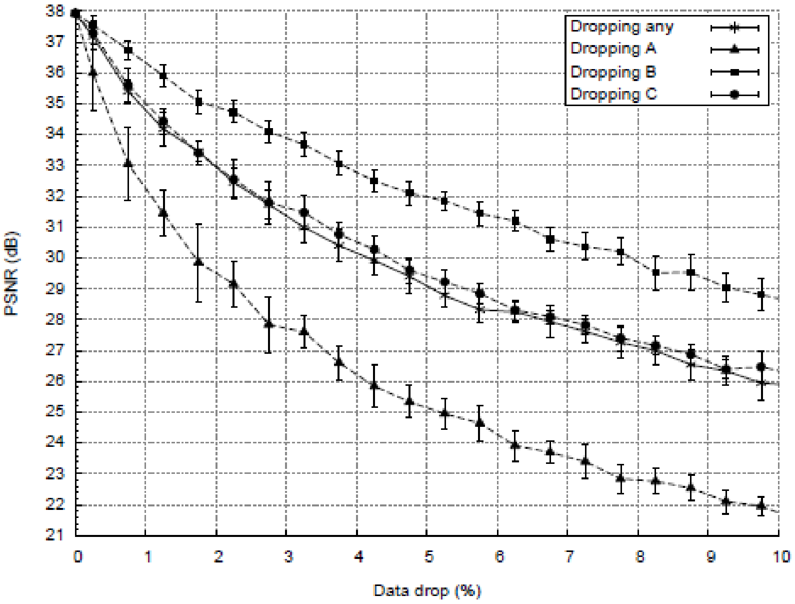


Figure 4.4: The Effects of dropping partitions in H.264/AVC Paris test sequence [43]

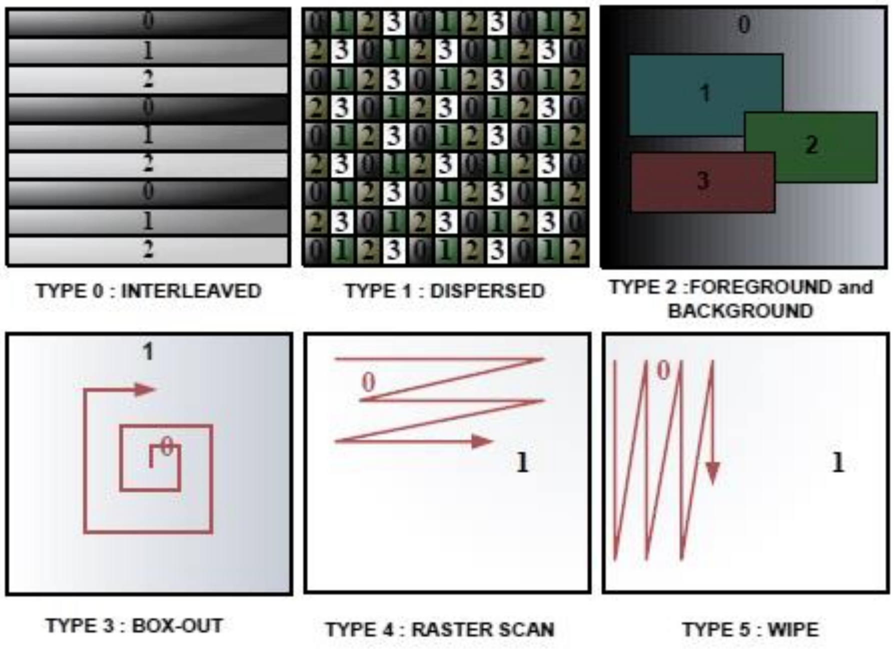
The reason for the severe degradation when A is dropped is because of the loss of header information that is present in data partition A. This information is required and necessary to reconstruct the frame at the decoder with good quality. Also by dropping packets from data partition B only, result in quality improvement in comparison with when packets are dropped randomly. On the other hand, dropping from only data partitioning C results to similar effect as random packets dropping. This is because data partition C is the largest partition out of the three partitions and when packets are dropped randomly, it is very likely that packets belonging to partition C will be dropped. At the time of writing this thesis, the author is not aware of a similar study or work that reported data partitioning technique for multiview view. The previous work in [158] initiated an approach in MVC based on H.264/AVC data partitioning technique. Even though, a full performance and evaluation of the technique could not be derived, however, the implementation process and architecture is presented.

4.3.2. Slice Structuring

Transmission of video content over wireless networks is generally provided through frames with small maximum transmission units (MTU). This is mainly because larger packets are more affected by bit errors than smaller packets [145]. As mentioned earlier, in error prone wireless transmission channel, a single bit error can cause a loss of an entire frame due to the de-resynchronization between the encoder and the decoder at the receiver. This problem and the effect of error propagation can be mitigated in a video frame by adopting the concept of slice coding, where a single frame is sub-divided into several other slices. In this technique, if an error occurs in any given slice of a frame, the affected slice is dropped and the decoder tries to find the next slice header to continue decoding. It is recommended to use slice coding especially for applications that require video transmission over error prone channels [159].

Furthermore, the coding of slice is achieved by a limited spatial prediction which can provide spatial distinct synchronization points [160] and the encoder can select the location of any synchronization point at any macroblock boundary. Slice structuring is basically a means to limit error propagation from a corrupted packet to subsequent packets. Flexible macroblock ordering (FMO) is a type of slice structuring error resilience mechanism that allows the scattering of possible errors around the whole frame as equally as possible in order to avoid error accumulation in limited region. It is a common understanding that as the distance between a corrupted block and the nearest error-free blocks increases, the distortion in recovered blocks also increases [160]. In FMO, macroblocks are mapped into slice groups and a slice group may contain several other slices. In this arrangement, it can be possible to achieve a flexible and efficient transmission of macroblocks out of raster scan ordering. The checker board/scatter macroblocks allocation is found to be very effective in conjunction with appropriate error concealment scheme [161]. This ordering enhances the concealment of lost blocks by their neighbouring blocks because images have smooth surfaces at block boundaries [162]. In FMO, when only a single slice group exists within a picture, it is considered as the case where no FMO is used at all. There are seven different types of FMO that are usually labelled Type 0 to type 6. The type 6 label is the most random one that gives full flexibility to the user [163]. The type 0 to type 5 is all represented in a specific pattern which can be exploited when transmitting the MBA_Map. Fig. 4.5 shows the different FMO patterns. FMO type 0 divides the frame into a number of different slices of slice group.

Usually, when the number of slice group increases, the number of MBs surrounding each MB also increases. FMO type 1 is commonly known as the dispersed slices; it uses a common function that is both known by the encoder and the decoder in order to spread the macroblocks. The more slice groups are used, the more each MB is surrounded by MBs of different groups. FMO type 2 is used to mark rectangular areas known as region of interest with a frame. The MBA_Map can be stored using top left or bottom right coordinates of those rectangles. FMO types 3 to 5 are dynamic in nature which allows the slice groups to grow and shrink over different pictures in a cyclic way. If a slice group is lost during transmission, reconstructing the missing blocks is simple with the support of neighbouring information from the MBs. However, the concept of FMO reduces the coding efficiency in a bitstream and has a high overhead cost in the form of MBA_Map which has to be transmitted with the bitstream [164].


TYPE 0 : INTERLEAVED

TYPE 1 : DISPERSED

TYPE 2 :FOREGROUND and BACKGROUND

TYPE 3 : BOX-OUT

TYPE 4 : RASTER SCAN

TYPE 5 : WIPE

Figure 4.5: Different types of Flexible Macroblock Ordering [165]

It has been reported experimentally in [152] that FMO with dispersed map allocation can keep the visual effects of losses so low in video conferencing applications with CIF-sized pictures and a loss rate of up to 10% that only a trained eye can identify the quality loss. Normally, in this technique, blocks with the same colour pixels are grouped in a single slice.

When FMO is enabled, a macroblock map (MBA_Map) is added to the parameter set to signal the arrangement of the macroblocks. The MBA_Map is structured information that maps the individual spatial address of a macroblock to a slice group in a raster scan order. Normally, when a slice is corrupted the decoder can conceal the lost region by using the upper and lower MB rows corresponding to a different slice group. In checker board mode, the prediction between neighbouring macroblocks is cut off in order to avoid error propagation from one slice to another [160]. One disadvantage of this scheme is the additional overhead bits which affect the coding efficiency. Slice structuring technique has been extended to MVC and is reported in some papers. The work in [166] combined FMO and Luby Transform codes as an error correction technique in order to improve the performance of the MVV bitstream over packet erasure network. The work in [167] proposed the use of dispersed FMO and error concealment technique for 3D stereo video over wireless mobile networks. While the work in [168] proposed the dispersed FMO with three slice groups with motion compensation concealment to combat the effect of transmission errors in multiview video coding. The authors in [166] proposed the concept of slice group in MVC, which combines error resilient and error control technique. In their technique, specifically, the dynamic formation of the Macroblock Allocation Map (MBAMap) using flexible macroblock ordering is considered to achieve high quality performance over packet erasure networks. The proposed technique is evaluated with non FMO coding over a packet erasure network of 10% packet error rate (PER) with race1 test sequence. The performance gain of more than 1dB is achieved for different transmission rates in Fig. 4.6.

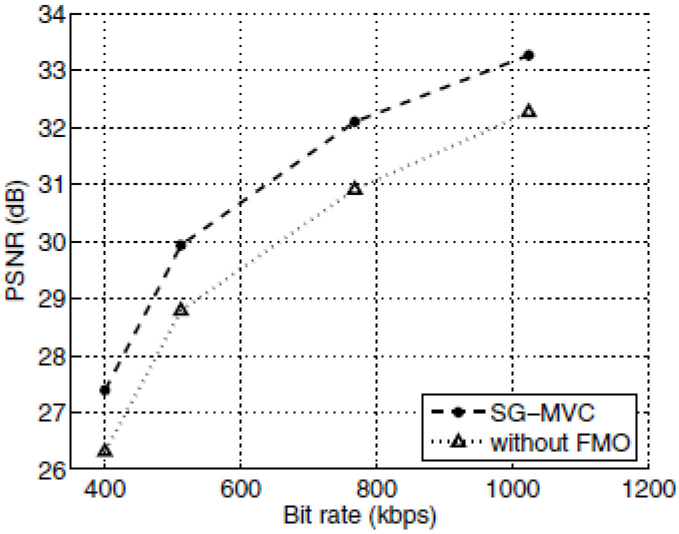


Figure 4.6: Quality performance over packet erasure network with 10% PER [166]

The result obtained illustrates that the use of FMO error resilient scheme can improve the quality of multiview video over a high error rate erasure network when compared with when FMO is not in use. The authors report that the enhanced performance of the proposed technique is attributed to efficient macroblock classification, which increases the performance of the channel protection and error concealment scheme as a whole.

4.3.3. Slice Interleaving

The coded video sequence in H.264/AVC consists of a sequence of coded pictures that can represent a frame or a field. The macroblocks of each frame are systematically organized into slices that are sub-grouped within a given picture. The number of MBs assigned to each slice does not have to be the same in a particular picture. The process by which H.264 video coding standard segments a picture into slices is usually referred to as slice coding. A video slice consists of an integer number of MBs of a one picture that can range from a single MB per slice to all MBs of a picture per slice [164]. Each slice in the picture carries all the information necessary to decode the MBs contained within it. The segmentation of a picture into different slices enhances the adaptation of the coded slice size into different maximum transmission unit (MTU) sizes. The concept in [169] aims to mitigate error propagation from a corrupted packet to subsequent packets in a video stream. Among the various error resilience technique employed by H.264, slice coding is a different approach that can subdivide each picture into one or more slices with increased level of importance and independence from other neighbouring slices. Thus, errors or lost video data from one slice cannot affect or propagate to any other slice within the picture [170]. Slices are independently decoded if the previously decoded frames are available in the decoder. This is achieved by utilizing the location information that is present in the slice header and by allowing spatial dependency only within the slice. For application that considers high compression efficiency, the use of one slice per frame is recommended in order to avoid header overhead [171]. In video communication, if the NAL unit size is bigger than the MTU unit of the corresponding transport layer, it will be fragmented into smaller packets. In error prone environments, some of these smaller packets can be lost which can lead to entire loss of frame because the decoder is not capable of decoding only part of the NAL unit. However, by encoding a frame into several other slices so that each individual slice is smaller than MTU can allow each packet to arrive at the decoder and can be correctly decoded [171].

4.3.4. Redundant Slices

Redundant coded slice is a new error resilience feature that is included in the recent standard of H.264/AVC. The technique enhances the robustness of video transmission in a packet loss network [172]. Multiple description coding is an example of redundant slice error resilient technique for video transmission over unreliable and non-prioritized network. It has the capability to combat packet loss without having to retransmit any corrupted slice, thus satisfying the requirement for real time applications and reducing the network congestion [173]. In this technique, the H.264/AVC encoder can send redundant representations of various regions of a picture that may not necessarily be used in the decoding process if the corresponding primary coded picture is received correctly. In this case the redundant coded slice is simply discarded by the decoder. However, if the primary coded picture is corrupted, the redundant slice is used to limit the visual degradation of the picture caused by transmission losses [174]. The key objective of MDC is the representation of the video data in more than one description so that high quality can be achieved. When all the descriptions are received and reconstructed successfully, then a high quality is achieved. However, when a description is lost in the process of transmission [175] the resulting visual quality should degrade gracefully. One of the most widely used methods of generating multiple descriptions are based on the pioneering MD scalar quantizer (MDSQ) proposed in [176]. The minimum distortion (central distortion) is achieved when all the descriptions are received [177]. The concept of MD with two descriptions is depicted in Fig. 4.7. The most common MDC models refer to two descriptions with rates defined as R1 and R2 respectively, which are transmitted over two lossy channels. The performance of the technique for two descriptions is evaluated in terms of the central distortion D_0, and the two-sided distortions D_1, D_2 for a condition where either of the two descriptions is received, as functions of the total bitrate $R_1 + R_{2,}$ where $R_{1,}R_2$ are the bitrates devoted to the encoding of either description. In the case of balanced descriptions, as it is always assumed in the following assumption that, $D_1 = D_2$ and $R_1 = R_2$. The same quality produced by the central decoder can be the same quality obtained by the reference single description coding scheme with a rate $R_1 + R_2 - \Delta R$, ΔR is the extra rate introduced by the MD scheme as an overhead to accommodate multiple quality levels [177].

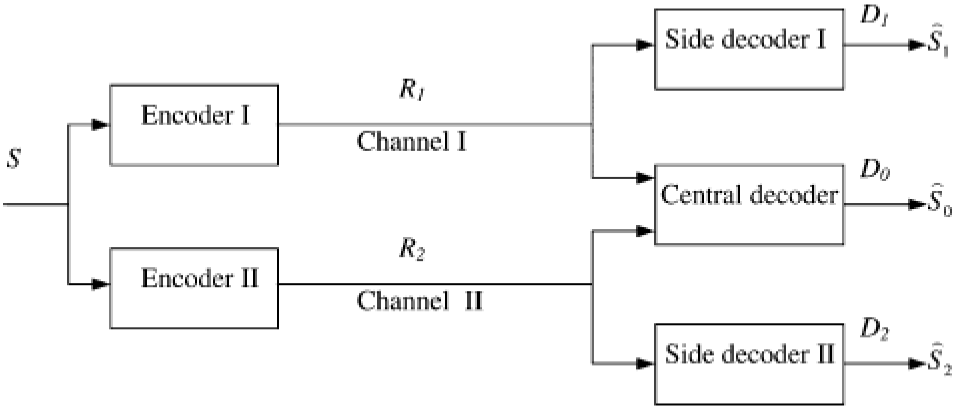


Figure 4.7: Multiple Description codec with two descriptions [177]

In MDC, when redundant slice is used to replace the missing primary slice during decoding, error is introduced in the decoder prediction loop because of a mismatch between the primary slice and the redundant representations. In this context, the authors in [177] proposed an algorithm in MDC scheme (Fig. 4.8) that can efficiently adjust the coding redundancy in order to mitigate the decoder drift when the redundant slices are used during transmission losses. It is important to notice that each redundant P-slice is predicted from the previously coded primary slices, which means the redundant slices cannot be used for prediction in the encoder. Rather, the redundant slices are only employed in the decoder to replace missing slices, which explains the concept of MDC in video coding. The H.264 compressed sequence with the redundant slices can be used to form two balanced descriptions of the original video sequence, by simply reordering the compressed video data.

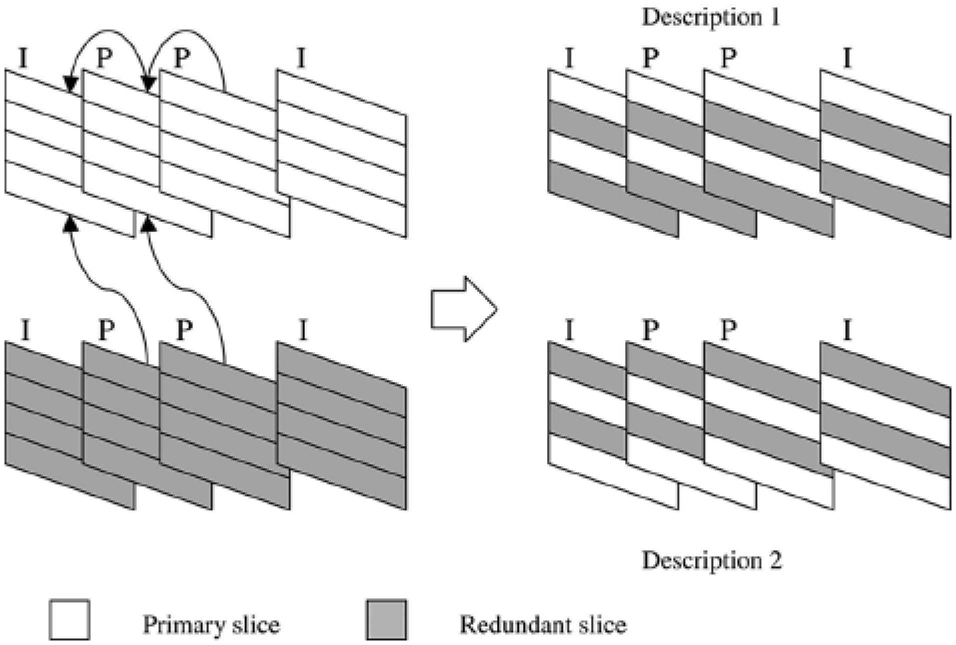


Figure 4.8: Multiple description technique with redundant interleaved slices [177]

This is achieved by interlacing primary and redundant slices in order to create two H.264 bitstreams which contain the primary and the redundant representations of each slice alternatively, as depicted in Fig. 4.8. If both descriptions are received correctly at the decoder, then the primary representation of every slice is decoded and reconstructed with high quality. However, if a description is lost, the received can still be decoded but with lower quality because of the drift generated by the redundant slice. By default, the redundant slice should be identical to the primary picture which is achieved at the cost of high bit rate overhead. In this error resilience scheme, the decoder experiences a better performance when more redundant coded slices are available, and there is no limitation on how much information is sent to the decoder. Through this technique, MDC reduces the adverse effect of packet losses through the transmission of different descriptions along various channels. In addition, error concealment techniques work well in supporting the recovery of the lost information [178]. A major setback to redundant slice techniques in general is the need for extra bandwidth to carry those redundant slices which affect the performance of the technique in general [61].

Redundant slice and in particular MDC should be designed to minimize the redundancy while meeting the overall distortion requirement in an error prone environment. Redundant slice is an error resilient candidate that has been extended to MVC. The work in [179] proposed a technique that can generate and exploit redundant disparity vectors in order to provide error resiliency to the primary data stream in addition to error concealment technique. However, their methodology considered the Joint Scalable Video Model (JSVM) codec and assumed each layer represents a different camera view. The authors in [180] present redundant coding as an error resilient scheme to mitigate the effects of packet loss during transmission. The concept generates selected macroblocks that only falls within the Region of Interest (ROI). The proposed scheme which is depth map enhanced employs a content aware ROI based filtering method to select visually important macroblocks, for redundant data coding. The decision for each and every MB selected is taken with reference to the corresponding depth map information that provides details about the structure of the objects and the intensity of the details embedded in the sequence. Conceptually, the scheme considers the effect of the loss of colour data in areas of the frame where the strength of the depth map is high. Affected MBs within the selected regions are redundantly encoded in order to provide additional protection against losses. However, when primary data is lost, the redundant MBs within the bitstream are utilized to recover the lost information either temporally or from other views. The main setback of this concept is that the quality of the reconstructed video when redundant frame is used is usually low compared to when an error free video frame is being used. While Fig. 4.9 shows the objective result of the study, table 4.1 illustrates the subjective quality comparison between the proposed redundant technique and when no redundant scheme is used. The authors demonstrate that frames 29 and 43 of the Akko test sequence can be reconstructed at 20% PLR using the MDC with better quality compared to when no MDC is used. It is obvious that the work is able to improve the quality of video over erasure network when redundant coding is utilized.

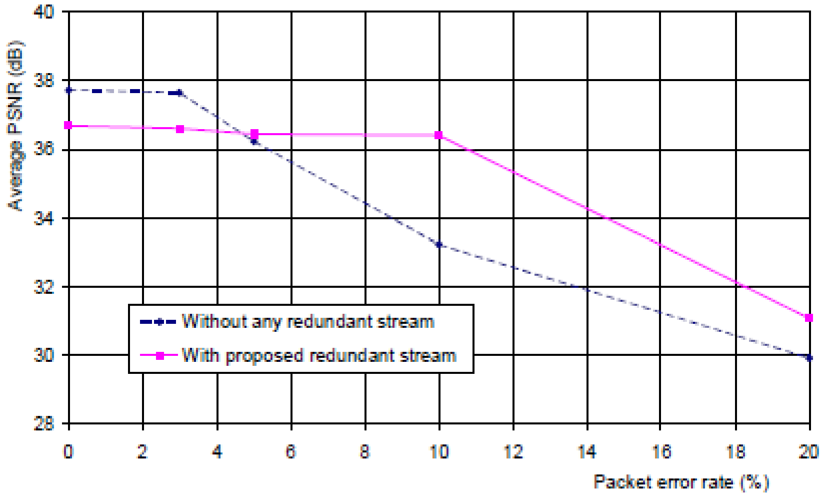
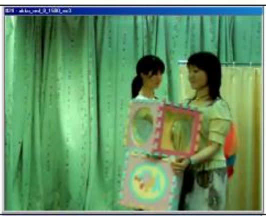
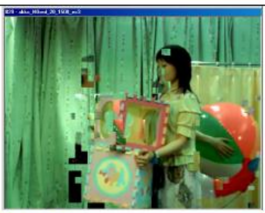
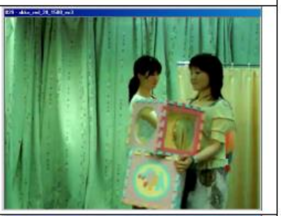
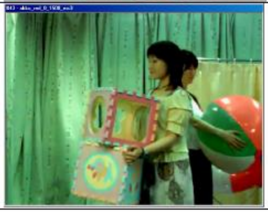
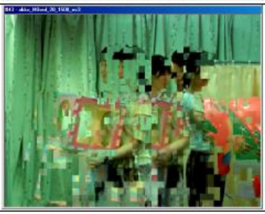
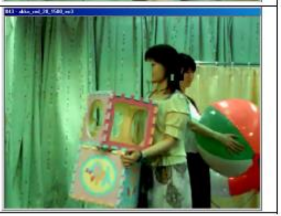


Figure 4.9: PSNR result for Akko sequence at 1Mbps for different error rates [180]

Table 4.1: Subjective quality comparison for MDC with standard technique [180]

Frame number	(a)Error free environment	(b) With the reference scheme	(c) With the proposed scheme
29			
43			

4.3.5. Intra Refresh

In H.264/AVC encoded video sequence, there is a high dependency between many parts of the coded video sequence which gives a better compression gain. However, this dependency has the disadvantage of allowing spatio-temporal error propagation within the bitstream as a result of transmission error or packet loss [181]. Furthermore, as high video compression is required for video streaming applications over the network, error sensitivity also increases which also increases the level of quality impairment when transmitted over error prone wireless networks [182]. The H.264 video coding standard has employed different techniques to limit the spatio-temporal error propagation effect, one of which is intra refresh coding [183]. A simple and straight forward approach is by intra coding the entire frame (I-frame) in order to mitigate the effect of error propagation in the bitstream. This approach is effective in limiting the spatio-temporal error propagation as it breaks the temporal dependencies linked between all the previous encoded frames and also provides error robustness in the bitstream. However, intra coded frames in the bitstream reduces the compression efficiency which makes them not preferable for continuous usage. In video coding, it is more preferable to utilize more highly compressed inter coded frames with some periodic intra coded frames to mitigate any potential error propagation that may occur. Even with this approach, intra frames introduce data bursts at high bitrates when transmitting video data over especially high bandwidth constraint channels [182]. Also, increased number of I-frames introduces delay and is not practical especially for real time video streaming and conversation applications as a result of buffering and transmission overhead [184]. There is a distinct difference in H.264/AVC between regular intra picture and instantaneous decoder refresh (IDR) pictures. An intra-frame does not provide random access property because pictures before the intra pictures may be used as reference for successive predictive coding [145]. A more efficient coding approach in the H.264/AVC standard to mitigate error propagation and to prevent high bitrate bursts is to adopt intra MB refresh [185], [186] where a group a macroblocks (MBs) in each frame of the compressed video sequence is considered for intra coding. Intra MB refreshing is efficient and strictly encoder based, it does not in any way introduce decoding overhead and can easily be combined with error concealment scheme in order to achieve high visual quality in video communication applications [160].

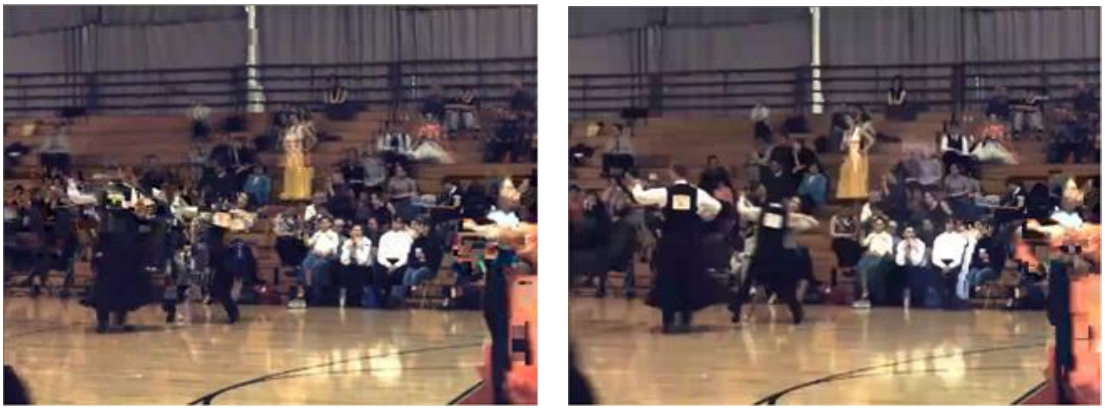
There are various types of algorithm that exist for encoding MBs in intra mode. These algorithms can be categorized into non adaptive and adaptive techniques, and non-adaptive techniques include the circular intra refresh method that scans the picture area in a pre-defined order and encodes a particular number of MBs in a random fashion from chosen MB locations [187]. The adaptive technique can be further classified into cost function-based and rate-distortion optimized algorithms. In general, Adaptive MB mode decision method selects the intra coded MB locations so that the content of the picture is taken into consideration.

For instance, a moving object in a scene will often be refreshed in intra mode as compared to static object or background. The method in [188] explains the concept of cost function based algorithm in details and reports some interesting results that demonstrate with a certain function the cost of each MB, and a certain number of MBs having the highest cost that can be coded in intra mode. The other category of adaptive technique is the rate-distortion optimized MB mode selection algorithm that uses the Lagrangian cost function. In this approach, the mode of each MB is selected so that the combined cost is minimized. Usually, the cost function takes into account an estimate of expected distortion that may be caused by transmission error or losses [187]. Furthermore, this approach can enable the video encoder to maintain a constant bitrate in conjunction with the rate distortion control mechanism in H.264/AVC. Usually, the rate distortion control mechanism is to determine the coding mode with minimum distortion (D) of a coded block with coding rate (R) and complexity (C) constraints. The parameters described can be expressed as a function of minimum Lagrangian cost function (J) in the following well known equation:

$$\mathbf{J = D + \lambda(R + C)} \qquad (4.1)$$

The λ symbol is the Lagrange parameter for appropriate weighting, which is associated with both rate and complexity. D is computed by the sum of absolute difference (SAD) in low complexity mode and by the sum of square difference (SSD) in high complexity mode. For intra block mode, R represents the coding rate of block coefficients, whereas for inter block mode it represents the block residual and corresponding motion vector(s) [189].

A recent and interesting study in multiview video coding based on adaptive rate-distortion technique is reported. The authors [190] proposed a rate distortion optimization algorithm that is based on an estimate of the expected distortion in the encoder could perform efficient coding mode decision. The algorithm aimed at regions of MBs within a slice that may cause significant impairment in the reconstructed picture when affected by transmission errors. Such regions of MBs may be refreshed with intra MBs in order to improve the visual quality of the multiview video. The algorithm could achieve a quality gain of up to 1.55dB for 10% error rate. The subject analysis of frame number 23 in view 3 of the ballroom sequence can be shown in Fig. 4.10



(a) No error resilience (b) with RD optimized algorithm

Figure 4.10: Subjective analysis of frame number 23 in ballroom [190]

It can be observed from the proposed rate-distortion optimized technique in Fig 4.10 (b) that intra MB refresh coding can be used to recover distorted regions of a picture, and also to improve the error robustness of multiview video bitstream when transmitted over error prone-channel. In addition to the recent study in error resilient 3D video coding, the authors in [191] extended the adaptive rate-distortion optimized technique from MVV to multiview video plus depth. In the study, an application is developed based on rate-distortion optimized algorithm that is capable of switching between coding modes during 3D video transmission with losses.

Their approach considers replacing the main source coding distortion of each texture and depth MB by the expected overall distortion of the decoder MB reconstruction. Eventually, the proposed algorithm can optimally select the spatial, temporal or interview mode for each MB while encoding based on the overall estimated distortion. The result obtained is an indication that efficient use of rate-distortion optimized technique can improve the visual quality of the reconstructed sequence.

4.3.6. Reference Frame Selection

In the techniques discussed thus far, the encoder operates independent of the decoder in order to combat the effects of transmission error in the video bitstream. In a situation where both the encoder and decoder are required to interactively combat transmission losses, then a feedback channel can be designed between the decoder and the encoder. The decoder can relay information about which part of the transmitted video data is corrupted by errors, and the encoder can strategize its operation to mitigate the effect of such errors [192]. A simple approach is to retransmit the lost packet especially in an underlying network that supports ARQ or to intra code the data. However, as previously explained, this will incur delays and losses that can be unacceptable for a real time video interactive application. Reference frame selection allows flexible selection of a reference picture on a slice basis or MB basis, because temporal prediction is still possible from other correctly received frames at the decoder buffer, this concept can improve error robustness by avoiding to select corrupted picture regions as reference and also capable of providing temporal scalability. RFS can be used with feedback channel or with no feedback channel, and the feedback channel message can either be a positive or negative acknowledgement about the decoding status of a particular frame [193]. Usually, positive acknowledgement message consists of information received by the encoder about the correct reception of a particular frame from the decoder while, the negative acknowledgement message is a feedback information from the decoder that indicates the presence of error in a particular frame. H.264/AVC allows the application of RFS to a particular portion of a picture rather than the complete picture, the decoder must be informed about the reference picture that was selected to predict a particular segment of the frame. This requires that the temporal reference (TR) of the reference picture to be utilised is sent along with the picture header information to indicate which of the several reference pictures the decoder can use to decode a particular frame [193]. Furthermore, this concept can allow frames to be kept in short term or long term memory buffers for future reference.

The technique can be exploited by the encoder for different purposes like achieving efficient compression, bit rate adaptation, and for error resilience [157]. Technically, when the encoder become familiarized via a feedback channel about corrupted regions of a previously coded picture, it can then choose to code the next P-frame not relative to the current frame but to an older reference frame which is also available to the decoder. The use of reference frame selection does not mean an additional delay to the decoder. The encoder need not wait for the arrival of feedback information about the previous frame in order to be able to code the current frame. Instead, it can select the reference frame prior to the corrupted picture whenever feedback information is received. One form of reference picture selection can be demonstrated in Fig. 4.11, it shows the transmission of frames 1 through 9. The encoder prediction is indicated by the use of curved arrows in the figure. In this concept, the most recent frame is usually utilized as the reference frame for motion-compensation prediction. However, if frame 4 gets corrupted by packet loss, the decoder immediately signals a negative acknowledgement (NACK) message to the encoder about the error. This informs the encoder that frame number 4 is damaged, and that all subsequent frames are prone to error propagation. In this regard, the encoder then selects the most recent frame known to be correctly received (frame 3 in this figure) as a reference for the encoding of the next frame, which is frame 6. Round Trip Time (RTT) determines the required amount of time for the prediction structure to be altered, and consequently determines the number of frames that will be affected by the error propagation [194].

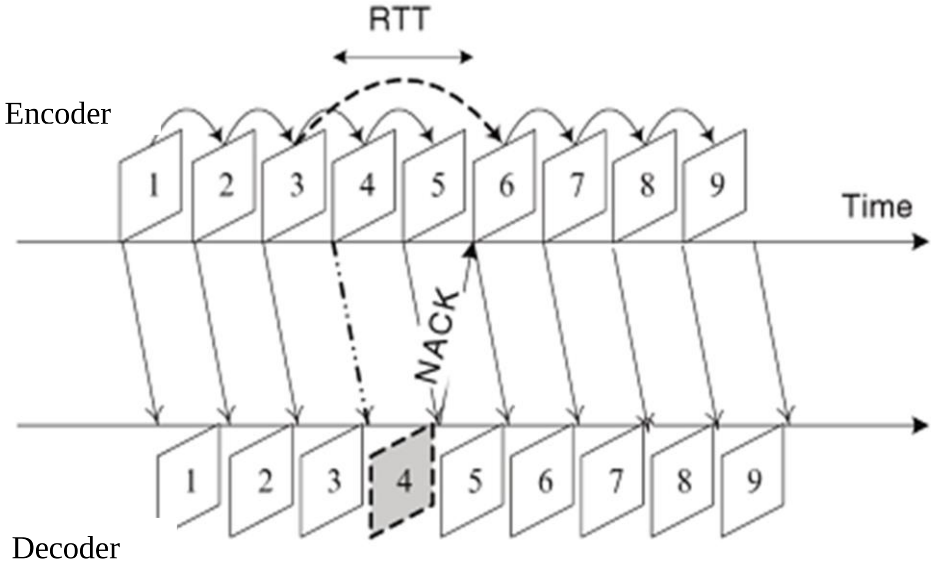


Figure 4.11: Reference frame selection [194].

The concept of reference frame selection has been extended to MVC and different proposals are reported, which include the work in [195]. The paper proposed a multi reference algorithm for hierarchical B-pictures prediction structure that would reduce complexity and improve the quality of the video sequence by exploiting the high reference frame and direction among variable block size coding modes. The authors in [196] proposed a concept of reference picture selection for MVC that could exploit view interpolation for disparity compensation by assigning reference picture indices to interpolated images. In [197], the authors proposed a recursive algorithm that could also estimate the distortion in a synthesized view due to errors in both texture and depth maps information. Furthermore, the algorithm is designed to formulate rate distortion optimization that could select reference pictures for MB encoding. The results demonstrate that, better video quality can be achieved by up to 0.73 dB at 5% loss compared to random intra refresh insertion. Also, [198] reports similar approach and demonstrates that their algorithm could reduce up to 90% of encoding time without affecting the visual quality.

4.3.7. SP-/SI-Synchronization/Switching Frame:

The H.264/AVC design includes a new feature known as SP/SI mechanism [199][200] that is designed specifically for switching between video bitstreams. However, it can also be regarded as an error resilient technique in a network with feedback channel otherwise known as back channel [160]. The concept of switching pictures supports the decoding process of some decoders in order to precisely synchronize with an ongoing video stream produced by other decoders. The process is achieved without losing the decoding efficiency of all the decoders by sending an intra-coded frame [12]. Two types of switching frames are specified that use motion compensated predictive coding which has a better coding efficiency than I-frames [199]. They are the primary and the secondary SP-frames. In this approach, primary SP- frames are introduced into the H.264/AVC encoded bitstream that in general are slightly less efficient than the regular P-frames, but significantly more efficient than regular I-frames. The SP-frames make use of motion compensated predictive coding in order to exploit temporal redundancy in the sequence similar to P-frames [199]. The distinction between the SP and P-frames is that SP-frames allow identical frames to be reconstructed even when they are predicted using different reference frames.

Because of this trend, SP-frames can be comfortably used as I-frames in such applications as bitstream switching, splicing, random access, fast forward, fast backward, and error recovery. More so, since SP-frames are not like I-frames and can utilize motion compensated predictive coding, then fewer bits will be required than I-frames to achieve almost the same level of quality. In some practical applications mentioned above, SI-frames (secondary SP-frames) are used in conjunction with SP-frames. Usually, an SI-frame makes use of spatial prediction as an I-frame and still be able to reconstruct the corresponding SP-frame, which uses motion compensated prediction [199]. Conceptually, to enable a drift-free switching, the streaming server will have to store several copies of the same sequence that are encoded at different quantization parameters, hence with different qualities. Every SP-frame in the bitstream has a specific switching location and as long as switching is not required; the primary SP-frames are transmitted instead of the P-frames. However, if switching is required then the secondary SP-frame is transmitted instead of the SP-frame [201]. Fig. 4.12 illustrates a typical picture switching scenario between two coded sequences. The two coded sequences consist of the SP-frames at strategic locations; the use of arrows indicates the direction of transmission. Switching is achieved at the second SP-frame of the first coded sequence where the SI-frame or secondary SP-frame is transmitted instead of the SP-frame.

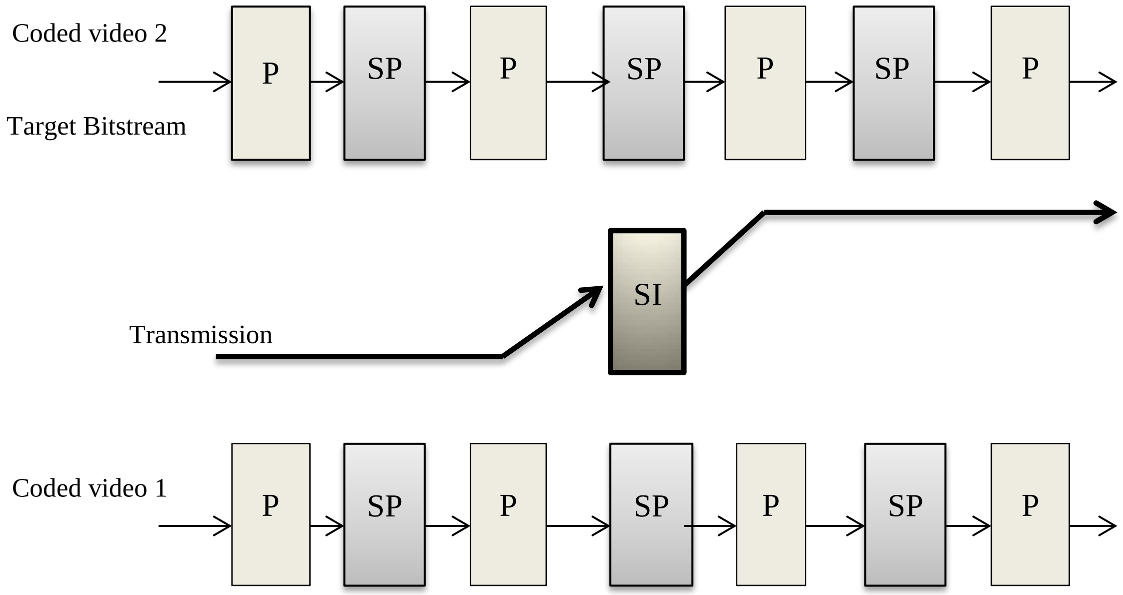


Figure 4.12: Pictures switching between H.264/AVC bitstreams [201].

In order to demonstrate an efficient approach to free viewing for MVV, the authors in [202] report some work in MVV streaming which can enable a user to switch freely and efficiently between two adjacent views. The authors demonstrate that the efficiency could be improved while limiting the effect of mismatch between the references for prediction and reconstruction. The authors in [202] also proposed a free viewpoint switching scheme for MVV that employs distributed video coding technique. Their approach considers producing an alternative bitstream for every frame based on Wyner-Ziv coding method for error correction when the view switching takes place. The Wyner-Ziv bits that correspond to the actual reference frame at the switching point are transmitted to recover the true reference.

4.4. Error Control

In an attempt to minimise the effects of channel errors, error control techniques are usually adopted. The most widely used error control techniques in video data transmission are the Automatic Repeat request (ARQ) and the Forward Error Correction (FEC). The main problem with the ARQ technique is a long delay, which is unsuitable especially for real time applications. For this reason, the use of FEC has been widely suggested because of its reliability in real time applications [159]. Basically, FEC is a type of error control that is widely used to reduce the effects of channel errors in a wireless network by introducing channel codes. Channel codes are classified into codes that can cope with bit errors as well as packet erasures as it is extensively reported in [203]. Typically, the H.264 video codecs can be made more resilient to channel errors by employing FEC codes such as Reed-Solomon codes, BCH codes, convolutional codes, turbo codes, low-density parity-check (LDPC) codes, and product codes. These FEC codes are employed by the encoder to protect the bitstream before transmitting to the decoder, and when received, the FEC codes are then utilized to correct errors in the bitstream. FEC techniques are efficient in mitigating random bit errors; however, their performance against longer duration burst is less efficient [204]. Furthermore, the FEC technique incurs constant transmission overhead even when the channel is error free. Hence, the coding efficiency of the video may be compromised and lost [159]. In order to efficiently utilize a bandwidth constraint wireless channel and to maintain coding efficiency, the Joint source channel coding technique is introduced. The concept is aimed at developing a technique where compression, protection, and transmission are jointly put together to ensure a high level of system performance [205].

Shannon theory considers the concept of separation theorem in JSCC for applications that requires source video transmission over rate constrained networks in [206]. Usually, improvement in performance may be achieved by moving from separate design and operation of source and channel codes to joint source-channel coding [205]. The JSCC technique allows minimum distortion since the distortion in video communication can be classified into two categories: source distortion and channel distortion. The source distortion does not depend only on a particular source coding bitrate, but also depends on the characteristics of the input videos and the data representation scheme which is employed by the coding algorithm [159]. For H.264/AVC compliant videos, the Quantization Parameter (QP) number can be selected and varied in order to achieve suitable bitrate and quality. The QP regulates the strength of quantization, and its value is usually selected in the rate distortion optimization process. However, specific bitrate requirement can be achieved by adjusting the value of QP that depends on the nature of the video content. On the other hand, the channel distortion is caused by transmission error in an error prone channel which affects the transmitted video data. There are several related work and study in the literature on joint source and channel coding which demonstrate a better visual quality performance in video communication. The authors in [166] proposed a scheme that exploits the FMO error resilient features of H.264/AVC reference software and employs Reed-Solomon codes to protect the compressed video. The experimental evaluation of their work has demonstrated a higher quality performance compared to the conventional H.264/AVC transmission scheme. The authors in [207] presented some work that combines data partitioning technique with FEC technique. Their proposed framework considers the use of both unequal error protection (UEP) and equal error protection (EEP) channel coding schemes on 3D stereoscopic H.264/AVC video over a noisy channel. The analysis of their work demonstrates an overall quality improvement in both the main and auxiliary views if partition C is protected appropriately. Error correction and control techniques have successfully being adopted in MVC representation due to its hierarchical dependencies which can support the capturing of priority video data. The technique is suitable for quality improvement of the reconstructed multiview video content. Thus, this feature of MVC can be exploited through the use of FEC coding among views in order to reduce the effects of transmission losses in an error-prone channel.

4.5. Error Concealment and MVC Resilient Decoder

4.5.1. Introduction

H.264 has standardized all the syntax and semantic elements that are necessary to decode an error free compliant bitstream. In addition to the design requirements and implementation, a decoder should be able to deal with transmission errors and one way a decoder can handle losses caused by transmission errors is through error concealment technique. Error concealment is a non-normative feature in H.264/AVC. It is an efficient post processing technique that is capable of ensuring error control in the decoder without extra cost in bitrate or further delay. A H.264/AVC error concealment decoder should be able to detect and conceal transmission errors by reducing the visual impairment in a frame by interpolating the missing MBs from correctly received neighbouring intra or inter MBs [165]. There are several error concealment schemes in the literature ranging from the basic simple methods to the complicated approach. Most of the techniques assume that the pixel values are smooth across the boundary of the lost and well received regions in both the spatial and temporal domains [160]. The concept of EC is based on Boundary Matching Algorithm (BMA), which is a common and very popular motion compensation technique. The technique is recommended as a non-normative part of the H.264/AVC standard for temporal concealment [208]. The BMA calculates the motion vector of an entire block of pixels instead of individual pixels and the same motion vector is applicable to all the pixels currently in the block [24]. The BMA computes the block difference as the sum of absolute differences between the boundary pixels as depicted in Fig. 4.13. The MV with the least distortion by the BMA is selected as the concealed MV.

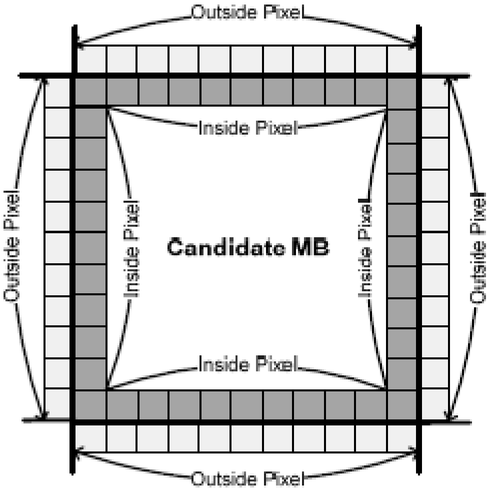


Figure 4.13: The concept of Block Matching Algorithm [165]

Usually, the current frame is divided into blocks of pixels, and motion estimation is performed independently for each available block of pixel. In BMA, a block of pixel from the reference frame that best matches a block of pixels in the current frame is identified, and the motion is estimated. The reference block of pixels is generated by displacement from the location of the current block in the reference frame which is represented by motion vector (MV). Motion vector consists of horizontal and vertical displacement values (x, y). In general, error concealment algorithm utilises the concept of BMA in the decoder to minimize the visual degradation of a sequence by interpolating the lost or erroneous samples from spatially or temporally correlated blocks. The H.264/AVC has specifically suggested a scheme as part of the standard that uses inter and intra picture interpolation algorithms, which can also be used as a benchmark to evaluate other error concealment schemes [160]. The spatial error concealment tries to estimate a pixel of a lost block by interpolation based on a weighted average of correctly received boundary pixels as shown in Fig 4.14.

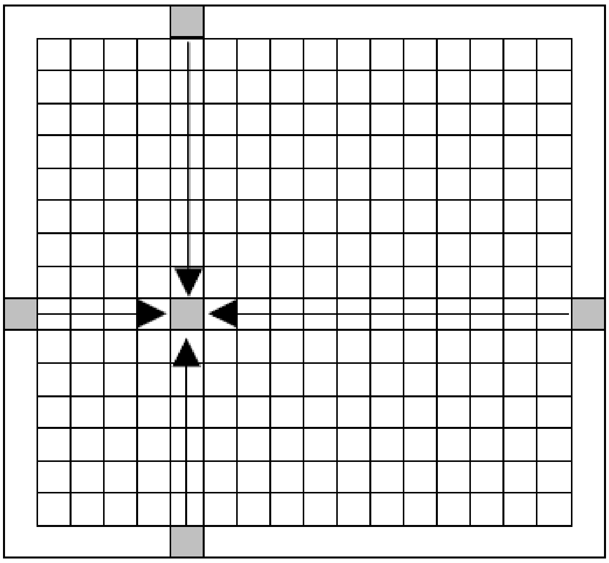


Figure 4.14: H.264/AVC spatial concealment scheme in a 16x16 block [160]

From this scheme, if there are at least two correctly received blocks available around the boundary of the missing pixel, then only those error free blocks are utilized for the interpolation. Else, the surrounding already concealed blocks are used. Some disadvantages of the spatial error concealment include the blurring of interior pixels, artefacts and rough edges [165].

Temporal error concealment algorithm is based on motion vector estimation of the lost MB in an inter-coded picture by interpolation in order to exploit the correlation that exists between the lost block and its spatial neighbours. The approach is based on the assumption that motion of a small region is usually consistent; hence, it is reasonable to predict the MV of a lost block from the surrounding and adjacent MVs. Thus, motion compensation is then applied to obtain a temporal replacement for the lost MB by using a reference frame, as described in [209]. In order to compute the motion activity of the correctly received MBs, a simple assumption is made and computed as follows. If the average motion of the correctly received MBs is smaller than a predefined threshold usually ¼ pixel, then all the lost MBs are concealed by directly substituting the MB from the collocated position in the first forward reference frame [168]. Otherwise, the missing MB is assigned an MV from one of the four neighbouring candidate MBs based on the assumption that, the motion of neighbouring regions is highly correlated. Usually, the smallest block size that can be considered from the surrounding blocks is an 8x8 luminance block. However, for an MB with smaller block sizes, the average of the MVs in the 8x8 blocks is used as a candidate. Eventually, the list of the candidate MVs is then utilized to motion compensate the macroblock. The MB that gives the least boundary distortion from equation (2) is selected as shown in Fig. 4.15.

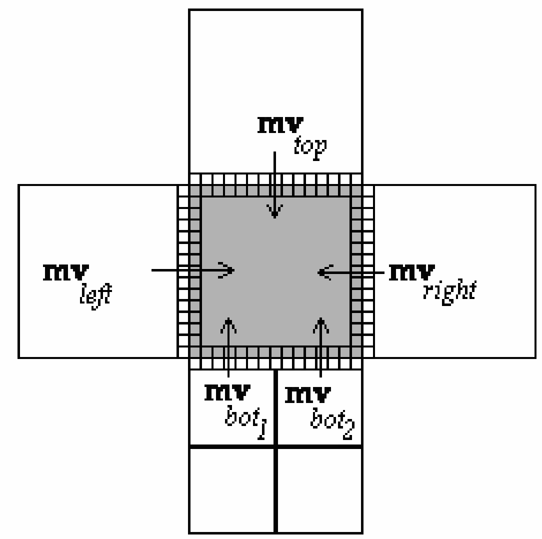


Figure 4.15: Motion vector estimation for prediction [160]

The distortion gives an estimate of the luminance change across block boundaries. Thus it computes the smoothness between the image and the motion compensated MB [168].

$$\mathrm{d} = \underset{dir\,\epsilon\,\{top,bot,left,right\}}{\mathbf{min}\,arg} \left\langle \sum_{j=1}^{16} \left| Y_{ref}(mv_{dir})_j - Y_{rec_j} \right| / N \right\rangle \qquad (4.2)$$

From equation (2) above, d is distortion error for the surrounding luminance blocks, $\mathrm{Y_{ref}}$ is a luminance pixel value from the boundary of the MC MB, $\mathrm{MV_{dir}}$ is the MV used to MC the MB, $\mathrm{Y_{rec}}$ is a luminance pixel from the boundary of the reconstructed frame, and N is the number of pixels averaged. In inter-frame concealment, the zero MV block in the first forward reference frame (MB SKIP) is also considered as a candidate MV for concealment.

For B frame concealment, several MVs from the surrounding of the candidate MBs are available and are potential candidates that can be used for concealment. The choice of selection is such that, if only the forward MV or backward MV is available, then it is selected. However, if both the forward and backward MVs are available, then only the forward predicted MV is used. Furthermore, when computing the boundary distortion as in equation (2), where more than one neighbouring MB is correctly encoded, then only these MBs are used for the computation [168]. The BMA algorithm can be extended into MVC to exploit the disparity vectors for interview concealment, providing a fair performance [210].

4.5.2. Error concealment for MVC

In general, the MVC adopts a coding structure which is similar to the single view H.264/AVC format and standard. This feature of MVC makes it possible for many concepts in single video coding to be extended in MVC. An example of these concepts is the use of Boundary Matching Algorithm (BMA) in MVC, which is a basic temporal motion compensated error concealment technique. The BMA is modified to accommodate and utilize the disparity vectors from the non-base view for interview concealment, thus minimizing visual degradation and providing a fair performance [210]. In MVC bitstream, the base-view (view 0) is independently encoded without any interview dependencies except with temporal dependencies. This is also exhibited in single view videos, which means that the base view of MVC bitstream can be concealed by techniques already available in H.264/AVC standard.

In order to mitigate temporal error propagation and to achieve random access, intra coded frames are also used, which are also concealed in the same way as single view video. That can be done through weighted average interpolation method, while the temporally predicted frames are concealed by the use of temporal motion-compensated concealment [168]. For non-base view such as view 2, which is a single interview also predicted have their anchor frames forward interview predicted. Since this is similar to temporal prediction but utilizes disparity vectors with the interview reference frames, such frames can utilize the BMA technique but with a list of candidate DVs from the interview reference frames. The non-anchor frames of single forward predicted views in MVC are all temporally predicted in a similar way as in single view video. Thus, the typical temporal motion compensated technique can be used. The interview bi-predicted sequence, such as view 1 from Fig. 4.16 is considered as B-frames. The anchor frames in this view are bi-predicted from two different interview frames; hence the motion compensated candidate list is generated, in the same way, as that of B-frames. The lost MB from the current frame is MC from the reference inter-view frame of the candidate DV under consideration. Non anchor frames in MVC are bi-predicted from both temporal and interview frames so that the choice of the reference frame depends on the MV or DV under consideration. In such condition, a combined temporal and interview EC is achieved where the MC MB is generated from both temporal and interview reference frames. The best MC MB is then selected from the temporal or interview reference frame [168]. Frame copy error concealment method on the other hand is a simple algorithm whereby each pixel value of a concealed picture is copied from the corresponding pixel of the previous picture in the reference picture list 0 (RefPicList0) [211]. It will be demonstrated later in chapter six how frame copy concealment is modified to support the decoding of multi-layer data partitioned MVV bitstream and improve the perceptual quality of the reconstructed views for different error rate. For conciseness, chapter six will also focus mainly on decoder optimization and performance using frame copy concealment in MVC since it is adopted to support the decoding of the bitstream with losses and to improve the quality performance of the sequences.

4.5.3. Error Resilient Decoder

The H.264 compliant decoder must be capable of using a defined subset of tools known as a profile [13]. There are two main problems that an error resilient decoder would have to deal with. A robust H.264 decoder should be able to detect transmission errors, and on the other hand to apply an appropriate concealment technique on the errors detected [145]. Usually by default, the decoders always expect error free slices in the correct decoding order. However, when corrupted slices are received they are discarded before been sent for decoding. The received error free slices are stored in a buffer in a way that the right decoding order is recovered before decoding commences. A standard H.264 compliant decoder is capable of detecting if an entire slice or frame is lost in a bitstream. The decoder can work this out through the use of frame number syntax element. Each slice header contains all the useful information such as frame number and all other information the decoder may require to decode the pictures. Frame number increments by one for each coded frame sent from the encoder and then used for motion compensation by the decoder [212]. However, when decoding a slice in the picture, if the frame number of the next slice equals the expected frame number then the decoder decodes the slice and updates the binary map. If the frame number is greater than the expected frame, then the decoder assumes that all the received slices of the previous pictures have been decoded. Thus, the binary map is checked and if all the MBs in the picture are not entirely recovered then a slice loss is inferred. In this case all the losses are concealed [145] by copying previously decoded pictures from the reference picture list and replacing all the lost pictures in the coded bitstream. Eventually, the binary map is reset by the decoder and the next slice in the picture is presented for decoding [145].

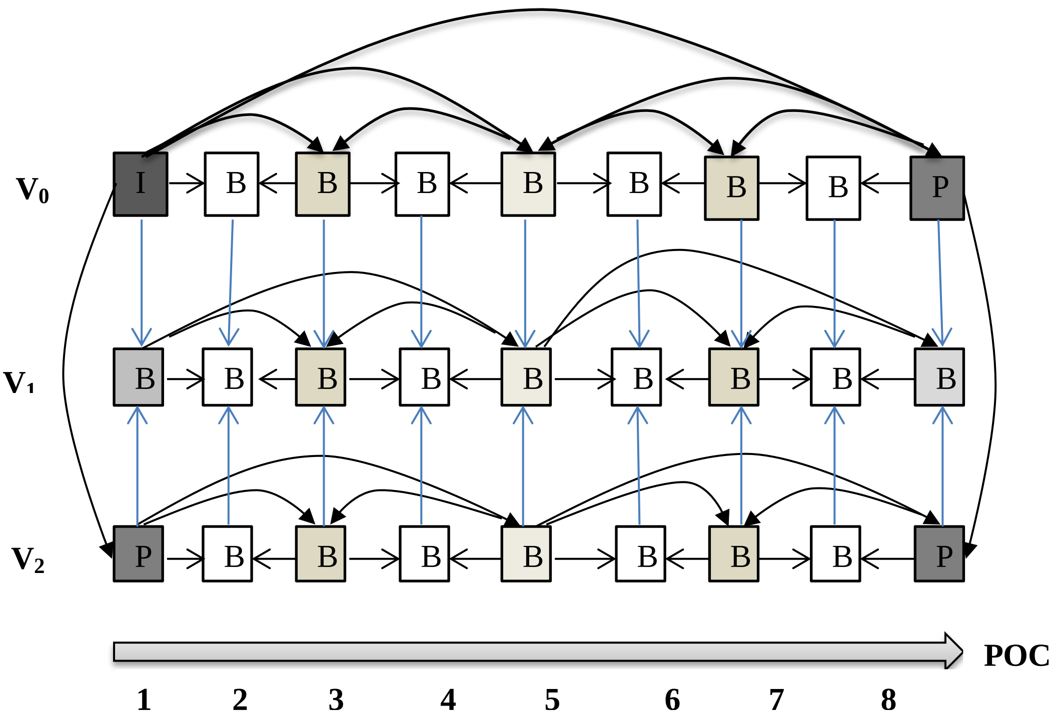


Figure 4.16: GOP Structure for MVV Bitstream

In H.264 video coding standard, frames are arranged into a group of pictures (GOPs). Usually, A GOP includes an I-frame and all the subsequent frames till the next I-frame within the video sequence. I-frames mark the start point of the GOP that also contains the full image without requiring any additional information for reconstruction. Standard H.264 compliant encoders use GOP structures that can allow each I-frame to be a suitable random access point in order that errors within the GOP structure are corrected by the next I-frame. However, in the recent designs such as H.264 MVC extension and HEVC, the encoders have a high level of flexibility in the referencing structure. In this type of structuring they can use more pictures as references and can also utilize more flexibility of the coding order relative to the display order. B-pictures are important and in advanced structures they can be used as references when coding other (B or P) pictures, this structuring is otherwise known as hierarchical B-picture concept [31]. Fig. 4.16 is an illustration of hierarchical B-frame structuring for MVC bitstream. This advanced structuring and flexibility can improve compression efficiency accordingly and at the same time it is very susceptible to error that can cause error propagation due to transmission errors or losses within the MVV bitstream.

The effect of error propagation in the MVV bitstream can be mitigated through the concept of hierarchical B-pictures which can ensure that the number of pictures affected is limited. The frame ordering within GOP is determined by the inter-relationships between the frames in accordance to which frame information is used to reference the other frame. It is crucial to understand GOP structuring especially when considering MVV quality loss over IP packet drop network. The interdependency between the frames or views affects the decoding order to be different from display ordering (picture order count). This will take us to another powerful feature of MVC known as time first coding, whereby pictures of any temporal position are grouped together in decoding order. However, pictures of the same time instance can be defined, but belonging to different views as one access unit [34]. Time first coding depicted in Fig. 4.17 is an MVV bitstream format representation that allows all views to be encoded and then assembled in a time domain for suitable transmission. The decoder on the other side can receive and reorder the bitstream in the right decoding order to decode all the pictures in different views. The decoder is designed to decode the MVC bitstream in the same time domain and display the videos in the correct order. Time first coding supports the implementation of frame copy error concealment in MVC because of the display nature of all the frames across the views in the same time domain which makes it easier to conceal missing pictures from previously received pictures in the reference list.

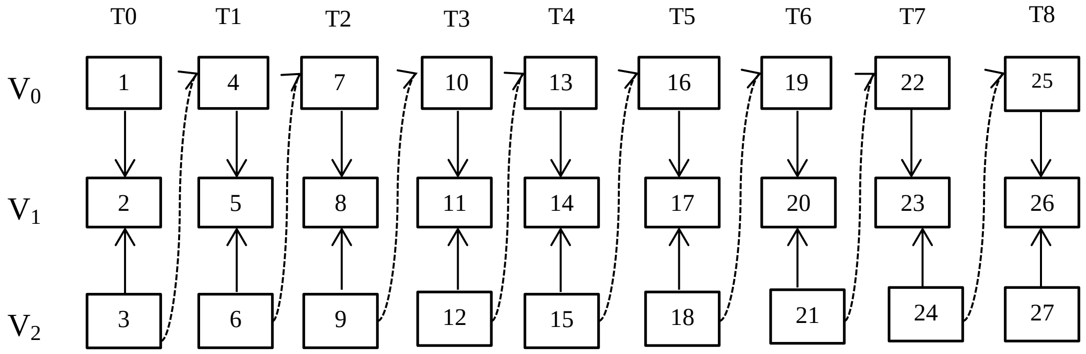


Figure 4.17: Time first coding [34]

4.5.4. Decoding MVC Erroneous Bitstream

Transmission of compressed multiview video bitstream is more susceptible to transmission errors than a 2D video coded bitstream because, in MVV bitstream, each video has interview dependencies [133]. The extensive use of motion vectors prediction and variable length coding further makes the H.264/AVC compliant bitstream vulnerable to these transmission errors. A single bit error can cause a catastrophic effect in the visual quality of the reconstructed views. The effect may cause the decoder to lose synchronization, so that even the correctly received following bits become useless. Hence, the MVV bitstream may be impossible to decode [192]. Currently, the MVC reference decoder only accepts H.264 compliant bitstream and does not support the decoding of erroneous coded bitstream. In order to be able to decode a corrupted multiview video coded bitstream, the H.264/AVC error concealment technique is implemented in the MVC reference decoder. The idea is to make the modified decoder capable of adapting and coping with the losses within the MVV coded bitstream. In general, the reference JMVC decoder, unlike the reference encoder must consider all the required measures of a standard to be able to decode a compliant MVC bitstream. The H.264/AVC extension identifies how the decoding algorithm should reconstruct all the frames in the coded sequence. Frame copy error concealment technique is simple and usually quite effective in video content where the motion is not large [148]. Also, the JMVC 8.5 reference codec has two types of reference frame lists that is also part of the standard and can be used to support frame copy error concealment in MVC. The main difference between the two reference lists is that list 0 utilizes the temporally earlier key frames (I or P) within the GOP, while the reference picture list 1 utilizes temporally closer reference frames which could be a B frame [213]. Conceptually, reference list 1 can ensure smoother pictures because the frame to be copied is nearer the picture to be reconstructed.

4.6. Conclusions

This chapter presents a review of the error control techniques in H.264/AVC standard video communication systems such as error resilience, error correction, and error concealment. As discussed earlier, error control techniques are not part of the MVC extension of the H.264/AVC. Few researchers have proposed some different schemes that can make the multiview video bitstream more robust to transmission losses and to improve the perceptual quality of the multiview video sequence. Some of these relevant proposals are also reported in this chapter that demonstrates the quality improvement achieved. The MVC decoder operation with losses is also discussed in order to highlight the effects and to specify the need for a solution to reduce the impact of losses in the MVV bitstream. The next chapter gives a detailed description of the experimental work, setup and the network simulation test bed that is used in the research work. The simulation results from the developed and proposed technique are presented and analysed further in chapter six.

5. Chapter Five: Simulation, Experimental Setup, Conditions and Analysis

5.1. Introduction

The overall goal of the network simulation is to examine and evaluate the performance of multi-layer and H.264 data partitioning techniques over a simulated network with different loss rates. In this work, we employed the Sirannon network simulator which is an open source multimedia streaming application. Sirannon is accepted and defined in the final test plan of the Video Quality Expert Group (VCEQ) for streaming video sequences and simulating network impairments [214]. In addition to that, the application supports streaming with different codecs including the H.264/AVC with its extensions such as SVC and MVC. There are many valid and available loss simulators which can be used to simulate different types of network. In our experiments, the multiview video bitstream file of each technique is uploaded and read in the Sirannon environment for the network simulation of various desired packet loss rates using the predefined test bed in Fig. 5.1. In the experiments conducted, the Gilbert loss model [215], which is a very common error modelling tool is adopted and used to introduce random errors in the bitstream according to Gilbert modelling. The impaired MVV bitstream of each of the technique is decoded using both the H.264 DP and multi-layer DP reference decoder. After successful decoding of each bitstream, the quality of each reconstructed view is measured and recorded for further analysis. Note that the quality degradation experienced by the MVV bitstream during compression is not taken into consideration; rather, the amount of video packets that are lost during transmission in the simulated network is taken into account. In this simulation, two different MVV bitstreams are used for evaluation. The two different MVV bitstreams are evaluated in the network simulation, the first MVV bitstream is generated using the H.264 data partitioning technique and the second MVV bitstreams is generated using the multi-layer data partitioning technique. Both MVV bitstreams are simulated with different error rates of 1%, 5%, 10%, 15% and 20%. For each packet loss rate, 10 runs of simulations are carried out and the average result is computed.

5.1.1. The Network Simulation Test bed

The Sirannon network simulator is a modular multimedia streamer which supports a wide variety of video formats and streaming protocols for use both in real time video streaming and offline simulation [216]. In this simulation, the offline mode is used. Fig. 5.1 shows the network simulated model that is used to introduce packet loss in the MVV bitstream for different percentage error rate. The multiview coded sequence is read and packetized by avc-reader and avc-packetizer. The avc-packetizer is capable of packetizing the H264 compliant bitstream into packets suitable for real network and the simulated network as defined in RFC 3984. The gilbert classifier component has a random chance of introducing packet loss across the bitstream based on the Gilbert loss model. The damage stream is unpacketized by avc-unpacketizer block back into the original NAL units format and the resulting coded stream which has lost some of the original video slices based on the error rate selected is written to the basic component writer. Statistics component measures and generates at interval regular information about the passing stream and losses in the buffer. An important block called sink helps to terminate the program gracefully when the last packet of the sequence has passed through the sink.

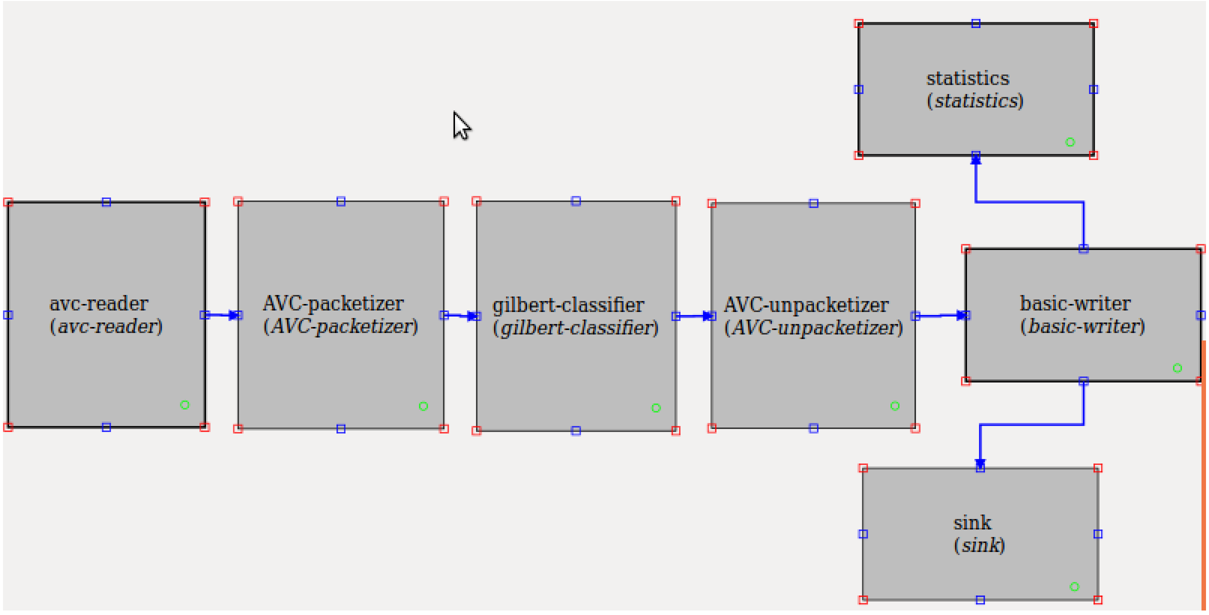


Figure 5.1: Network simulation test bed

5.1.2. Test Model Validation

The Sirannon network simulator as previously explained in section 5.1 is accepted as part of the reference tool chain. The tool is defined in the final test plan of the Video Quality Experts Group (VQEG) Hybrid Perceptual/Bitstream project. The tool is efficient for streaming video sequences and simulating network impairments. Several methods are available for validating a network simulation model which includes common sense and intuition, measurement, alternative models, and incremental analysis [217]. The simulation test model in Fig. 5.1 is validated by employing the simple common sense and intuition method. This approach confirms that the network simulator that produces the model is thoroughly debugged, and the test model generates anticipated results that are similar to the ones in practical applications. The procedure employed consists of a comparative evaluation of the results generated by the model against results that are historically produced by real and practical systems. It is a requirement that the simulation model should be performing under similar conditions as a real network. Expert intuition is another methodology adopted in the validation process. Consequently, the results obtained in the simulations follow a general characteristic that is anticipated especially in video transmission. It can be observed from the simulation results in chapter six for quality versus error rate that it is logical that the quality of a video sequence deteriorates as the error rate increases. This characteristic is observed in practical scenarios and is confirmed by the simulation results.

5.1.3. Gilbert-Elliot loss Model

The bit errors were first modelled in [218] through the use of Markov chain. Gilbert modelled the transmission channel as having two states, Good state and Bad state. When the channel is in the Good state, all the bits are transmitted correctly, which means that the channel is equal to a perfect channel. On the other hand, when the channel is in a bad state, the channel is said to be in a binary symmetrical channel [219]. The transmission of bits in the bad state will suffer a specific bit error rate. For this reason, the Gilbert model is modified by Elliot [220], where the good state is also modelled as a binary symmetrical channel. The transmission channel can assume to have two states, which are the good state and the bad state. These two independent events can be denoted as G and B, respectively.

It can be seen in Fig. 5.2 that the probability of finding the next state is determined by only the current state without any relationship with any previous state.

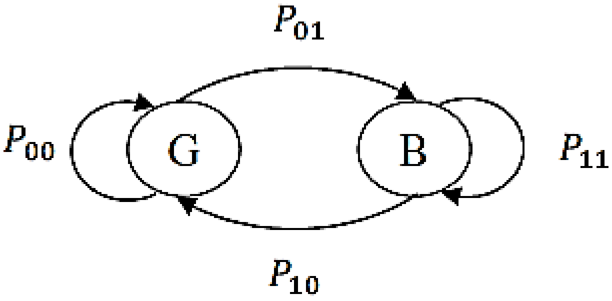


Figure 5.2: Gilbert-Elliot state diagram for packet level [219]

The value of the transition matrix, P_{10}, P_{00}, P_{01}, and P_{11}, can be computed according to the condition of the channel or from real world tracing results. Usually, the bits that are transmitted over simulated wireless channel are handled by certain bit error rate (BER) to be corrupted and the value of BER is determined by the state of the channel. However, when the transmission channel is in the good state, the bit error rate is low and the bit error rate for the bad state is usually high.

5.1.4. Multiview Video Encoder Settings

All the experiments and simulations conducted in this work were tested on the MERL sequences, Ballroom, Vassar, and Exit [221]. The multiview video sequences were encoded with the JMVC 8.5 reference soft for H.264/MVC [222]. The 4:2:0 Chroma sub-sampling format was considered and a resolution of 640 x 480 pixels. The H.264/MVC codec as part of the standard supports the profile classifications. Our experiments are all based on the Extended Profile (XP) which is intended as a video streaming profile. The XP profile has relatively high compression capability and can support some standard error resilient schemes for the video data. For simplicity and efficient decoder buffer management in this work, we employed three views and considered the first view to be the base-view and the second and third to be bi-predicted interview and forward predicted view respectively. Symbol mode is set on Content Adaptive Variable Length Coding (CAVLC) to support the DP in the extended profile, also one slice per NAL unit is considered as part of the H.264/AVC network friendly design [223].

The Quantization parameter (QP) was carefully selected and set to 31 and for each experiment with different GOP, a suitable value for intra-coded frame was also carefully selected and inserted periodically in order to limit the temporal error propagation. The JMVC 8.5 reference software and simulations were configured as in [224]. Table 5.1 summarizes the key parameters used for setting up the JMVC reference software in the experiment

Table 5.1: Experimental Settings

MVV test sequence	Ballroom, Exit, and Vassar
Number of Views	3
Frame Size	640x480
Frame rate	25Hz
Number of frames per view	250
Quantization parameter (QP)	31
Group of Pictures (GOP)	4, 8, 12 and 16
Entropy Coding	CAVLC
Intra period coding	Enabled
Bitstream format	Packet oriented bitstream

5.1.4.1. Extended Profile

The system of profile as part of the H.264 standard involves some sets of capabilities aiming to achieve specific classes of applications. The use of extended profile which is an important classification that is designed to support multimedia content over error prone environments is necessary and useful in this work. Being the only profile that allows the use of error resilience tools in video coding, this important feature is further exploited in this work towards the implementation of data partitioning into JMVC reference software. Because the JMVC reference software does not have any error resilient mechanism, as the first step, we had to import the H.264 DP algorithm into the JMVC reference software. Part of the modification made in order to support and extend data partitioning in MVC is to change the entire SPS header NAL units from the main profile to extended profile for the base view.

In this approach, the MVV bitstream can be decoded by a H.264/AVC decoder thereby achieving downward compatibility. The partitioned non-base view is made compatible and decodable by the JMVC reference decoder by adding three new NAL units into the bitstream. This is also a means to distinguish all the non-base view NAL units from the base view NAL units in the partitioned MVV bitstream.

5.1.4.2. Quantization Parameter (QP)

Quantization parameter is a key coding parameter that controls the use of spatial information in an image or sequence. When the value of QP is very small, the detail of the image is retained. When QP increases in value, some of the spatial detail is aggregated as a means to reduce the amount of bit rate at the expense of increased distortion and quality degradation [225]. This experiment considered different values of QP from high to low in order to get the good quality result with a fair bit rate. By default, QP value changes dynamically as the video coding progresses. This is because the complexity of the pictures is also changing in a video sequence. However, when a high motion scene is reached you will notice a quick rise in the bit rate instantly. Because the JMVC reference software does not have the rate control algorithm that can dynamically adjust the QP to achieve target bitrate, by manually and through trial and error selected a QP value of 31. This results to better quality for the experiments and simulations.

5.1.4.3. Group of Pictures (GOP)

Group of Pictures (GOP) structure is an important coding parameter that determines the video quality and compression efficiency in H.264 video codecs. We have examined the performance of different GOPs on the quality of multiview video sequence over a simulated and erroneous network in chapter six. Our aim in this investigation is to find a suitable GOP size that can provide the best visual quality for multiview video sequence. Because transmission errors or packet losses cause severe effects particularly on MVV bitstream over wireless network, it is very important and necessary to find an optimal GOP size in order to minimize quality degradation resulting from error propagation and to maximize compression ratio. Also in this experiment, we observed the importance of trade-off between compression efficiency and video quality. Based on the type of video application required, sometimes we trade video quality for coding efficiency and vice versa.

Various GOP sizes are evaluated and based on experimental results, a recommended optimal value for the size of a GOP that can give an acceptable perceived quality and low bitrate especially for MVV streaming applications is presented. Another observation we made in our analysis is that different MVV bitstreams perform differently in a packet loss network with different GOP size, and there is a limit where the GOP size can reach that it can either have no effect or decrease in video quality.

5.1.4.4. MVV Test Sequence

The MVV test sequence used in our experiments and simulations are all found in [221] with all their configurations and validation. MVV content can have an impact on the picture quality. This can be viewed based on the following factors such as motion or panning of the camera and background complexity. Three types of MVV sequences are used by the JMVC reference encoder to generate the MVV bitstream for our experiments and simulations namely: Ballroom, Exit and Vassar. Each one of the video sequences has a different characteristic and reacts differently with the coding parameters used in this experiment. The ballroom test sequence contains a period of rapid motion, large objects and very complex background. The exit test sequence contains moderate motion and it has a lot of background details and information such as lighting, shadow and reflection. The vassar test sequence contains moderate motion with lower detail background information. Usually, videos with very high motion or camera panning and complex background are difficult to predict and will require more bits for detailed information such as the Ballroom sequence, unlike a video with minimum or no motion and very simple background. The encoder can predict all the MBs with ease and can be made to concentrate bits in a required region.

5.1.4.5. MVV Bitstream Structure and Analysis

In the H.264 bitstream such as the type that is used in this work, each frame is placed into one single slice whose video data is stored in a single NAL unit.

The slice layer within the H.264/AVC bitstream structure (Fig. 5.3) is very important in the handling of errors in the MVV bitstream. The slice layer is also part of the bitstream that the DP concept is implemented in the MVC codec.

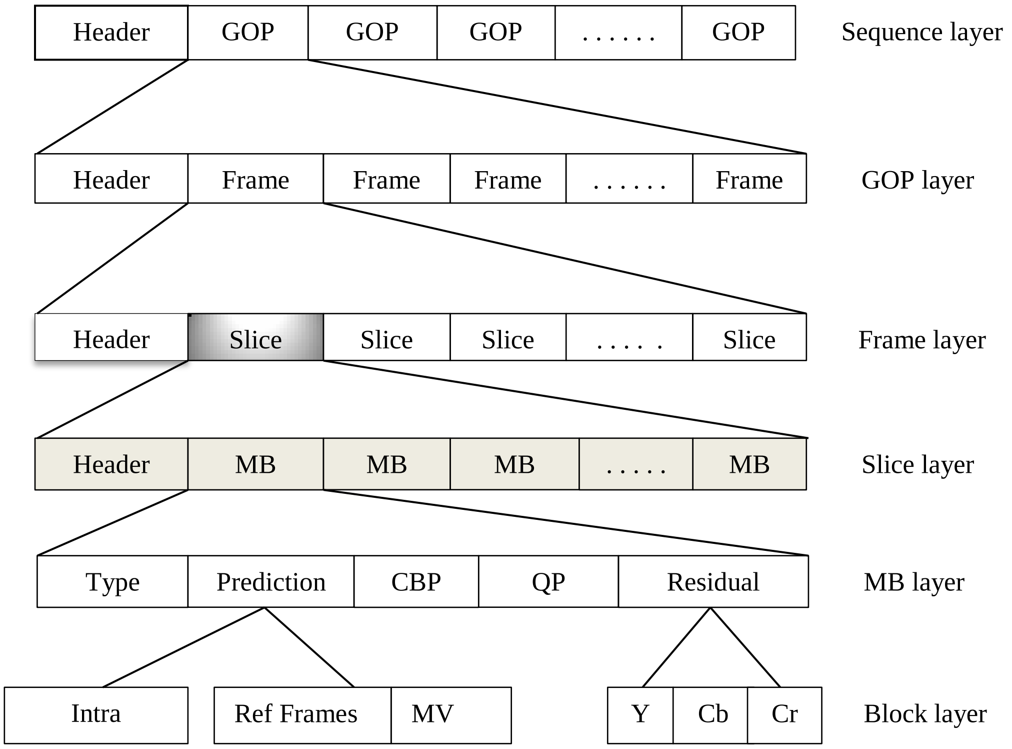


Figure 5.3: H.264 bitstream layers

5.1.4.6. The H.264/MVC modules and Library

Technically, the JMVC reference software is a complex tool, which consists of the encoding and decoding section. Both the encoder and decoder consist of several modules to execute a specific task. It is not possible to explore and work on all the modules in the tool as that would mean re-writing the entire standardized version. Rather, key modules that are related to the work are explored and utilized. In this work, a separate application that can parse and partition all the VCL slice elements in the MVV bitstream is developed. The developed application is then compiled with the encoder core modules in the reference software. On the decoder side, the key modules such as the slice and NAL unit modules are modified in order to accommodate and decode the data partition MVV bitstream. The error concealment frame copy application is modified and compiled with other decoder modules within the library of the JMVC 8.5 reference software as was done with the data partitioning application on the encoder side.

5.1.4.7. Analysing the MVV Bitstream

CodecVisa is a powerful but commercial real time bitstream viewer and analyser that supports the bitstream view and analysis of H.265/HEVC and H.264/AVC/MVC video sequence [226]. The tool is used to visualise and extract the information of any syntax element from the bitstream down to MB level. Useful header information such as first_mb_in_slice, slice_type, frame_num and so on is analysed both in binary and hex format especially when reading and writing the video streams. Fig. 5.4 to Fig. 5.8 demonstrates some experimental analysis conducted in the research work. Another very useful tool used in the research work that helps us in understanding the bitstream structure and analysis is the Source Forge open source software [227]. The software provides a complete set of functions to read and write video streams conforming to the ITU H264 (MPEG4-AVC) video standard.

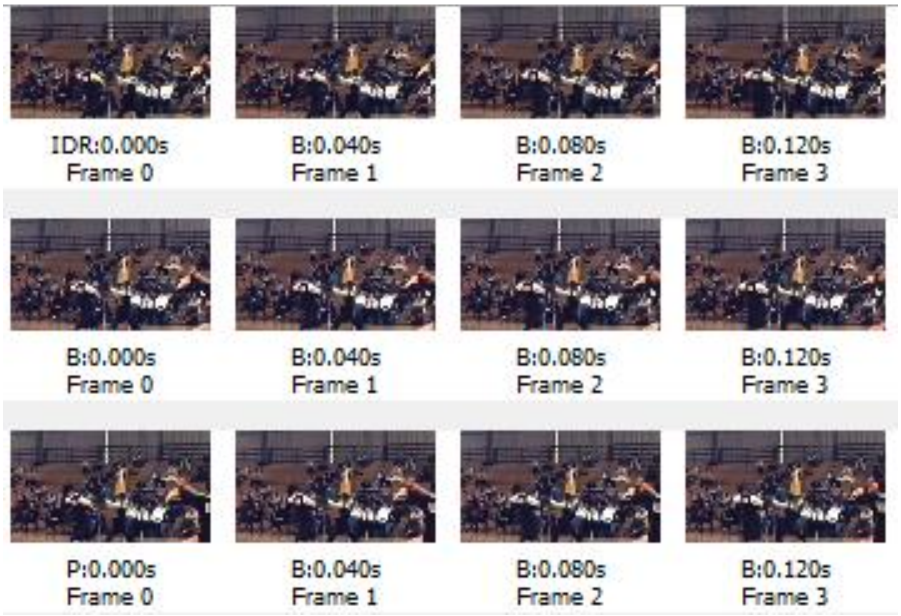
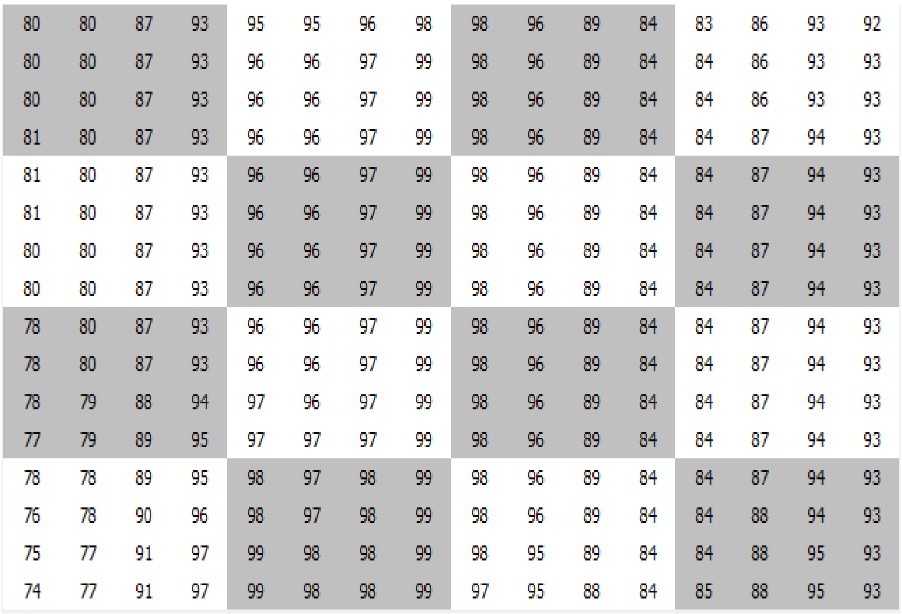


Figure 5.4: Frame layout for spatial and temporal view



80	80	87	93	95	95	96	98	98	96	89	84	83	86	93	92
80	80	87	93	96	96	97	99	98	96	89	84	84	86	93	93
80	80	87	93	96	96	97	99	98	96	89	84	84	86	93	93
81	80	87	93	96	96	97	99	98	96	89	84	84	87	94	93
81	80	87	93	96	96	97	99	98	96	89	84	84	87	94	93
81	80	87	93	96	96	97	99	98	96	89	84	84	87	94	93
80	80	87	93	96	96	97	99	98	96	89	84	84	87	94	93
80	80	87	93	96	96	97	99	98	96	89	84	84	87	94	93
78	80	87	93	96	96	97	99	98	96	89	84	84	87	94	93
78	80	87	93	96	96	97	99	98	96	89	84	84	87	94	93
78	79	88	94	97	96	97	99	98	96	89	84	84	87	94	93
77	79	89	95	97	97	97	99	98	96	89	84	84	87	94	93
78	78	89	95	98	97	98	99	98	96	89	84	84	87	94	93
76	78	90	96	98	97	98	99	98	96	89	84	84	88	94	93
75	77	91	97	99	98	98	99	98	95	89	84	84	88	95	93
74	77	91	97	99	98	98	99	97	95	88	84	85	88	95	93

Figure 5.5: Pixel information of a typical IDR Picture

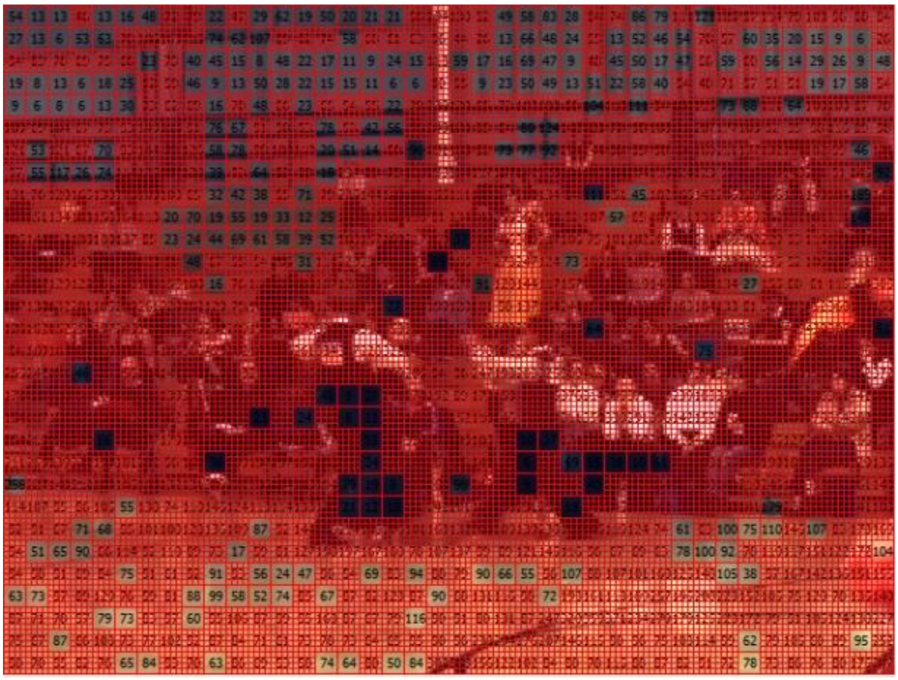


Figure 5.6: Layout of a typical I-frame showing pixel number

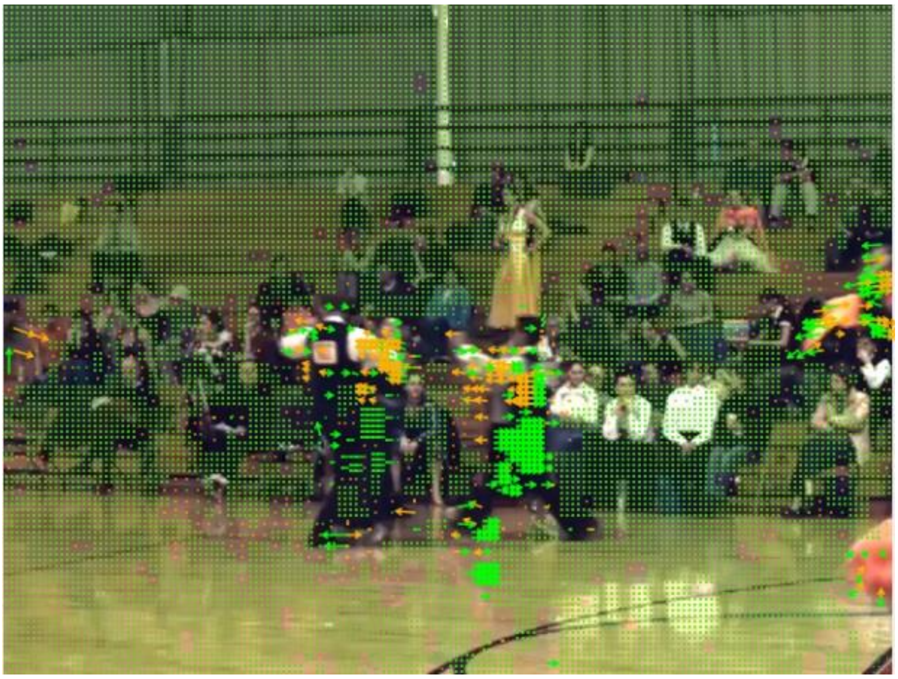
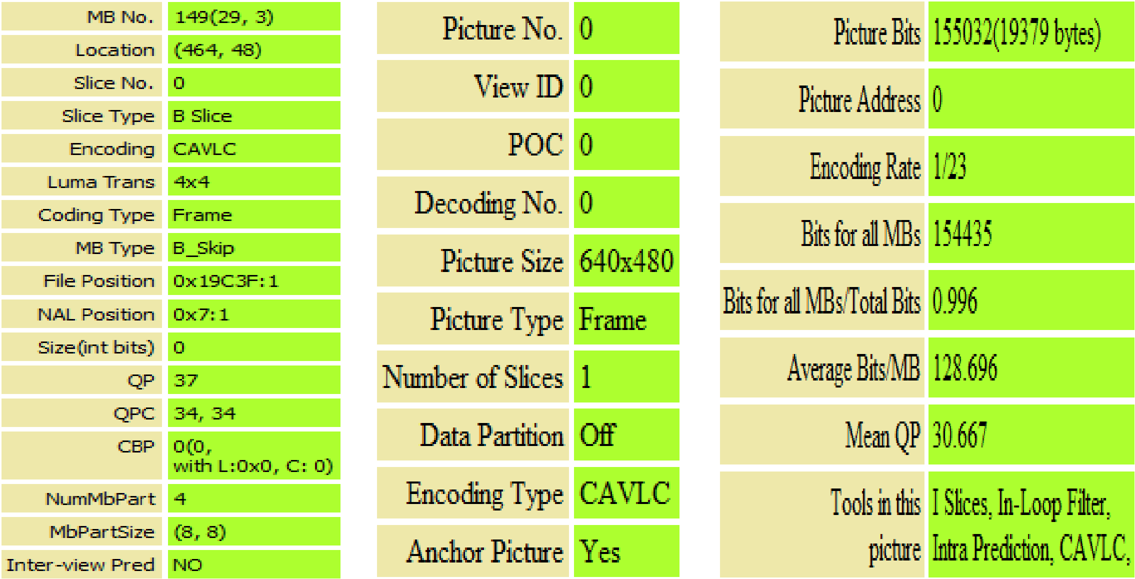


Figure 5.7: Picture layout indicating MVs and directions



MB No.	149(29, 3)
Location	(464, 48)
Slice No.	0
Slice Type	B Slice
Encoding	CAVLC
Luma Trans	4x4
Coding Type	Frame
MB Type	B_Skip
File Position	0x19C3F:1
NAL Position	0x7:1
Size(int bits)	0
QP	37
QPC	34, 34
CBP	0(0, with L:0x0, C: 0)
NumMbPart	4
MbPartSize	(8, 8)
Inter-view Pred	NO

Picture No.	0
View ID	0
POC	0
Decoding No.	0
Picture Size	640x480
Picture Type	Frame
Number of Slices	1
Data Partition	Off
Encoding Type	CAVLC
Anchor Picture	Yes

Picture Bits	155032(19379 bytes)
Picture Address	0
Encoding Rate	1/23
Bits for all MBs	154435
Bits for all MBs/Total Bits	0.996
Average Bits/MB	128.696
Mean QP	30.667
Tools in this picture	I Slices, In-Loop Filter, Intra Prediction, CAVLC,

Figure 5.8: Statistical and coding information of an I-picture

5.1.5. Design condition for Error resilience and concealment in MVC

When designing an error resilient application on a compliant source encoder, the main design goal should include a fair trade-off between coding efficiency and quality. Conceptually, additional bits are introduced into the video bitstream to improve on the video quality performance when affected by errors in a transmission channel. A main problem with error resilient encoders that are designed to overcome the effects of transmission errors is that, they are less efficient when considering compression gain. Typically, they operate with more redundant bits in order to achieve the same level of perceptual quality even when operating in an error free environment. It is understood that, while error resilience techniques focus on introducing extra bits into the video bitstream, video compression techniques focus on the removal of various redundant bits from the same video bitstream. In this context, it is important to carefully trade-off between compression efficiency and error robustness and to also identify the type of video application that will be suitable for a particular error resilient tool. In this work, we perform experiments on the use of data partitioning technique for MVV over packet erasure networks. Another design consideration for the error resilience technique is the operating channel. We know that error prone environments can typically have very diverse characteristics, which can vary over time, it is necessary and important to consider an error resilience application that will be network-aware and can easily adapt to the varying nature of the network [228]. For this reason also, we chose to experiment and simulate data partitioning technique over a validated simulated network with varying network conditions to examine these effects. In practice, transmission errors can inevitably overcome the resiliency of a bitstream and make way up to the decoder. This can be a situation where some parts of a decoded frame in the reconstructed videos are still subject to errors. In this situation, a post decoder technique is necessary and required in order to improve on the quality performance of the video. Error concealment approach tries to make the effects of transmission errors less visible in the decoded video sequence. Technically, the employed error concealment technique targets the restoration of corrupted video slices in the decoder from already reconstructed slices of an error free region of the video sequence [154].

5.1.6. Relationship between packet losses and bit errors

There is a common misconception between packet losses and bit errors, which will be briefly discussed in this section. The H.264 advanced video coding achieves high compression gain by removing statistical and subjective redundancies in a video sequence. This concept employed the use of variable length codes (VLC) in order to generate a string of bits that represents the coded data. Packet loss may occur as a result of buffer overflow in a wireless network, while bit error in an error prone channel like the wireless network may affect the multiview bitstream by either flipping or deleting a bit. Bit error may also occur from the channel as random or burst error within the bitstream. The presence of bit errors in a coded bitstream can increase the bit error rate but affects the packet loss rate differently [229]. Bit error may be seen as packet loss depending on how the video decoder targets and handles error in the bitstream. Depending on the type of codec or design, if the video decoder discards part of the video data that represents a slice as a result of bit error, then in this situation a bit error can be assumed to be equivalent to a packet loss. If the decoder is not capable of processing corrupted data, then bit errors may be present in the bitstream during decoding. This may also result to synchronization failure between the video encoder and the decoder. Since it is difficult for the decoder to locate the exact location of the bit error during the VLC decoding, [230] all the video data will be discarded until the next header information is received by the decoder. When bit errors are handled in this manner, they may be considered as packet losses.

5.1.7. Quality of Service (QoS) and Quality of Experience (QoE) for MVV

Quality of Service and Quality of Experience are very important terms and parameters in 3D video communication system. The two parameters are very useful and required in order to describe the efficiency of the network system and provide a measure of how good the video content can be presented respectively. Quality of Service in a network can be considered as when the network system is capable of offering efficient service and delivery to the users [231], while the Quality of Experience is a description of the end user’s satisfaction based on how he/she perceives the video content presented. The two parameters are closely related to each other in a way that when the best QoE is required in a cost effective and efficient way, then the network system or service providers must ensure an optimal QoS in the network that can operate reliably and efficiently.

However in many cases, good network QoS will lead to better QoE. On the other hand, satisfying all the network traffic conditions may not guarantee user satisfaction. A network with excellent throughput might not be useful to a user if the coverage is not within reach. In general, what is important and economical is good QoE, and the bottom line of any service provider or network QoS should be the provision of high QoE.

5.2. Simulcast versus MVC Experiment

This section presents an experimental study that is conducted with the H.264/AVC reference software to demonstrate the coding efficiency between simulcast video coding and multiview video coding. The H.264/AVC reference software is used in this experiment to encode the three standard multiview test videos namely Ballroom, Vassar and Exit. The resolution of each test sequence is 640x480 pixels and number of frames encoded for each video is 125 frames at 25Hz, which makes 5 seconds video play time. Hierarchical B-coding is used with the coding format of IBBBBBBBPBB… and the number of reference is set to 2. Quantization Parameter (QP) is varied to different values of 28, 31, and 34.

5.2.1. Experimental Results

In this experiment, different simulations are performed and the numerical values for each test sequence ballroom, vassar and exit are generated in a tabular form as shown in table 5.4, 5.5, and 5.6 respectively. From the experiment conducted, a bit rate saving of up to 24% is recorded when stereo coding is used as it utilizes the interview redundancies.

5.2.1.1. Objective and Subjective results

From the objective results obtained in this experiment, it can be observed that the bitrate saving for stereo coding has significantly improved compared to simulcast coding of the H.264/AVC video codec. Higher coding efficiency can be achieved with less bitrate when redundant frames are used from other views in MVC.

Table 5.2 Quality performance and bitrate saving for Ballroom

Ballroom Sequence								
	Single View Coding		Simulcast Coding		Stereo Coding			
QP	PSNR (dB)	Bitrate (Kb/s)	PSNR (dB)	Bitrate (Kb/s)	PSNR (dB)	Bitrate (Kb/s)	Bitrate Diff. (%)	PSNR Diff (%)
28	38.53	1088.33	38.49	2218.65	38.55	2139.78	3.55	-0.06
31	37.08	727.7	37.06	1479.3	37.14	1415.78	4.29	-0.07
34	35.53	484.44	35.53	985.36	35.63	938.61	4.74	-0.09

Table 5.3 Quality performance and bitrate saving for Vassar

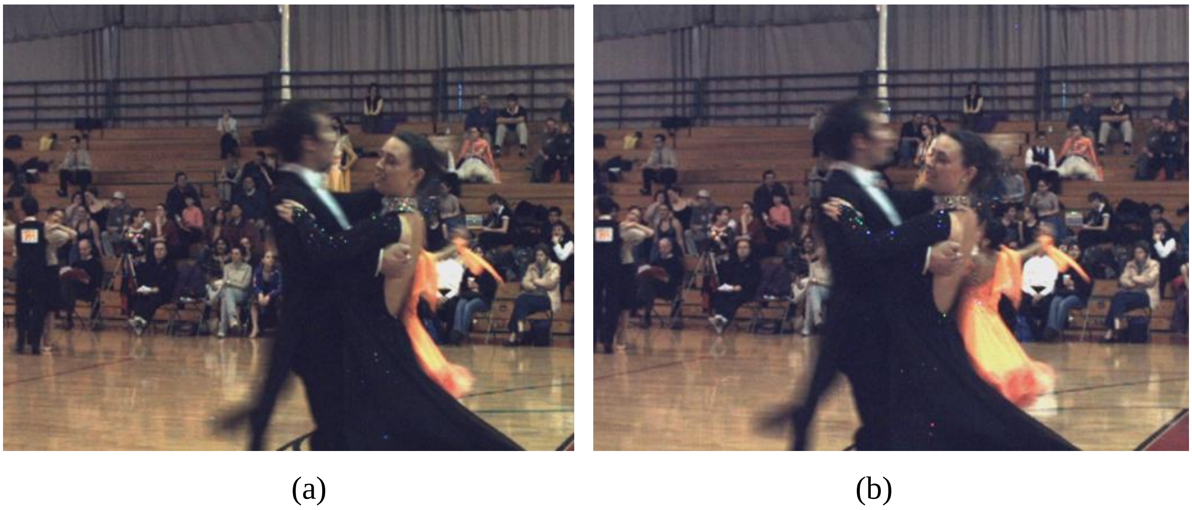
Vassar Sequence								
	Single View Coding		Simulcast Coding		Stereo Coding			
QP	PSNR (dB)	Bitrate (Kb/s)	PSNR (dB)	Bitrate (Kb/s)	PSNR (dB)	Bitrate (Kb/s)	Bitrate Diff. (%)	PSNR Diff (%)
28	38.21	289.99	38.2	582.59	38.55	447.98	23.11	0.28
31	37.19	142.15	37.22	280.03	37.14	234.05	16.42	0.26
34	36.14	79.53	36.18	155.48	35.63	137.15	11.79	0.25

Table 5.4 Quality performance and bitrate saving for Exit

Exit Sequence								
	Single View Coding		Simulcast Coding		Stereo Coding			
QP	PSNR (dB)	Bitrate (Kb/s)	PSNR (dB)	Bitrate (Kb/s)	PSNR (dB)	Bitrate (Kb/s)	Bitrate Diff. (%)	PSNR Diff (%)
28	40	336.4	39.95	692.58	39.7	634.9	8.33	0.25
31	38.91	207.39	38.88	422.78	38.6	394.96	6.58	0.28
34	37.68	133.2	37.67	271.42	37.37	263.52	2.91	0.29

Similarly, the quality of the reconstructed videos is not affected in a considerable manner and is negligible. Frame 63 of the left and right views for ballroom sequence is shown in Fig. 5.9. The objective results obtained from the experimental study are in Fig. 5.10, Fig. 5.11 and Fig. 5.12, which shows the performance of the test sequences in terms of bitrate saving and quality when the single view, simulcast, and stereo video coding are used respectively. As anticipated, simulcast coding results to almost twice the bitrate of single view coding with minimum loss in quality.

However, coding the test videos with stereo coding format results in considerable reduction in bitrate especially in vassar test sequence. The reduction in bitrate is computed in percentage as shown in tables 5.4, 5.5, and 5.6 for the different test sequences used. The bitrate reduction observed in ballroom is not as significant as the vassar test sequence. Because ballroom video has a lot of motion activity and is complex which results in lesser redundancies across the views. In the vassar test video, there is less activity within the scene such as stationary background and moving object. Vassar test sequence is classified as less complex video while ballroom video is a complex multiview test sequence. The exit test sequence is considered to be moderate in terms of complexity with less moving objects within the scene.



(a) (b)

Figure 5.9: Left and Right view of ballroom sequence

Another observation in this experiment is how the QP affects bitrate, in Fig. 5.10, Fig. 5.11, and Fig. 5.12 for the ballroom, vassar and exit test sequences used. It can be seen that higher QP value results in lower bitrate which is suitable for storage or transmission. Normally, the QP value defines spatial details that can be retained in video. For an application that requires very high perceptual quality like movie theatres, small QP can be useful. Whereas for bandwidth constraint applications such as video conference and streaming, higher value of QP may be required. It depends on the application, but generally, a trade-off between cost of bitrate and quality is required.

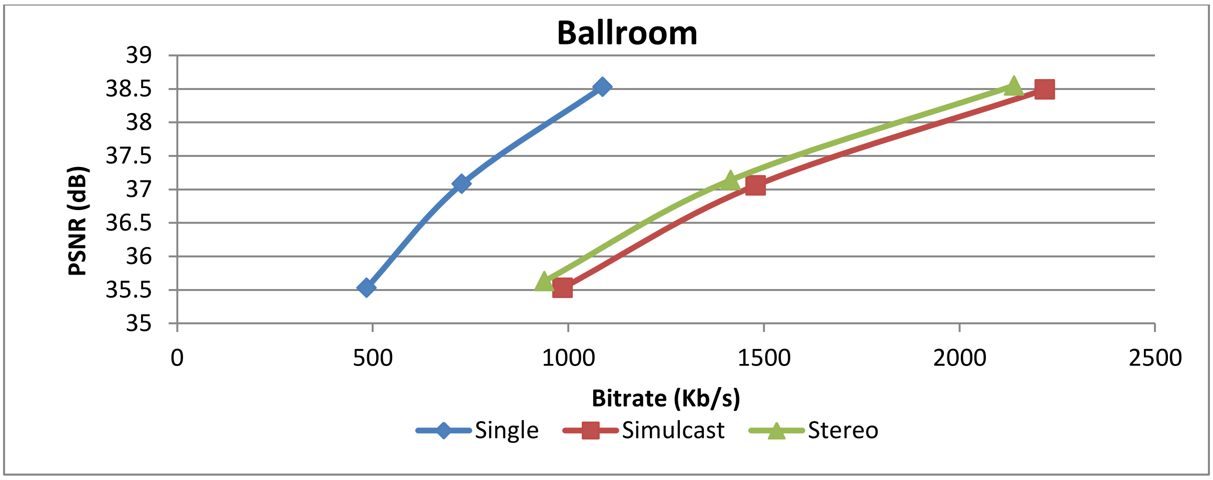


Figure 5.10: Bitrate performance and reduction for Ballroom

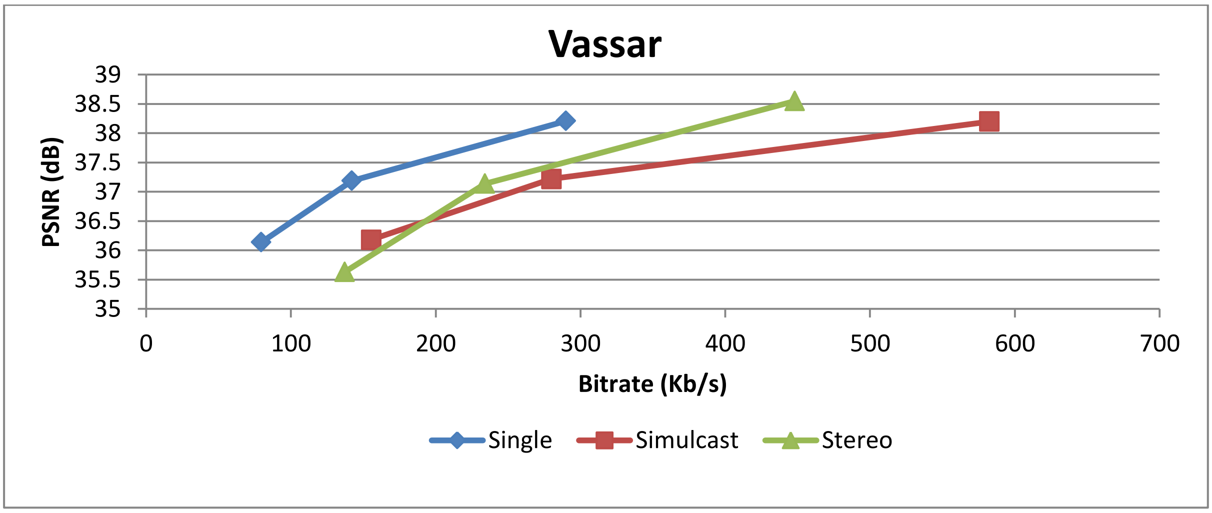


Figure 5.11: Bitrate performance and reduction for Vassar

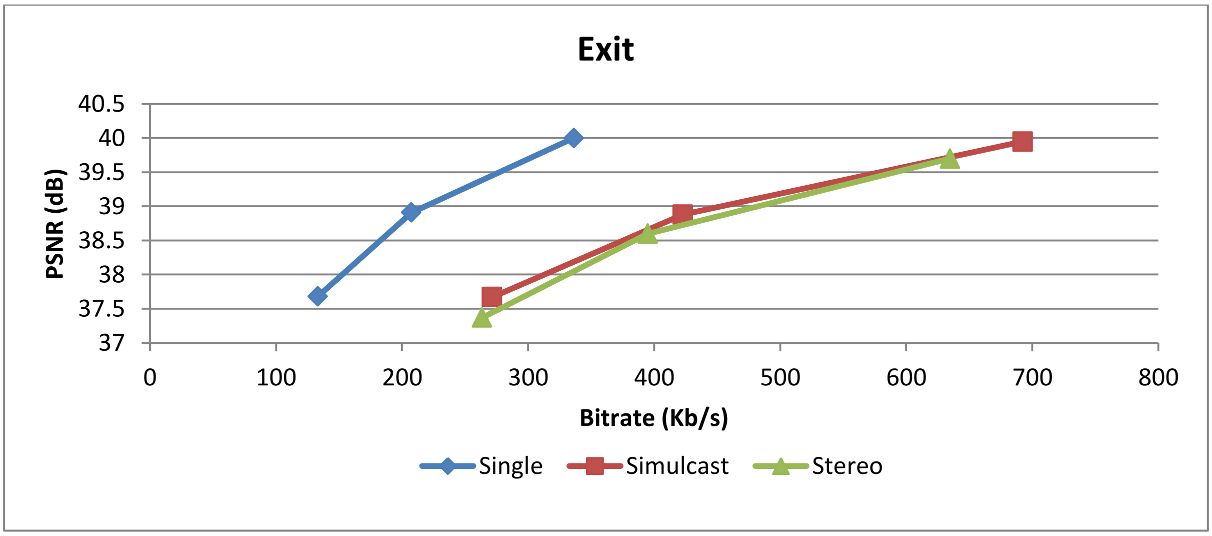


Figure 5.12: Bitrate performance and reduction for Exit

A comparable video quality is achieved between the simulcast video coding and single view coding, while the stereo video coding has lost almost negligible quality in order to achieve a higher bit rate saving. In this experiment, it is computed that the maximum quality loss is -0.3dB in exit test sequence for QP = 34. This value does not show noticeable effect in the subjective view.

5.2.2. Analysis and Discussion

Two distinct techniques for 3D multiview video are investigated in this experiment and conclusive results are obtained. From the simulcast coding where each view is coded separately, it only utilizes the temporal redundancies. This concept is straight forward and simple and does not exploit the redundancy between the two views involved. Therefore, the coding efficiency of simulcast video coding is low and may not be suitable for bandwidth efficient applications such as video streaming or transmission. In 3D stereo coding, both the temporal and interview redundancies are exploited, which from the results obtained improve on the compression gain by up to 24% in vassar MVV sequence compared to simulcast. This development is achieved without any noticeable effect on the perceptual quality of the reconstructed views.

5.3. Conclusion

This chapter describes in detail the experiment and simulation setup involved in the research work. The necessary coding parameters and encoder settings that are used in the JMVC 8.5 reference software are described and analysed. Error resilient design conditions in MVC and some key coding parameters that affect multiview video coding are highlighted and discussed. In addition, the chapter describes multiview video bitstream and the other tools that have been used in the research work. The network simulation setup and analysis is also described in this chapter. Sirannon network simulator is employed to validate the network simulation and the Gilbert loss model is used to generate transmission losses in the MVV. The network design architecture and key components that affect the simulation are also reported. In the network simulation, the design parameters are chosen to simulate a real network scenario in order to achieve optimal results. Some experimental results that are presented shows that the bitrate saving for stereo coding has significantly improved compared to simulcast coding of the H.264/AVC video codec. It can be concluded that higher coding efficiency can be achieved with less bitrate when redundant frames are used from other views in MVC. The chapter also introduces some key network simulation aspect of the work, which includes the relationship between bit errors and packet losses in a MVV coded sequence.

6. Chapter Six: Multi-Layer Data Partitioning

6.1. Implementation of H.264 DP in MVC

This section provides a comprehensive description of the stages involved in the implementation of the proposed error resilience technique. Since MVC is based on the H.264/AVC algorithms, the encoding scheme of the MBs is very similar to that defined by H.264/AVC standard. Note that, the H.264/AVC standard presents the coded video information as a slice, which can be illustrated as in Fig. 6.4. Implementation of DP technique in the JMVC 8.5 reference software encoder causes additional complexity and sudden failure as the number of frames increase. Instead a different approach employed is to develop an application that will parse the MVV bitstream generated by the MVC bitstream assembler. The algorithm separates the coded slice elements into three different partitions in the base view and non-base view and is developed strictly based on H.264/AVC syntax elements as defined [25]. Fig. 6.1 illustrates this approach on how the MVV bitstream can be partitioned into a more resilient structure and mechanism. It is less complex and does not significantly alter the overall bit rate of the MVV bitstream. The DP technique is designed to provide resilience to channel errors and packet loss that may occur during transmission and also to enhance the error protection scheme. The algorithm first reads a NAL unit from the bitstream by looking at the NAL unit header, and then it parses the NAL unit header to determine its type. In MVV bitstream, there are usually two sequence parameter sets (SPS), one for the base view decoding and the other for the decoding of all the other views. If the type of NAL unit read by the algorithm is an SPS and its profile_idc (profile of a bitstream) is Main Profile then it changes it to Extended Profile and writes it to the output bitstream. For NAL units whose type is PPS or Instantaneous Decoder Refresh (IDR), no change is made. All these are written as they are to the output stream. NAL units, whose type is either CODED_SLICE or CODED_SLICE_SCALABLE goes through a partitioning process. During partitioning, the slice header is parsed and written to DP A. Since a slice consists of an integer number of MBs, all the macro block meta data i.e. mb_type, motion vectors, quantization parameter, intra prediction modes, is written to DP A. Coded coefficients for intra MBs are written to DP B and coded coefficients for inter MBs are written to DP C. This can be further illustrated as in Fig. 6.5

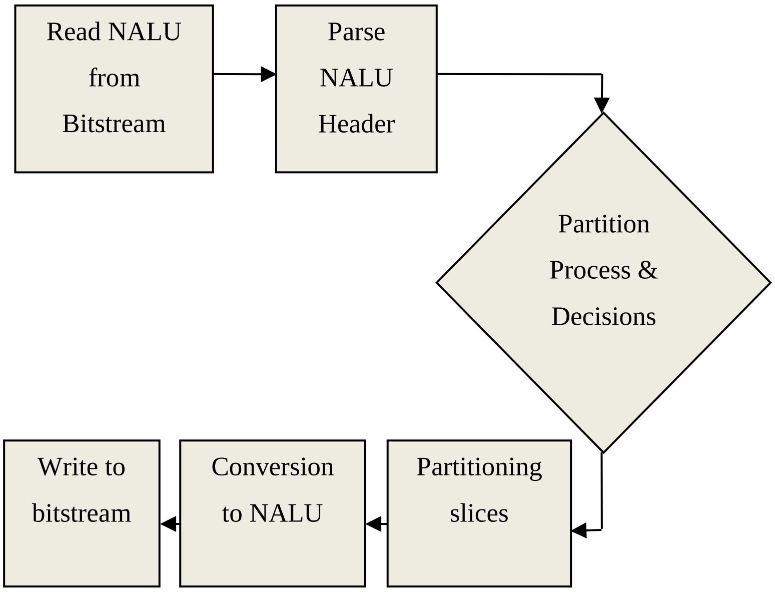
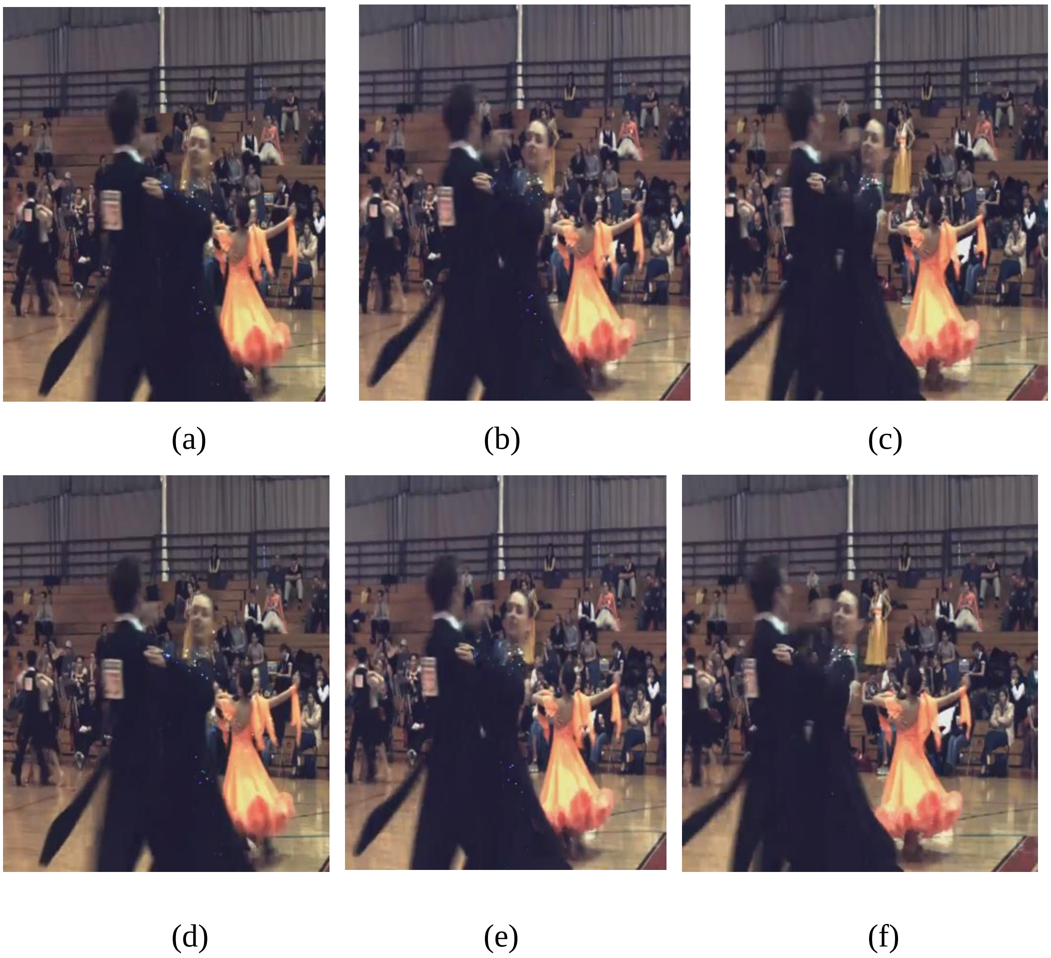


Figure 6.1: Flow diagram of the Data Partitioning model

Fig. 6.1 shows a simple flow diagram of our proposed algorithm. The technique introduces three new NAL units in the MVV bitstream to differentiate between the base view slices and non-base view slices and to support the decoding of the modified MVV bitstream. The new NAL units introduced are for the non-base views. For NAL_UNIT_CODED_SLICE_SCALABLE, it is divided into three partitions that represent different NAL unit types which are used to store different video data of the non-base view. The data partitioning bitstream is then fed to the MVC decoder and the decoder is modified to accept the changes made in the MVV bitstream and to decode all frames in the MVV bitstream. Finally, all the views were successfully reconstructed by the modified JMVC decoder. Fig. 6.2 illustrates the reconstructed views with and without data partitioning, where (a), (b), and (c) represents frames number 56, 57, and 58 for view 0, view 1, and view 2 respectively of the ballroom sequence without data partitioning and (d), (e), and (f) represents frames number 56, 57, and 58 for view 0, view 1, and view 2 respectively of the ballroom sequence with data partitioning. It can be observed that the reconstructed frames from H.264 with data partitioning technique are almost identical with the original frames as expected. This is an indication that the implementation of H.264 data partitioning technique in JMVC reference software is successful.



(a) (b) (c)

(d) (e) (f)

Figure 6.2: Frames samples from original ballroom sequence and H.264 DP

6.2. Multi-Layer DP Technique for MVC

In an attempt to make the MVV bitstream more robust to errors encountered in an error prone channel, the multi-layer DP technique is proposed which can create another layer of partitioning for each frame in the MVV bitstream. The general architecture of the technique is illustrated in Fig. 6.3.The MVC bitstream is parsed in the Multi-Layer DP application for increased robustness against channel errors before sending over a wireless network. The multiview video bitstream is received by the modified JMVC reference decoder in order to decode and reconstruct the multiview video bitstream for 3D viewing.

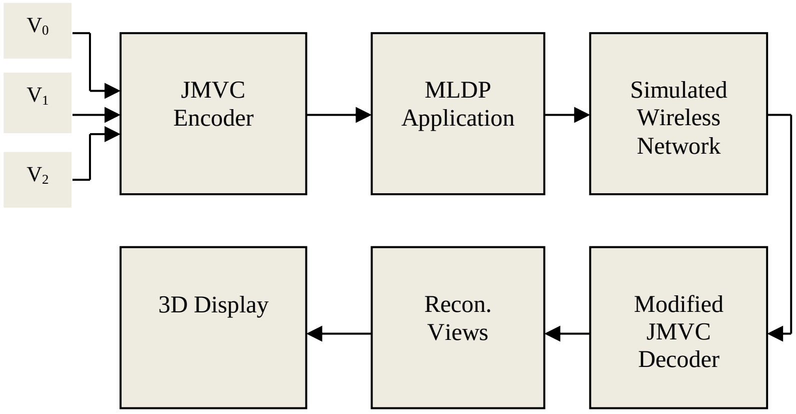


Figure 6.3: Architecture of the multi-later DP technique

Multi-Layer DP adopts a mechanism that restructures a video slice as shown in Fig. 6.6. A_0 partition consists of the header information of frame 0 from view 0, and A_1 partition consists of the header and motion information of frame 1 from view 1 and A_2 partition consists of the header and motion information of frame 2 from view 2. B_0 consists of the residual information of intra coded MBs of frame 0, B_1 consists of the residual of intra coded MBs in frame 1 and B_2 consists of the residual of intra coded MBs of frame 2 and C_0 is an empty partition, C_1 consists of residuals of inter coded MBs of frame 1 and C_2 consists of the residual of inter coded MBs of frame 2 and in that sequence it continues till nth view and nth last frame of the multiview bitstream.

Slice Header	**Payload**

Figure 6.4: Slice Layout in H.264/AVC

A	**B**	**C**

Figure 6.5: H.264/AVC Slice layout with data partitioning

A0	**A1**	**A2**	**B0**	**B1**	**B2**	**C0**	**C1**	**C2**

Figure 6.6: Multi-Layer data partitioning technique

Note that, partition C_0 is empty because there is no residual information of inter-coded MB's in frame 0 which is an intra-coded frame. I-frames are self-referencing and do not require any sort of information from other frames to be predicted, so they consist of only intra coded MBs. The source code for the implementation can be found in appendix B1. The H.264 compliant encoder needs not send empty partitions to the decoder because a standard H.264 decoder will assume missing partitions are empty partitions and are designed to handle the multiview bitstream accordingly [163]. During the decoding process, the decoder is modified to accept the MLDP bitstream and cope with the lost video data due to errors in the wireless channel. The mechanism and the design process adopted to deal with this problem are explained in detail in the next section and in appendix B2. The effects of displaying a frame reconstructed from corrupted data can adversely degrade visual perception by introducing artefacts. In order to support the MLDP technique more effectively and to minimise the effects of channel errors, a simple frame copy error concealment scheme is employed. Frames that are generated by copying related video data in order to replace lost information are not always perceptually noticeable by a viewer which is an advantage of this technique especially in low-activity scenes [232]. Frame copy error concealment works fairly well with multiview video bitstream and is simple to implement; however, there are more complex techniques that use an elaborate approach to exploit the redundancy within the video frame in order to come up with a more efficient estimate of the lost data [233].

6.3. Proposed Decoding scheme for MVC Erroneous Bitstream

The H.264/AVC frame copy error concealment technique is implemented in the JMVC reference decoder and further modified to decode the Multi-layer DP bitstream with losses. The technique is optimized to reconstruct all the views successfully from the multiview coded bitstream with a higher level of quality in compliance with the standard. Part of the reasons and motivation to adopt frame copy error concealment technique in this work is its convenience to replace missing pictures especially in the case of packet loss network. The flowchart in Fig. 6.7 illustrates the implementation of frame copy concealment algorithm in the JMVC reference software. The algorithm can conceal lost information in the MVV bitstream with an improved perceptual quality as shown in the experimental results of section 6.11. Technically, missing slice is copied from a reference list and good visual quality is obtainable if there is no motion inside the GOP.

However, the proposed algorithm can provide fairly good EC for low motion applications. This will be demonstrated later in section 6.12. When the ML data partition bitstream is transmitted over the network and is received, it is first buffered and rescheduled back to the standard H.264 DP format for processing. Note that, the multi-layer data partitioning technique employed during source coding is only to make the multiview video bitstream more resilient to channel errors during transmission or streaming over the simulated wireless network. After successfully delivering the bitstream across the network, the received bitstream is rescheduled back to the standard H.264 data partitioned format for decoding. The decoder checks if the buffer is full then all the frames are sent directly for decoding.

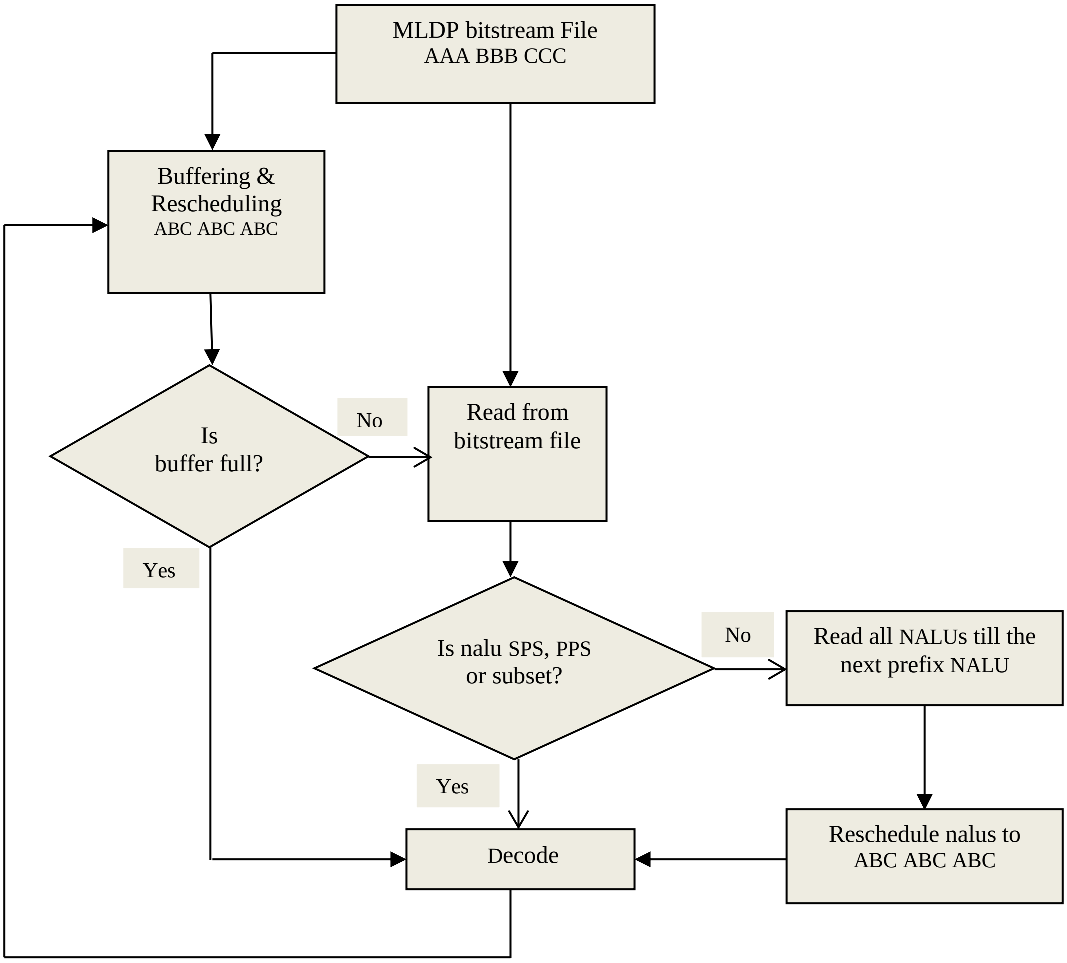


Figure 6.7: Proposed decoding scheme for erroneous MVC bitstream

Note that all the slices are partitioned into three different partitions encapsulated into VCL NAL units of DP A, DP B and DP C respectively. The decoding of these types of slices is such that the loss of one partition might make another partition useless. In order to correctly decode partitions B and C, it is important for the H.264 standard compliant decoder to know how each and every macroblock is predicted within a slice. This information is stored in partition A as part of header information. Therefore, loss of partition A can render partitions B and C useless even when correctly received and decoded. Partition A does not necessarily require the information from partition B and C to be correctly decoded. So, if only partition A is correctly received then the error concealment algorithm can utilize useful information such as motion vectors to reconstruct the slice. However, if partition A is lost regardless of whether partition B or/and C is/are received, the algorithm is invoked by the decoder to replace the missing picture information by a previously received picture in the reference list. If the buffer is empty, then the NAL units are read from the MVV bitstream and the decoder determines whether it is a non-VCL NAL unit or VCL NAL unit? All non-VCL NAL units are sent directly for decoding while the VLC NAL units are all read until the next prefix NAL unit is detected and are rescheduled to the H.264 format before decoding. The whole process is restarted again through a loop system.

Equation (6.1) below computes the pixel value during motion compensation.

$$E_{x,\,y} = I_{x,\,y} - P_{x,\,y} \qquad (6.1)$$

Therefore, the pixel value or reconstructed value can be expressed as

$$I_{x,\,y} = E_{x,\,y} + P_{x,\,y} \qquad (6.2)$$

Where $I_{x,\,y}$ is the pixel value and $P_{x,\,y}$ is the predicted value, and for each pixel, residual error $E_{x,\,y}$ is calculated. The values of x, y gives the coordinates of the variables, namely pixel, predicted, and residual error respectively. The predicted value can be obtained from the motion vectors (in the case of inter coded MB) or intra prediction (in the case of intra predicted MB). We know that motion vectors and intra predicted modes are placed in partition A. The residual information is placed in the form of transform coefficients for intra-coded and inter-coded MBs in partition B and C respectively.

When the residual information is lost, then

$$E_{x, y} = 0 \qquad (6.3)$$

pixel value becomes

$$I_{x, y} = P_{x, y} \qquad (6.4)$$

Because some part of the video data is lost in the form of residual information, the effect on the reconstructed video is usually indicated by grey scales around the pictures.

Table 6.1 Bitrate comparison between the techniques for different sequences

Sequence	**H.264 (Kb/s)**	**H.264DP (Kb/s)**	**MLDP (Kb/s)**	**Diff. (Kb/s)**
Ballroom	1515.95	1527.94	1527.94	11.99
Exit	688.86	700.24	700.24	11.38
Vassar	679.75	691.68	691.68	11.93

From table 6.1, it can be observed that there is no bit rate increase in the Multi-Layer DP technique when compared to H.264 DP. A small increase of up to 0.8% (percentage increase) in bit rate between a standard H.264 encoder with no DP and H.264 encoder with DP is recorded. This increase is reasonable for MVV bitstreams, because an extra four byte NAL unit header is required for MVC NAL unit according to Annex H standard of H.264. One byte NAL unit header for the base view and 3 byte NAL unit header for the non-base view and few more bits are added to the slice identification syntax element in each partition. Another reason for the increase in bit rate is trailing bits at the end of some partitions for byte alignment.

6.3.1. Objective Quality Evaluation

The graph in Fig. 6.8 represents the performance evaluation for ballroom sequence of the Multi-Layered DP and the H.264 DP technique. The performance of each view for different error rates is considered. Ten different network simulations are carried out for each error rate, and the average value is computed for each of the MVV bitstream. The quality of each reconstructed view is measured in terms of PSNR for error rates 0%, 1%, 5%, 10%, 15%, and 20% respectively. It can be observed from the objective results obtained that Multi-layer DP technique can improve the quality of the multiview video on each view objectively for the different error rates considered.

Three views are considered for simplicity and each view of the multi-layer technique is compared with the view of H.264 DP technique as a benchmark. Similarly, in Fig. 6.9, the average PSNR value of the ballroom sequence is obtained and plotted for each technique. It is obvious that Multi-Layer DP technique has better error robustness capability compared to the H.264 DP technique for ballroom sequence. Fig. 6.10 and Fig. 6.11 demonstrate the performance of the exit sequence, the multi-layer DP technique has improved the quality of the reconstructed video on each view and when compared to the H.264 DP method. The overall or average performance of the test sequence is also improved. Fig. 6.12 and Fig. 6.13 illustrate the objective performance for the vassar test sequence. The multi-layer DP technique can improve the video quality for different error rates than the H.264 DP technique. This is true for all the reconstructed views. It is obvious from the objective results obtained that for all the three test sequences, Multi-Layer DP can improve the quality performance of the test videos than the H.264 DP technique.

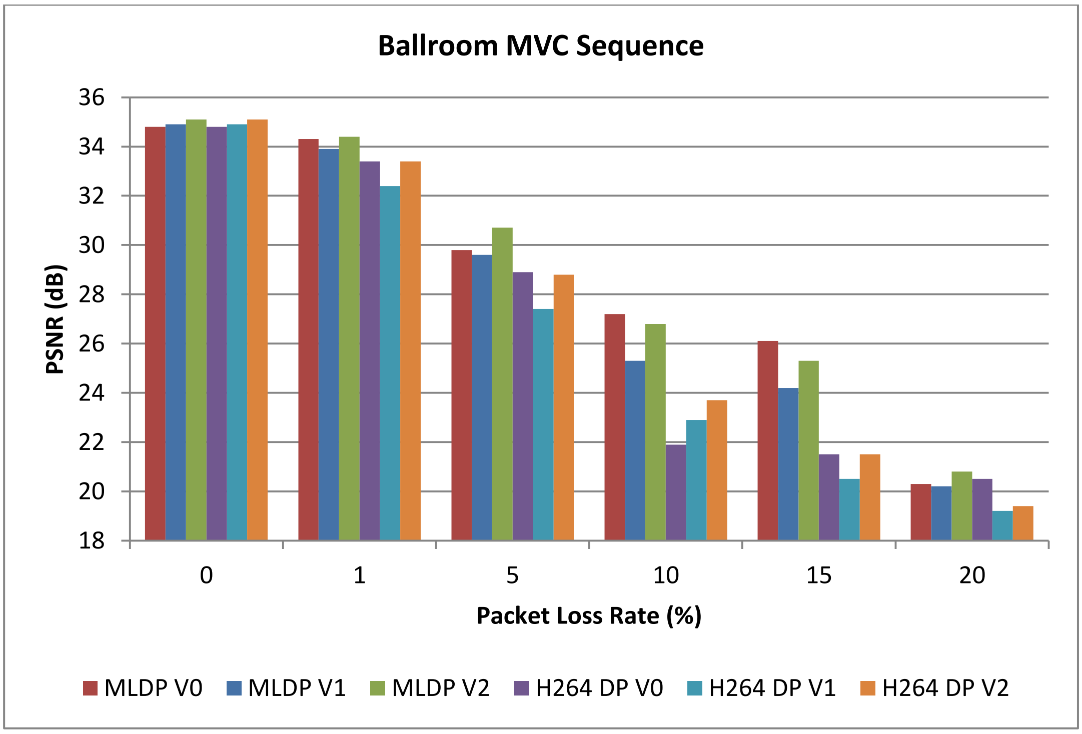
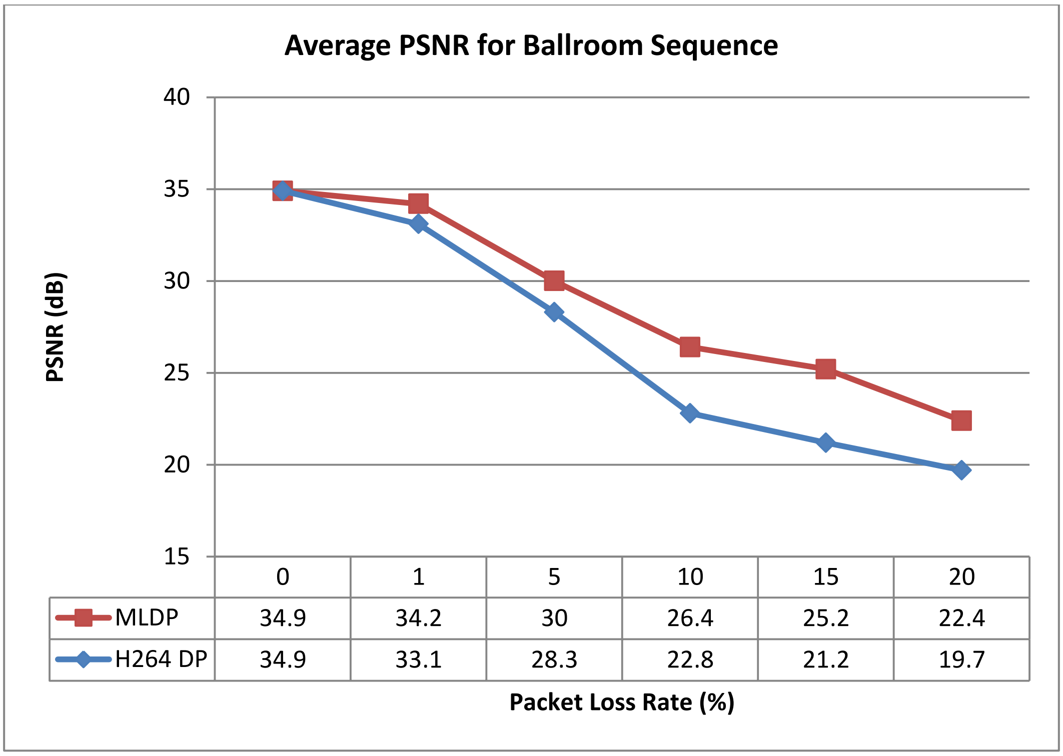


Figure 6.8: PSNR of different views for ballroom



	0	1	5	10	15	20
MLDP	34.9	34.2	30	26.4	25.2	22.4
H264 DP	34.9	33.1	28.3	22.8	21.2	19.7

Figure 6.9: Average PSNR for Ballroom sequence

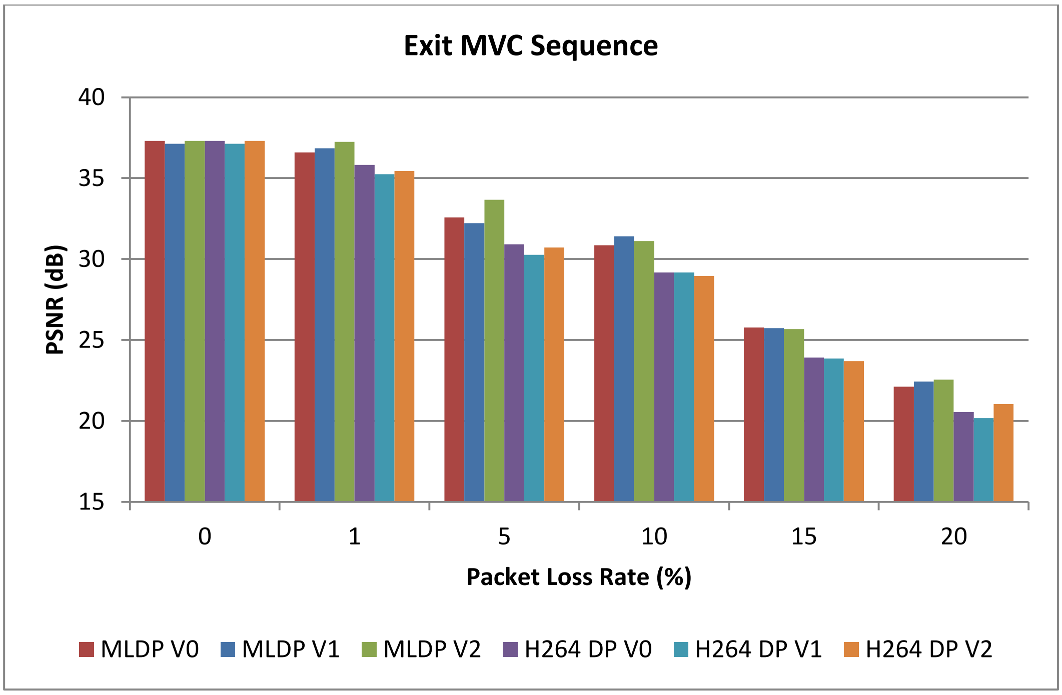
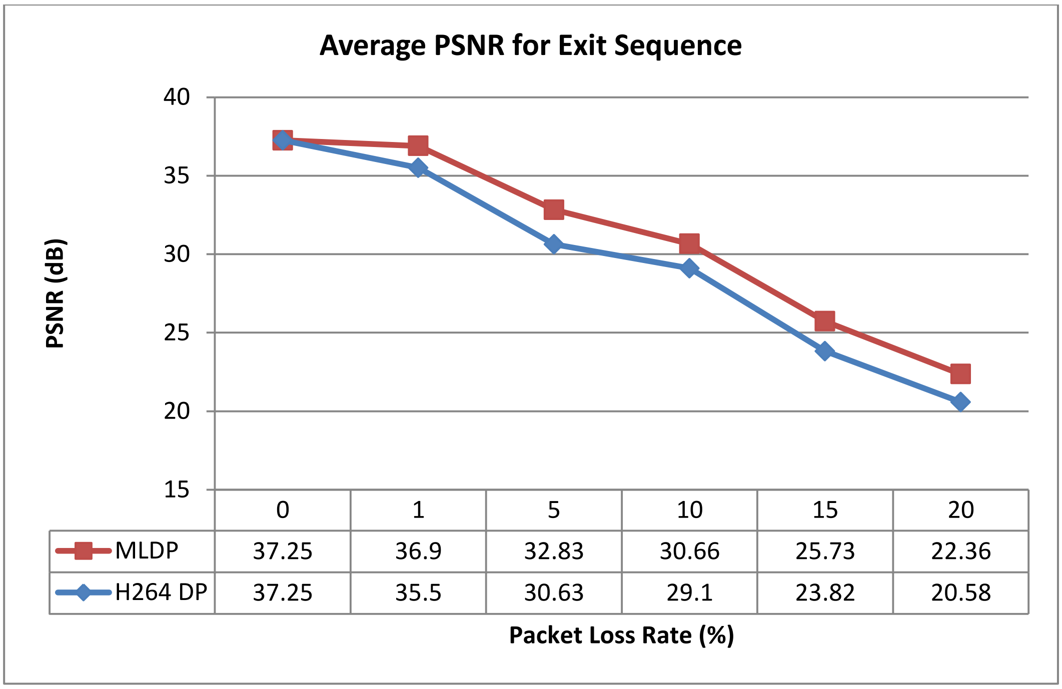


Figure 6.10: PSNR of different views for Exit sequence



	0	1	5	10	15	20
MLDP	37.25	36.9	32.83	30.66	25.73	22.36
H264 DP	37.25	35.5	30.63	29.1	23.82	20.58

Figure 6.11: Average PSNR for Exit sequence

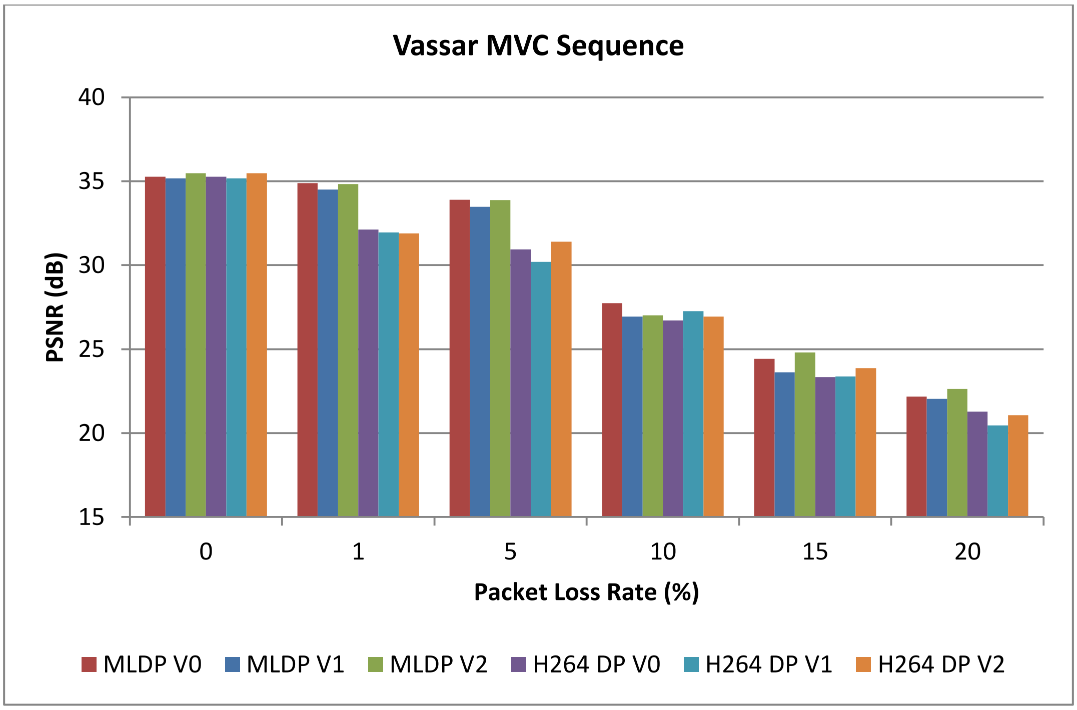
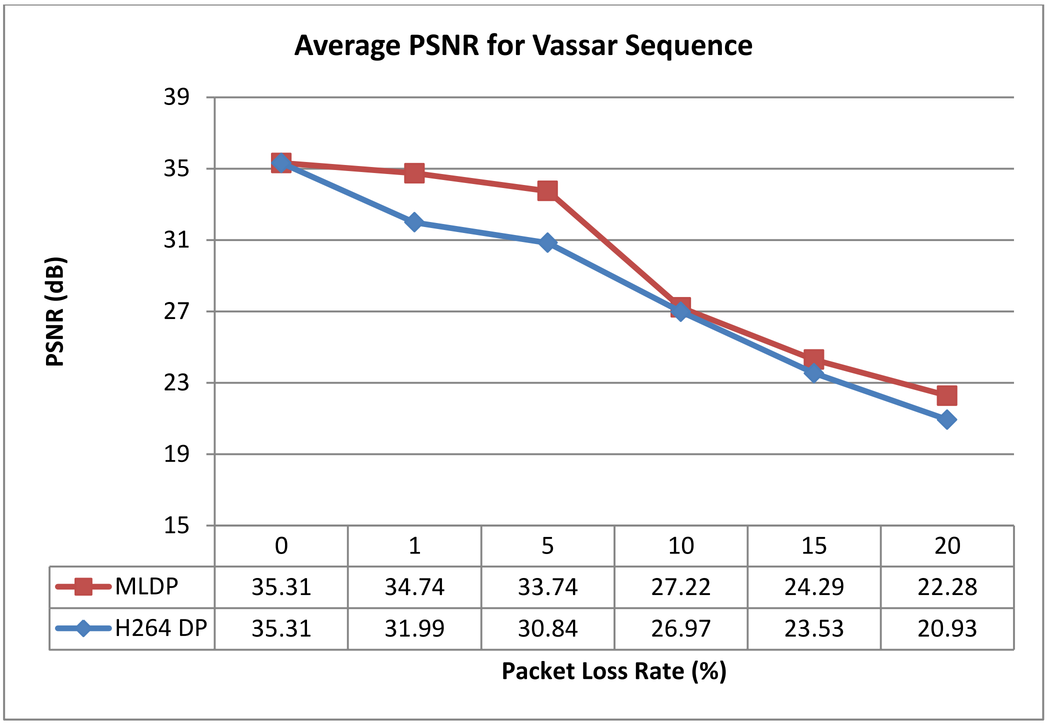


Figure 6.12: PSNR of different views for Vassar sequence



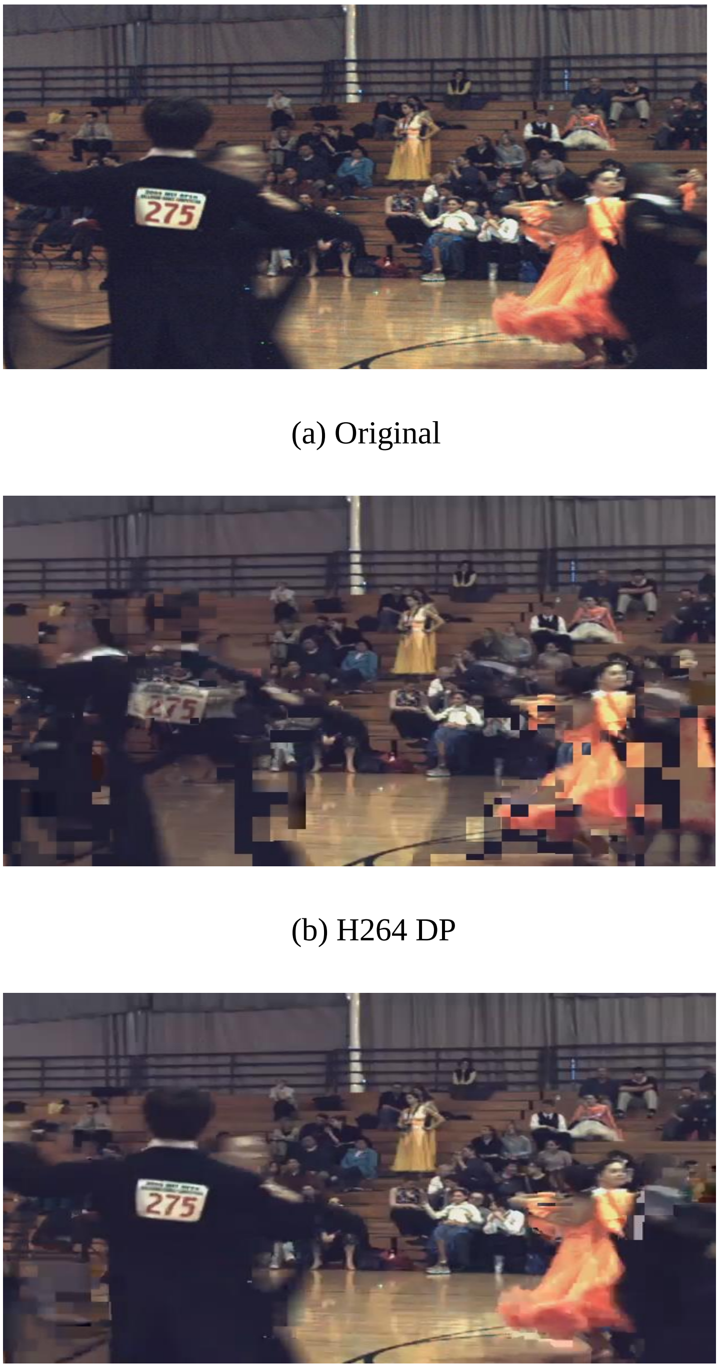
	0	1	5	10	15	20
■ MLDP	35.31	34.74	33.74	27.22	24.29	22.28
◆ H264 DP	35.31	31.99	30.84	26.97	23.53	20.93

Figure 6.13: Average PSNR for Vassar sequence

6.3.2. Subjective Quality Results

The subjective results are presented for different test sequences in this section. In Fig. 6.14, frame 47 is chosen from view 0 at 10% loss rate. The comparison between the original frame without error and the two techniques has shown that the multi-layer DP method is capable of improving the visual quality. In the H.264 DP method, a lot of the video data is lost and the frame could not be reconstructed correctly. While in the multi-layer DP technique, most of the video data that were lost in H.264 DP method are recovered with an improved quality even at 10% loss rate. Nevertheless, a high level of quality could not be achieved with the multi-layer DP technique because of the high motion activity in the scene and the limitation of the error concealment algorithm. The subjective test of exit sequence is depicted in Fig. 6.15. Frame 222 is used for comparison in this analysis which has some moving objects in the scene. The frame for H.264 DP is badly reconstructed as can be seen compared to the reconstructed frame in the multi-layer DP technique at the same 10% error rate. The greyscale effect in frame 222 of H.264 DP technique is as a result of severe loss of residual information that the decoder has lost out on from either type B or C partitions. Meanwhile, it has recovered the corresponding motion or header information in the type A partition.

It is obvious that the loss of residual information in H.264 DP technique has affected the full recovery of the frame information, whereas, in Multi-Layered DP, it is obvious that the recovery of the frame at the same error rate is of a higher level of perceptual quality. Fig. 6.16 depicts the subject evaluation for the vassar sequence. To capture the moving objects in this sequence for comparison and evaluation, frame 175 is used. We can see that the Multi-layer technique is capable of reconstructing the frame with better perceptual quality than the H.264 DP technique. Nearly all the lost video data in the H.264 technique has been recovered with improved quality in the multi-layer technique. The proposed multi-layer data partitioning technique like any other error resilient mechanism in video coding tries to make the multiview video bitstream more resilient to losses. The technique is not designed to recover any lost information as a result of transmission errors or losses in the coded multiview video sequence. Instead, the recovery of lost video data in the multiview video sequence is achieved by the use of error concealment mechanism as previously discussed.

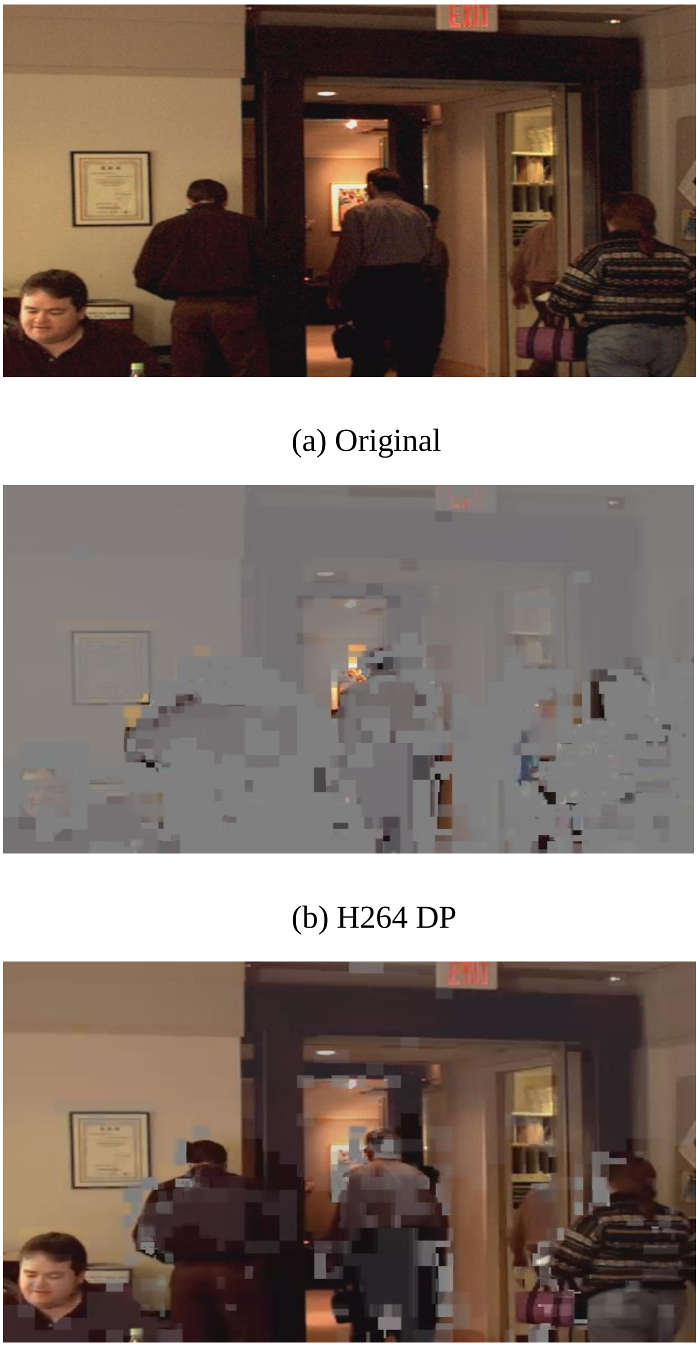


(a) Original

(b) H264 DP

(c) ML DP

Figure 6.14: Ballroom subjective comparison of frame 47 at 10% PLR

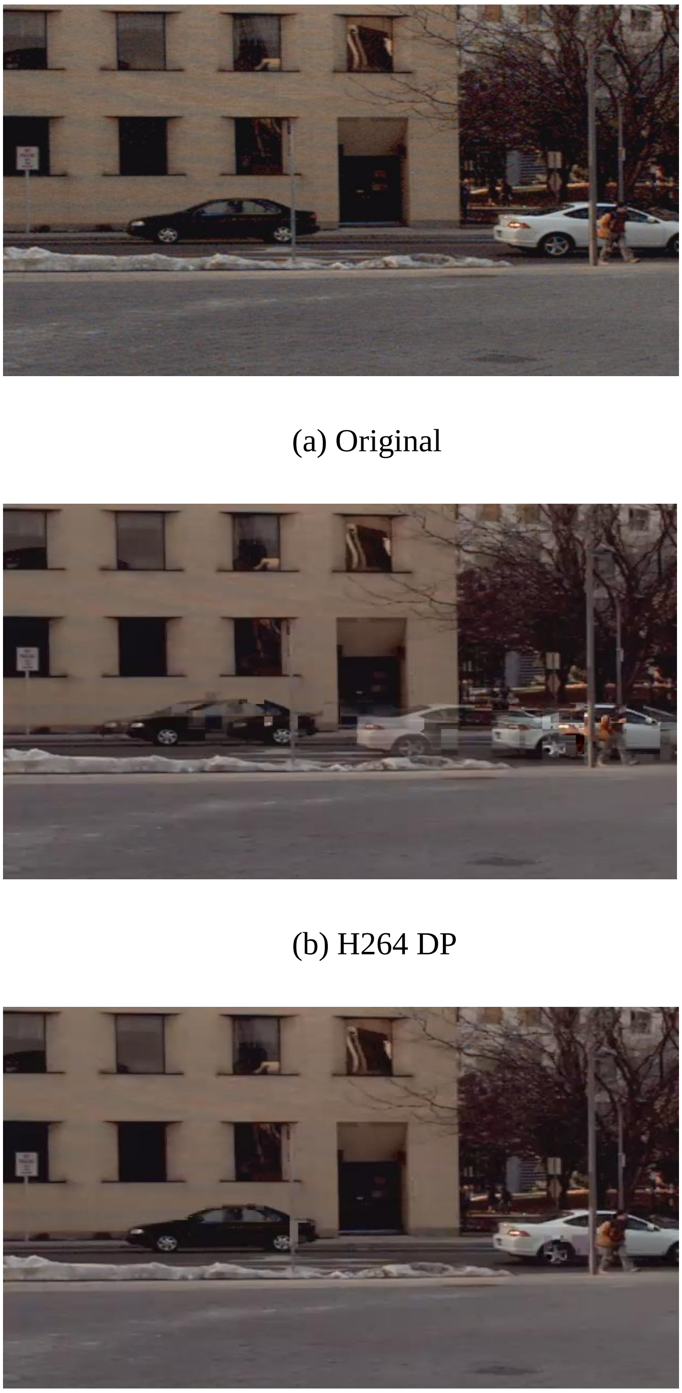


(a) Original

(b) H264 DP

(c) ML DP

Figure 6.15: Exit subjective comparison of frame 222 at 10% PLR



(a) Original

(b) H264 DP

Figure 6.16: Vassar subjective comparison of frame 175 at 10% PLR

Table 6.2: Subjective results for different PLR of GOP size 16

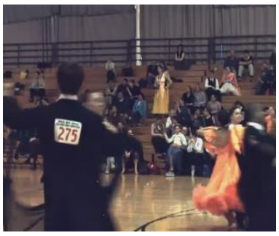
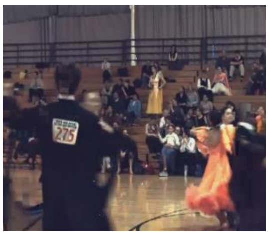
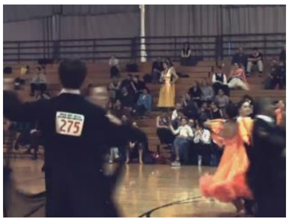
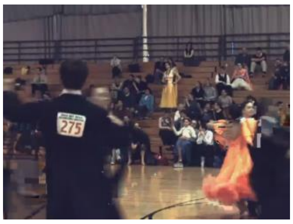
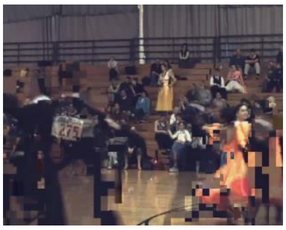
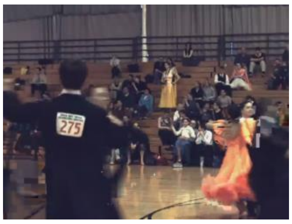
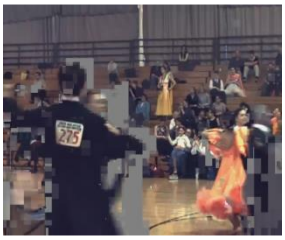
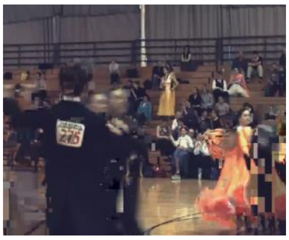
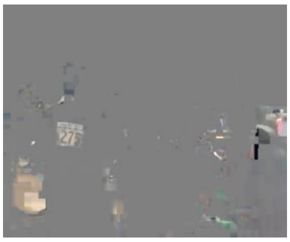
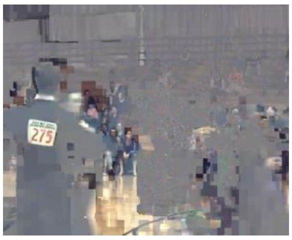
Frame 47 of Ballroom Test Sequence			
PLR	Original Frame	H264 DP	ML DP
1%			
5%			
10%			
15%			
20%			

Table 6.3: Subjective results for different PLR of GOP size 16

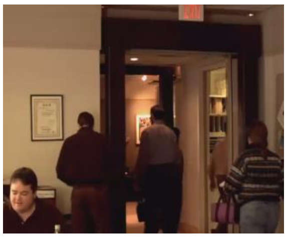
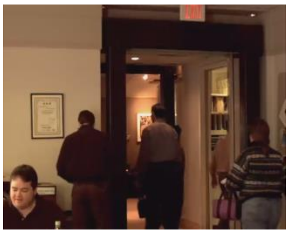
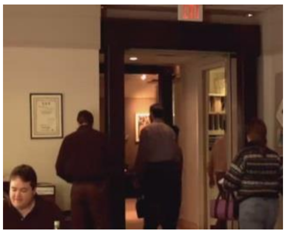
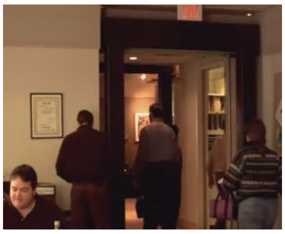
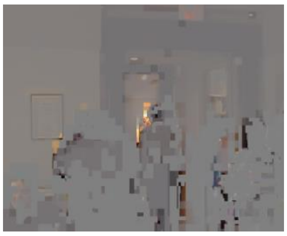
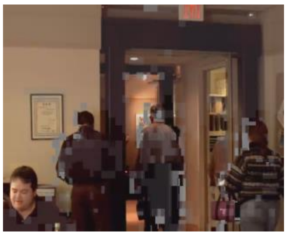
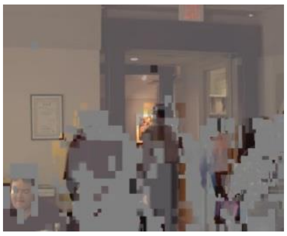
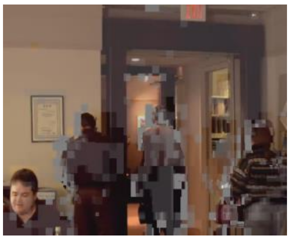
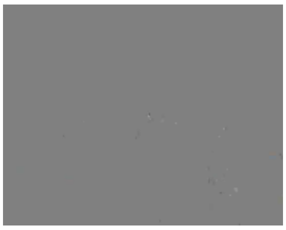
Frame 222 of Vassar Test Sequence			
PLR	Original Frame	H264 DP	ML DP
1%			
5%			
10%			
15%			
20%			

Table 6.4: Subjective results for different PLR of GOP size 16

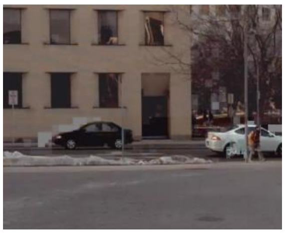
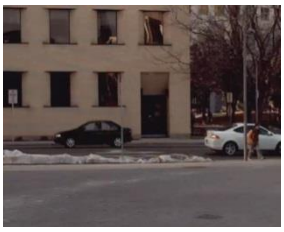
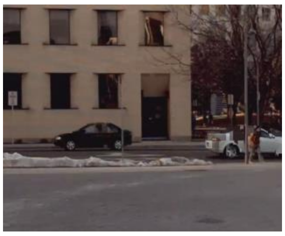
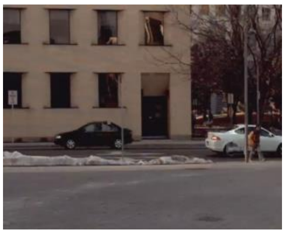
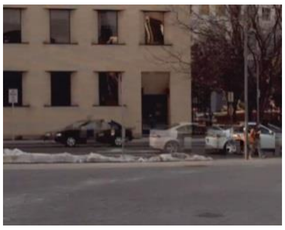
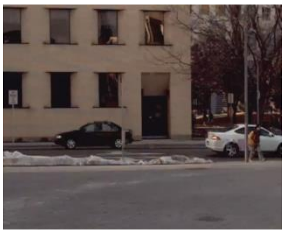
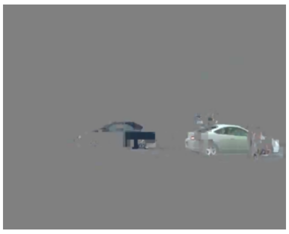
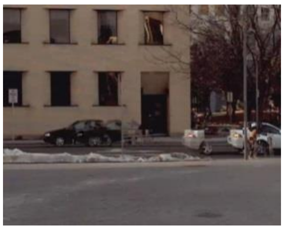
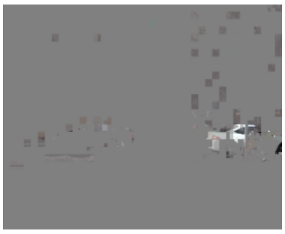
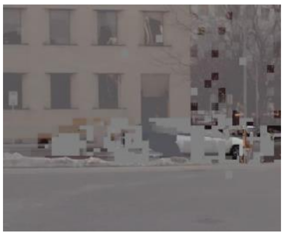
Frame 175 of Ballroom Test Sequence			
PLR	Original Frame	H264 DP	ML DP
1%			
5%			
10%			
15%			
20%			

Table 6.2 consists of the subjective results for all the different Packet loss rates used in the ballroom sequence simulation. The table shows the base view subjective comparison between the H.264 DP and multi-layer DP technique with reference to the original frame in terms of visual quality of frame number 47. It can be observed that both techniques can reconstruct the frame with an acceptable quality of 1% and 5% error rate. As seen and observed previously, the multi-layer DP technique could reconstruct the frame with better quality than the H.264 DP technique. However, at higher loss rate of 15% and 20% even though the multi-layer technique could reconstruct most of the information that is lost in H.264 DP technique for 15%, it is still not good enough for quality viewing. The two techniques are worst for 20% loss rate, as both techniques have poorly reconstructed the frame with accepted quality. As explained earlier, this is as a result of high losses of information in the multiview coded video that are permanently lost and beyond recovery by the error concealment technique.

Table 6.3 consists of the subjective results for all the different Packet loss rates used in the exit test sequence simulation. The table shows the base view subjective comparison between the H.264 DP and multi-layer DP technique with reference to the original frame in terms of visual quality of frame number 222. Similar to ballroom scenario, in the subjective result for 20% error rate, the two techniques could barely reconstruct any video information in the frame.

Table 6.4 consists of the subjective results for all the different Packet loss rates used in the Vassar test sequence simulation. The table shows the base view subjective comparison between the H.264 DP and multi-layer DP technique with reference to the original frame in terms of visual quality of frame number 175. Interestingly, in this particular simulation and scenario, it can be observed that there is a significant loss in the visual quality of 1% and 5% in the H.264 DP technique. These losses are all recovered with better quality with the multi-layer technique, and of course, the visual quality of 15% and 20% error rates has failed for H.264 and is not encouraging for the multi-layer DP technique.

6.4. Analysis of GOP Size and the Effects on MVC over Error-Prone Channels

6.4.1. Experimental Results

In our experiments, different experiments are carried out with the modified decoder for different GOP sizes. The decoding capability and performance of the decoder in terms of concealing losses is examined. This section describes the performance evaluation and results of the effects of GOP size on multiview video bitstream over the wireless network. The values of GOP sizes used in the experiments are 4, 8, 12, and 16 respectively. Also, the error rates used are 0%, 1%, 5%, 10%, 15%, and 20% respectively. For every GOP size and error rate considered, a minimum of ten different simulations are conducted, and the average results are generated. The perceptual quality of each reconstructed view is measured in terms of peak signal to noise ratio (PSNR) for all the different simulations and error rates used in the experiment. The experimental values for ballroom, exit and Vassar test sequences are recorded in table 6.5, table 6.6 and table 6.7 respectively for different loss rates and GOP sizes. The bitrate performance for different GOP sizes is recorded in table 6.8

6.4.2. Objective and Subjective analysis

Table 6.5 Numerical simulation results for Ballroom

Ballroom GOP4			Ballroom GOP8	
PLR (%)	H264 DP (dB)	H264 ML (dB)	H264 DP (dB)	H264 ML(dB)
0	35.45	35.45	35.16	35.16
1	34.53	34.93	34.67	34.72
5	28.54	28.90	30.28	27.97
10	24.73	24.37	26.82	24.96
15	21.04	22.93	21.35	21.90
20	18.65	20.04	18.09	19.04
Ballroom GOP 12			Ballroom GOP 16	
PLR (%)	H264 DP ((dB)	H264 ML (dB)	H264 DP ((dB)	H264 ML(dB)
0	34.99	34.99	34.83	34.83
1	34.74	32.83	34.38	33.41
5	30.42	30.10	30.42	31.82
10	24.24	24.22	24.61	25.59
15	20.94	21.63	19.23	22.52
20	18.23	20.09	16.01	19.17

Table 6.6 Numerical simulation results for Exit

Exit GOP 4			Exit GOP 8	
PLR (%)	H264 DP ((dB)	H264 ML (dB)	H264 DP (dB)	H264 ML (dB)
0	37.58	37.58	37.36	37.36
1	37.08	35.50	36.82	36.82
5	31.15	33.94	35.52	35.20
10	29.78	27.28	32.43	30.61
15	27.92	27.09	20.32	27.23
20	22.35	21.53	23.02	23.53
Exit GOP 12			Exit GOP 16	
PLR (%)	H264 DP (dB)	H264 ML (dB)	H264 DP (dB)	H264 ML (dB)
0	37.25	37.26	37.11	37.11
1	36.67	34.96	35.87	36.84
5	30.07	33.77	30.75	30.52
10	21.73	29.02	26.37	25.09
15	25.76	22.91	22.17	23.25
20	23.82	24.61	21.11	21.95

Table 6.7 Numerical simulation results for Vassar

Vassar GOP 4			Vassar GOP 8	
PLR (%)	H264 DP (dB)	H264 ML(dB)	H264 DP (dB)	H264 ML(dB)
0	35.37	35.58	35.30	35.30
1	34.13	35.00	35.08	35.11
5	32.34	31.23	32.82	32.75
10	29.60	27.45	28.22	30.44
15	27.47	24.55	23.08	23.85
20	22.33	20.85	22.44	22.19
Vassar GOP 12			Vassar GOP 16	
PLR (%)	H264 DP (dB)	H264 ML(dB)	H264 DP (dB)	H264 ML(dB)
0	35.32	35.32	35.27	35.27
1	35.10	33.53	33.12	34.67
5	29.72	31.18	31.99	32.44
10	25.94	28.05	26.97	27.91
15	22.60	24.66	20.13	23.00
20	20.83	22.21	18.42	20.11

Table 6.8 Bitrate simulation results for different test sequences

Ballroom		Exit		Vassar	
GOP	Bitrate (Kb/s)	GOP	Bitrate (Kb/s)	GOP	Bitrate (Kb/p)
4	1909.69	4	834.36	4	759.05
8	1619.76	8	722.12	8	657.69
12	1527.94	12	700.24	12	691.68
16	1374.75	16	535.23	16	572.54

Fig. 6.17 shows the ballroom performance evaluation for the H.264 DP and the multi-layer DP method for different error rates and GOP sizes. For 10 different runs of the simulation conducted, multi-layer DP has a better and improved quality performance than the H.264 DP technique in many instances of error rates. Note that, video coding works either as fixed quality and variable bitrate and vice-versa. So in this experiment, various quality levels are examined for constant bitrate as recorded in table 6.5. It can be observed objectively from Fig. 6.18 and Fig. 6.19 that the multi-layer technique has also improved the quality performance compared to the H.264 DP for exit and vassar test sequences respectively. The bitrate performance evaluation of the two techniques is reported in Fig. 6.20, Fig. 6.21, and Fig. 6.22 for ballroom, exit and vassar test sequences respectively. The results demonstrate a very low bit rate cost to implement the H.264 DP technique in the reference software. The figures further illustrates that the multi-layer data partitioning can be implemented with no additional bitrate.

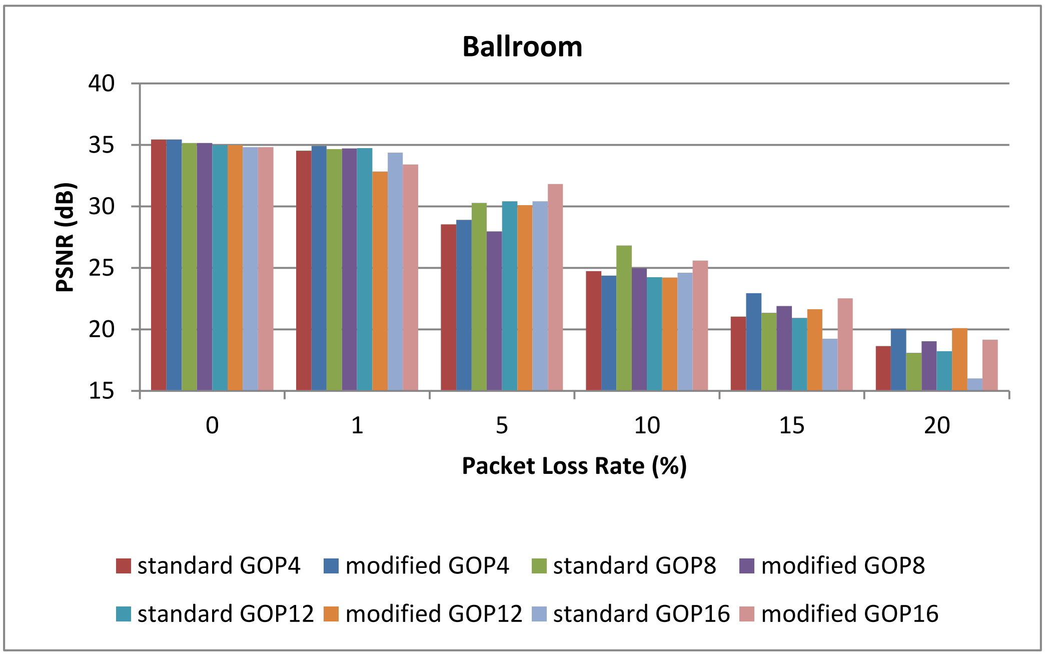


Figure 6.17: Ballroom quality evaluation with different GOP

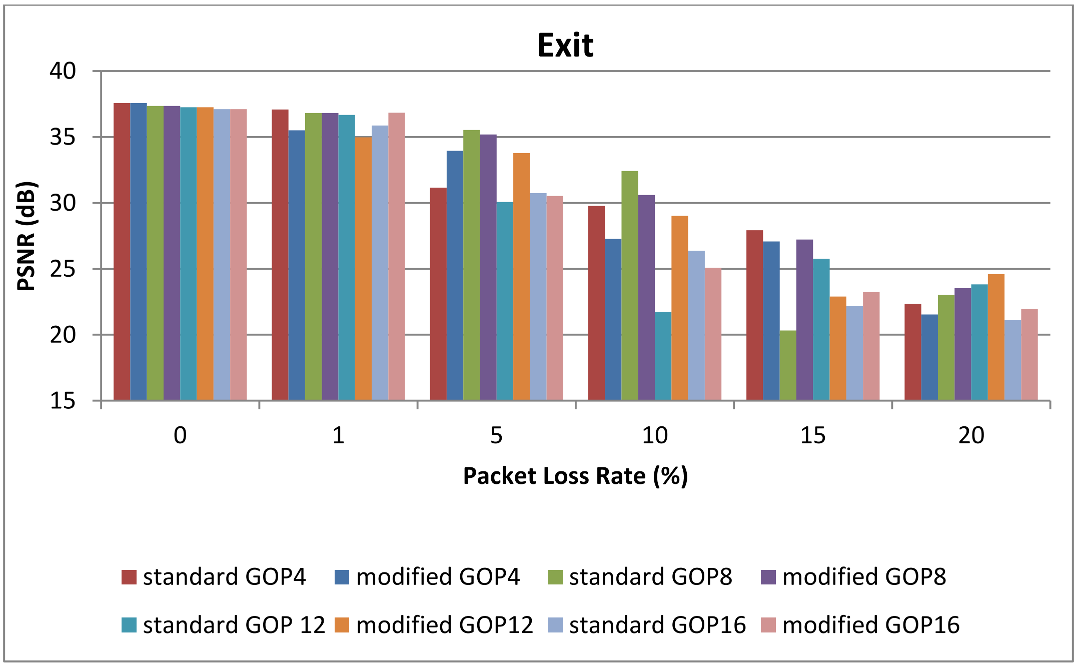


Figure 6.18: Exit quality evaluation with different GOP

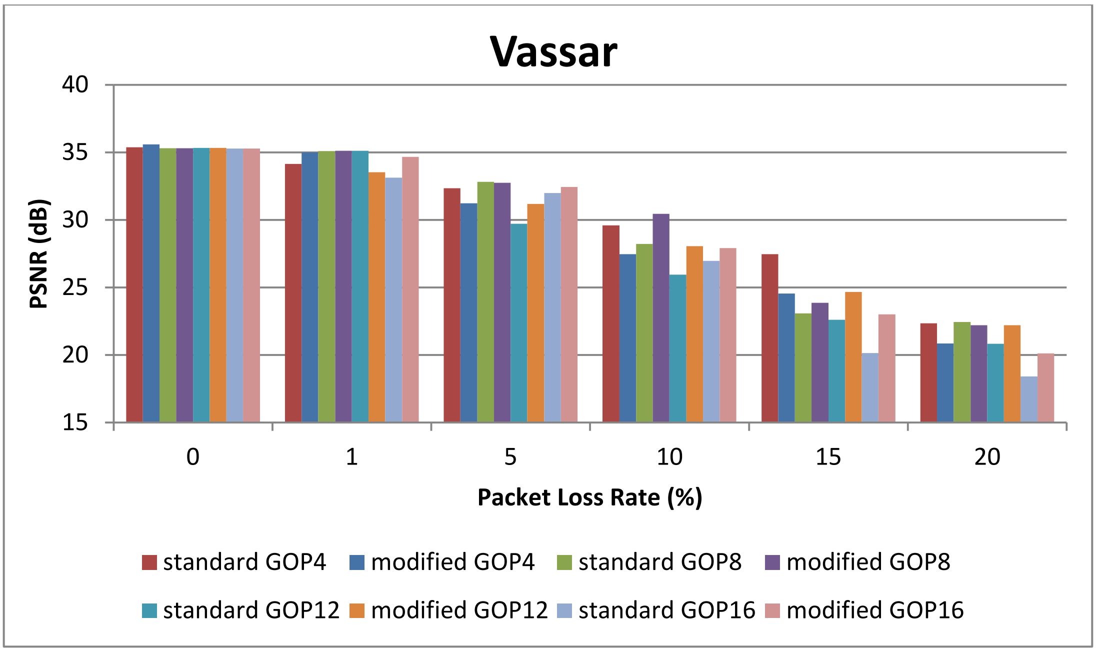
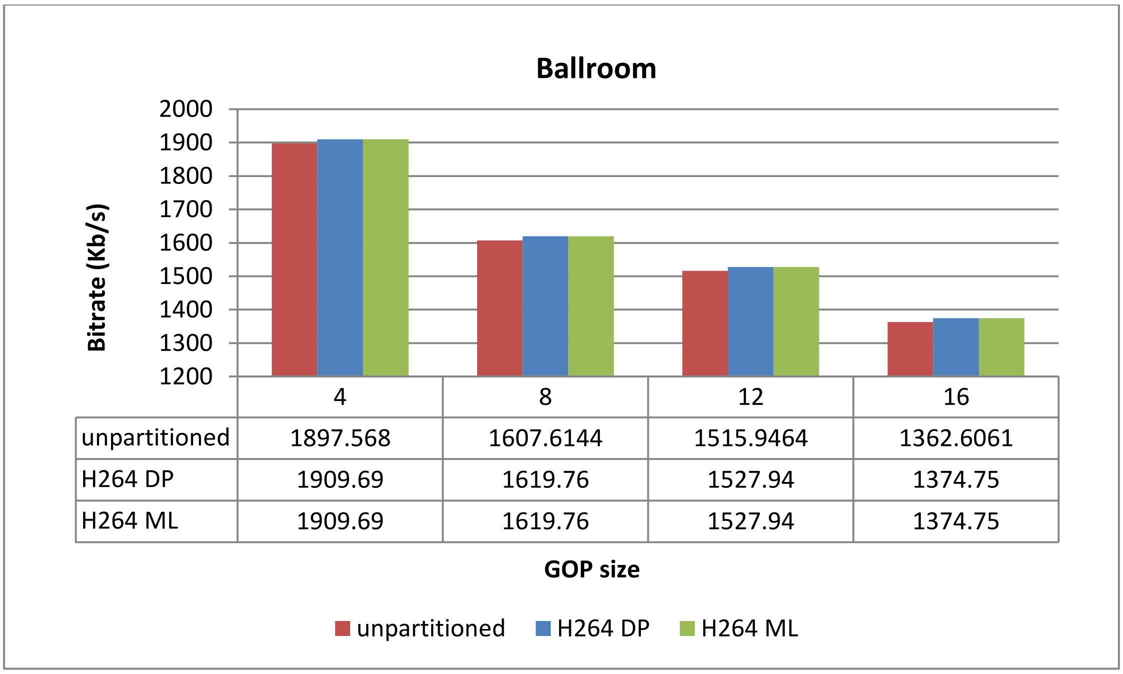
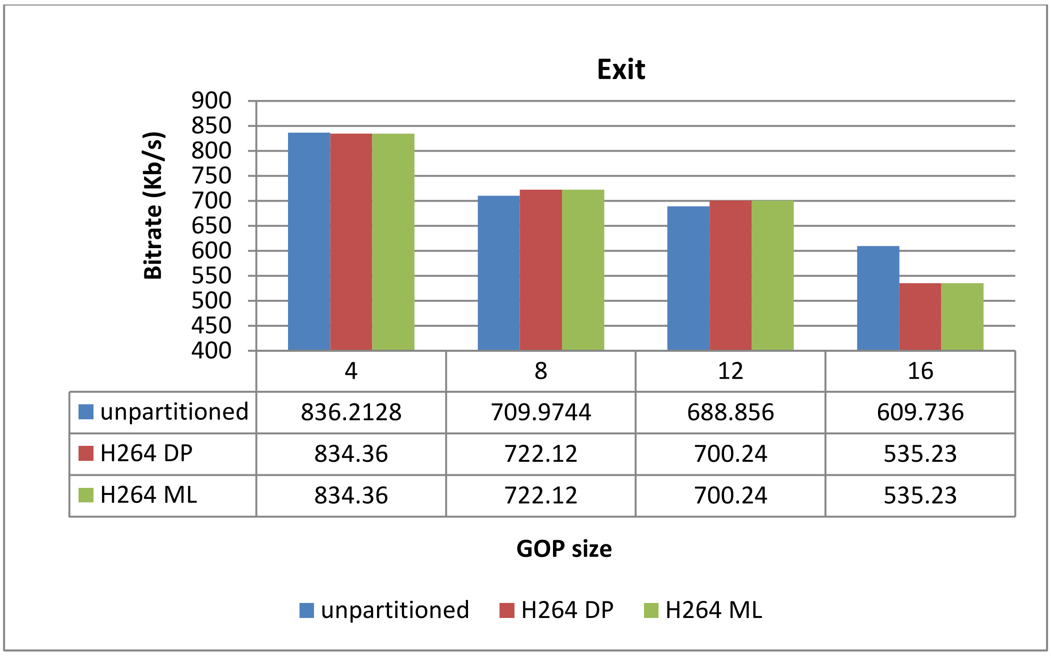


Figure 6.19: Vassar quality evaluation with different GOP



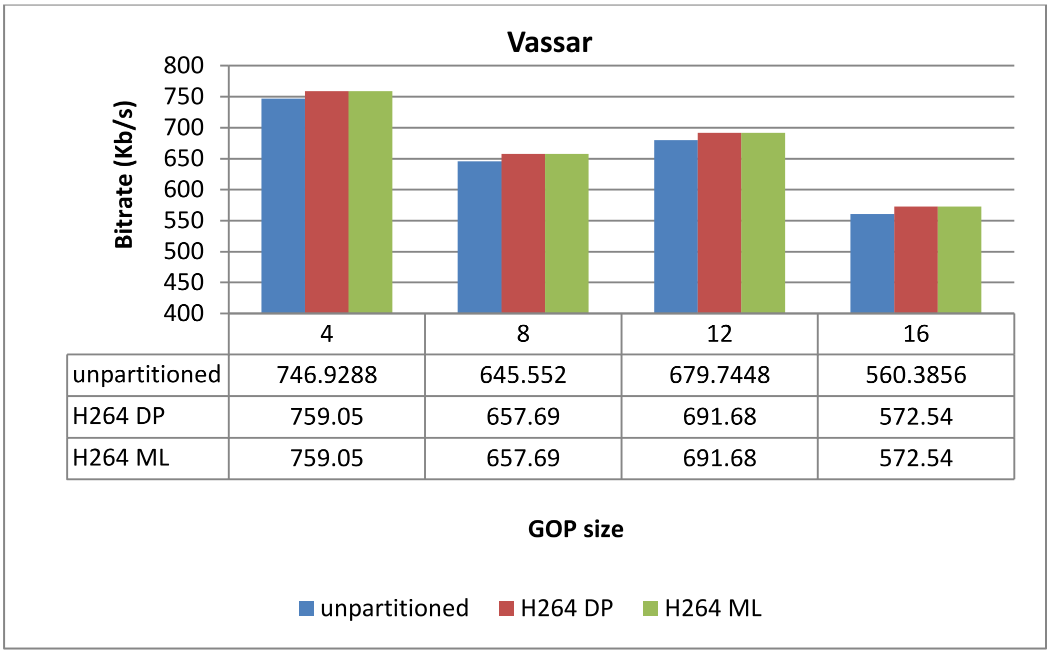
	4	8	12	16
unpartitioned	1897.568	1607.6144	1515.9464	1362.6061
H264 DP	1909.69	1619.76	1527.94	1374.75
H264 ML	1909.69	1619.76	1527.94	1374.75

Figure 6.20: Bitrate performance for different GOP sizes for Ballroom



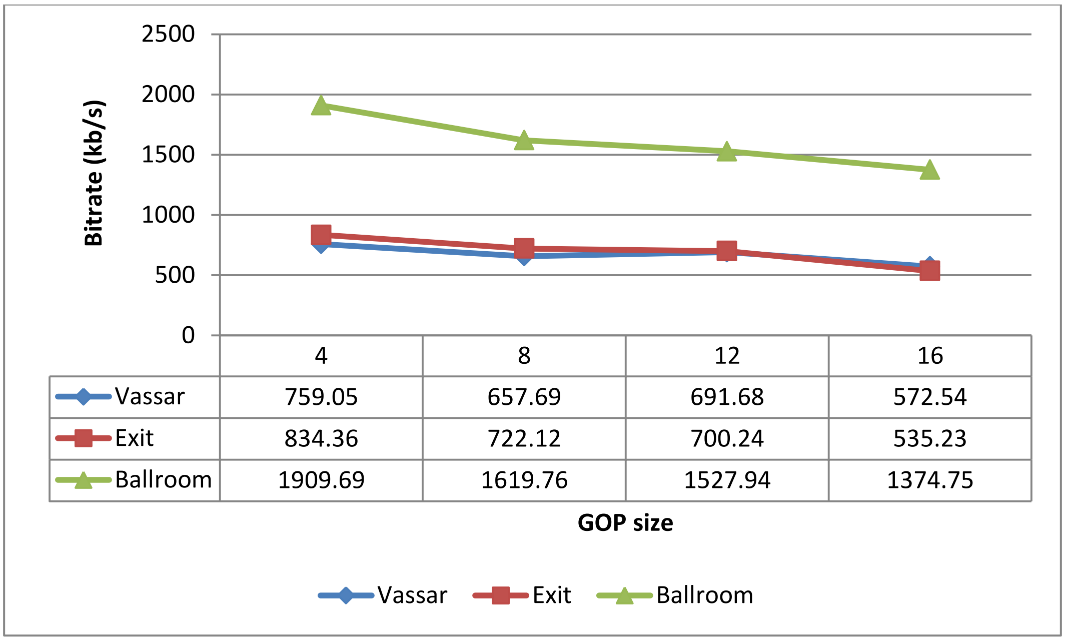
	4	8	12	16
🟦 unpartitioned	836.2128	709.9744	688.856	609.736
🟥 H264 DP	834.36	722.12	700.24	535.23
🟩 H264 ML	834.36	722.12	700.24	535.23

Figure 6.21: Bitrate performance for different GOP sizes for Exit



	4	8	12	16
unpartitioned	746.9288	645.552	679.7448	560.3856
H264 DP	759.05	657.69	691.68	572.54
H264 ML	759.05	657.69	691.68	572.54

Figure 6.22: Bitrate performance for different GOP sizes for Vassar



	4	8	12	16
Vassar	759.05	657.69	691.68	572.54
Exit	834.36	722.12	700.24	535.23
Ballroom	1909.69	1619.76	1527.94	1374.75

Figure 6.23: Bitrate performance for different test sequences

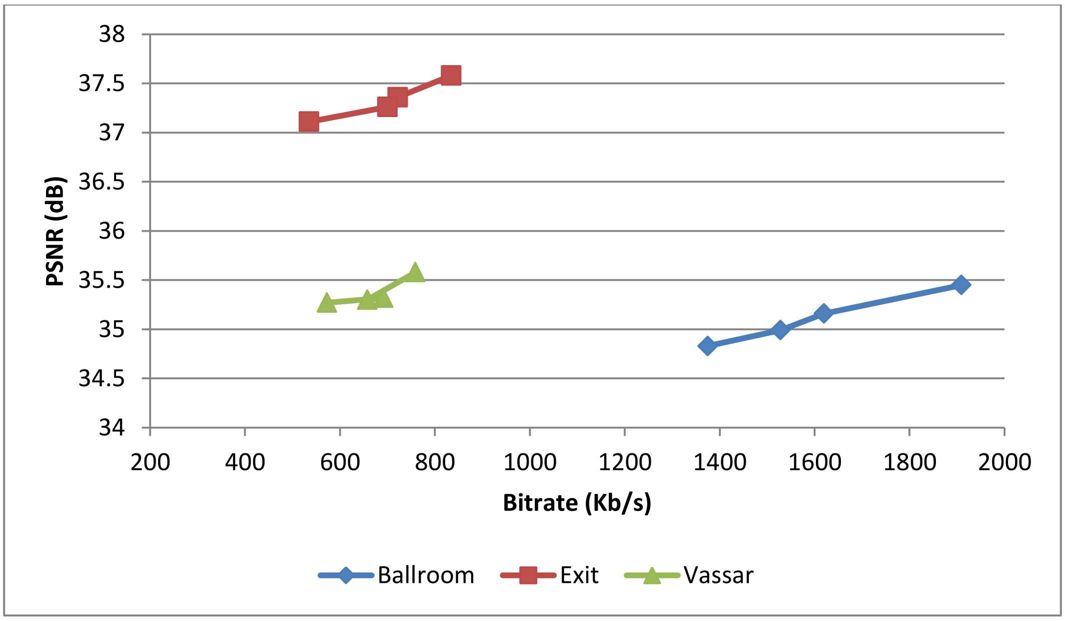


Figure 6.24: Relationship between quality and bitrate for different test sequences

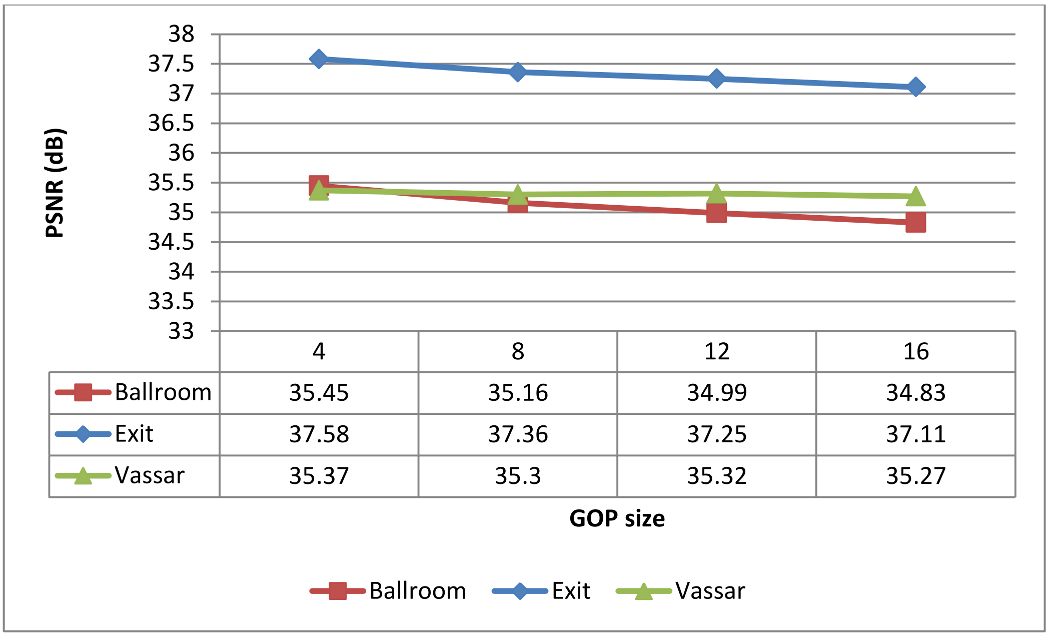
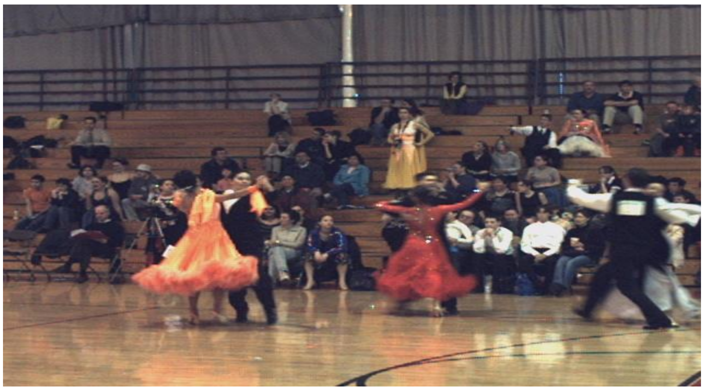
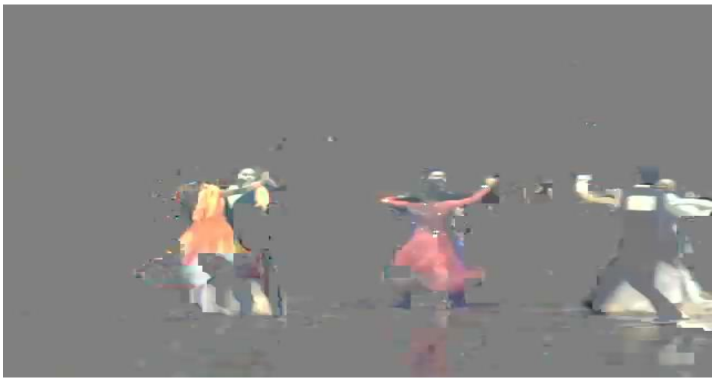


Figure 6.25: Quality evaluation for different test sequences with different GOP sizes

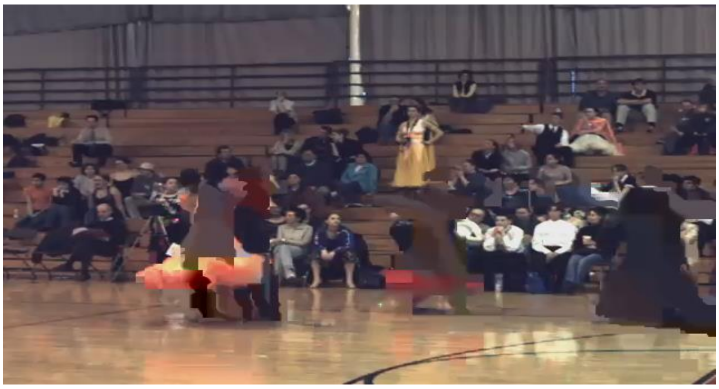
The subjective results are presented next for the ballroom, exit and vassar test sequences. In all our subjective analysis, we have observed from the results that multi-layer data partitioning technique can give an improved perceptual quality performance than the H.264 DP technique. Fig. 6.26, Fig. 6.27 and Fig. 6.28 represent the subjective results for view 0, view 1, and view 2 of ballroom test sequence. A subjective comparison is made between the original frame without data partitioning and free from error and a frame from the H.264 DP and multi-layer DP technique. Frame 121 is chosen from each view at 20% loss rate and a GOP of 16. The greyscale effect in Multi-layer DP technique is completely removed. We can observe closely in the Multi-layer DP that these frames are not reconstructed with the best quality when compared with the original frames. This is because of the high error rate used in the network simulations which result to several of the video information to be lost and also the limitation of the frame copy concealment to recover high losses. At such error rate of 20% and GOP of 16, the multi-layer DP technique could recover most of the lost video information with improved quality compared to H264 DP technique at the same error rate and GOP size.



Original

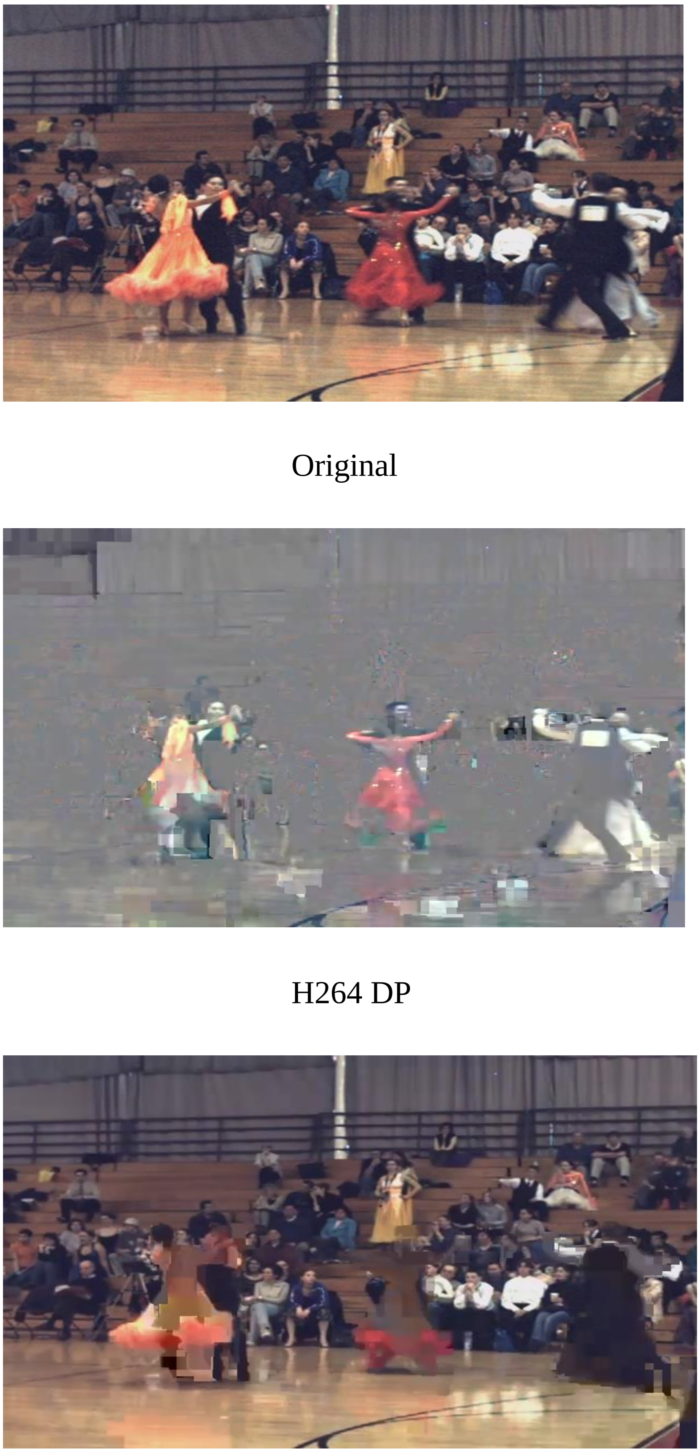


H264 DP



ML DP

Figure 6.26: Ballroom subjective comparison for frame 121 of view 0 at 20% PLR

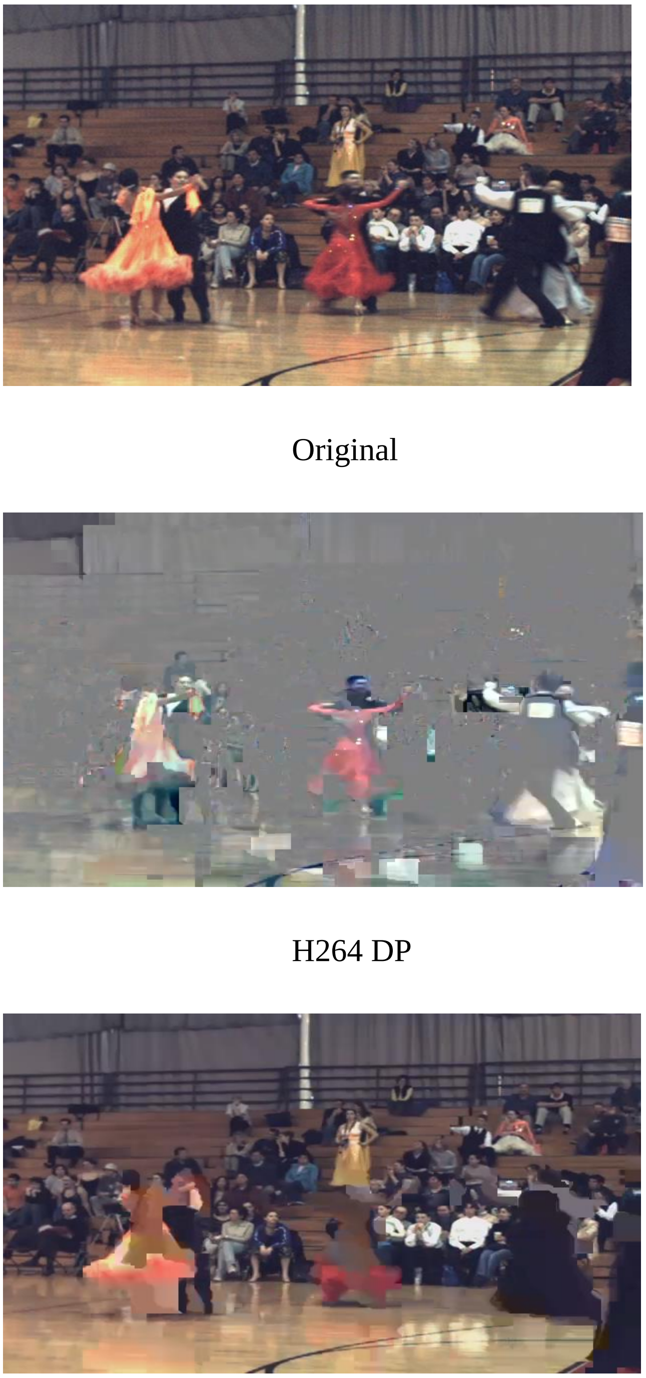


Original

H264 DP

ML DP

Figure 6.27: Ballroom subjective comparison for frame 121 of view 1 at 20% PLR



Original

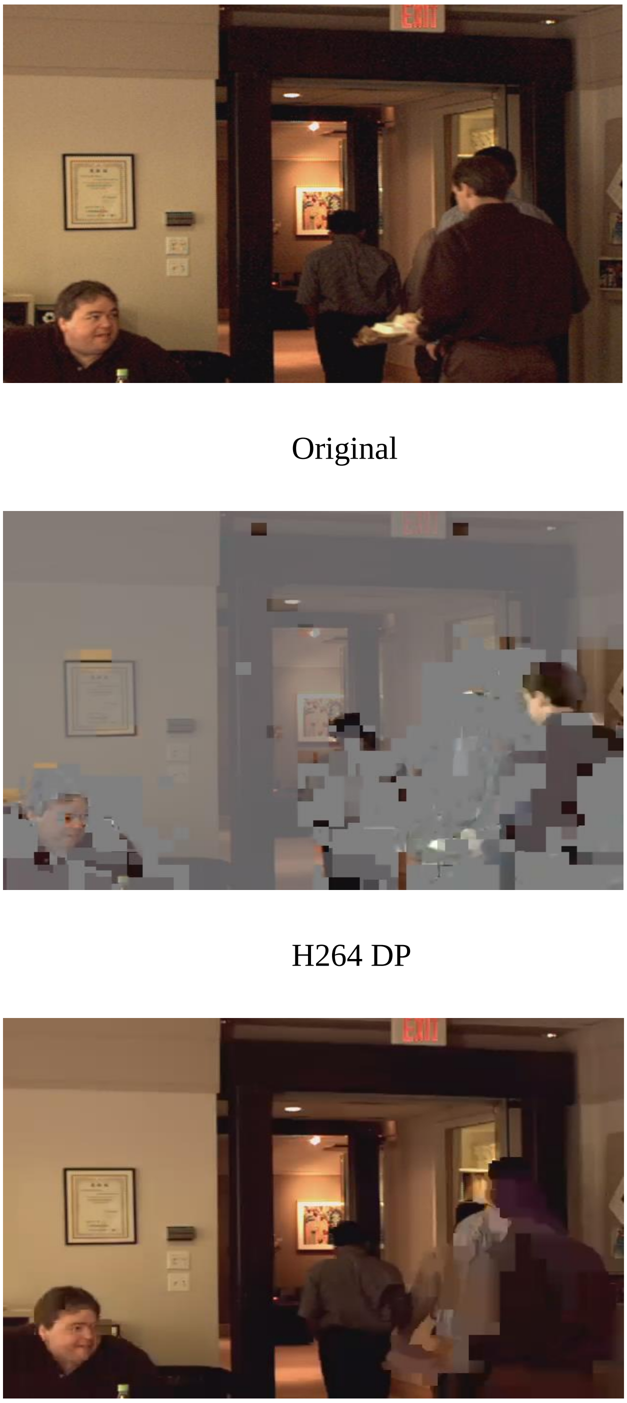
H264 DP

ML DP

Figure 6.28: Ballroom subjective comparison for frame 121 of view 2 at 20% PLR

Similarly, in the subjective test of exit and vassar test sequences, we have observed that the multi-layer data partitioning technique can improve the visual quality of the reconstructed video in a better way than the H.264 DP. Frame number 121 of the exit test sequence is selected for comparison and analysis at 20% error rate and GOP of 16. The subjective result for exit sequence of the three views is illustrated in Fig. 6.29, Fig. 6.30, and Fig. 6.31 respectively. Likewise, Frame number 250 of the exit test sequence is selected for comparison and analysis at 20% error rate and GOP of 16 in this experiment. The subjective result for the vassar sequence of the three views is illustrated in Fig. 6.32, Fig. 6.33, and Fig. 6.34 respectively. The reconstructed frames from the multi-layer technique are not well reconstructed with good quality as the original as can be seen. But can recover much of the video data that is lost in the H.264 DP method. The correct decoding of the multiview bitstream depends on how the reference frames are received. As can be observed, the loss of a slice can degrade the video quality and propagate to various other frames within the GOP. By increasing the packet loss rate proportionally increases the visual degradation to a much poorer quality. It is important to analyse the effects of error propagation within a GOP of the multi-layer data partitioned bitstream. In hierarchical GOP like the one in multiview video coding, the reference decoder uses the I-frame in the base view and the anchor frames in the non-base view either directly or indirectly as reference frames for all other frames within the GOP. Referring back to Fig. 4.13, if an error occurs in the I-frame of view 0, it can result to artefacts that can continue to propagate throughout the GOP structure. The effect can be experienced in both temporal and interview manner until the next random access point. At this point, the decoder refreshes with the next intra coded frame in view 0 or the anchor frames in either view 1 and 2. It has been noticed that losses within the I-frame that does not affect the header information such as intra coded MBs coefficient can also propagate errors throughout the GOP. P-frames are coded using motion compensation prediction from previous reference frames. Anchor frame such as the one in view 2 is forward predicted from the I-frame in view 0, subsequent prediction of other non-anchor frames in both view 2 and view 1 takes reference from their preceding P-frame. Any form of loss in this frame can further propagate error through the remainder of the GOP until the next refresh frame is received within the multi-layer partitioned bitstream.

It can be highlighted that the impact of P-frame or anchor frame of view 2 can be almost as significant as losing an I-frame because of interdependencies from other frames. Due to the hierarchical nature of MVC bitstream, anchor frame in view 1 that is interview predicted from view 0 and view 2 is used to predict other non-anchor frames temporally within the GOP. So the effect of error is limited to view 1 only and less severe than I and P-frames in the multiview video bitstream.

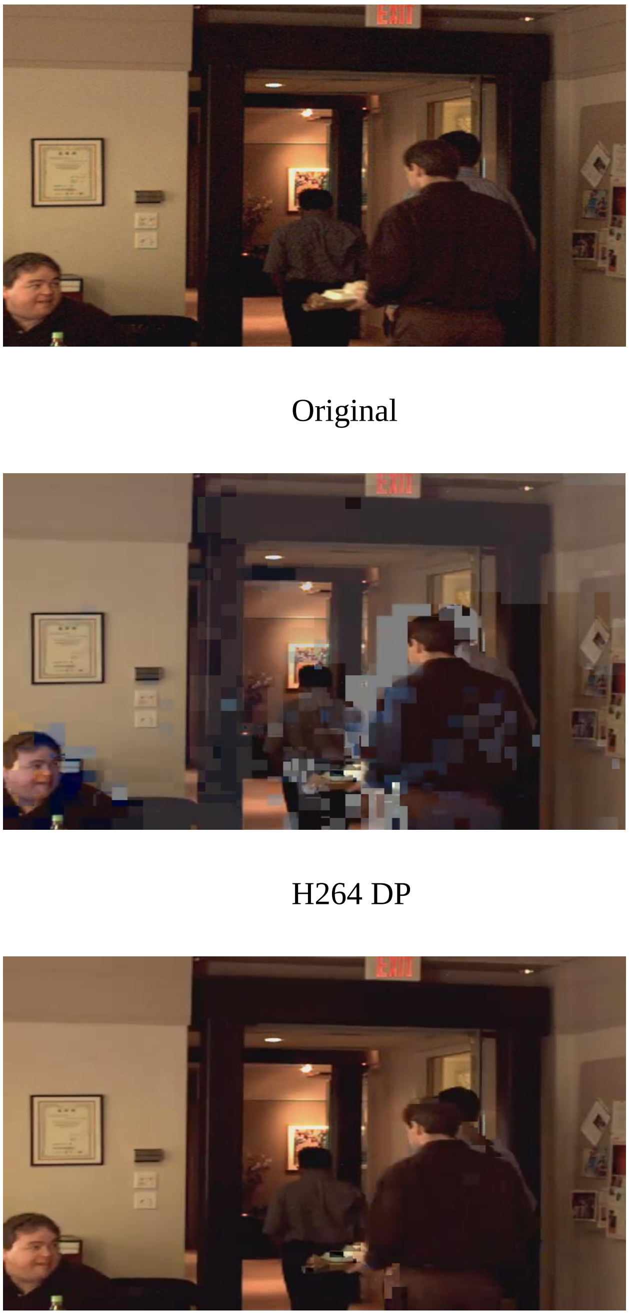


Original

H264 DP

ML DP

Figure 6.29: Exit subjective comparison for frame 121 of view 0 at 20% PLR

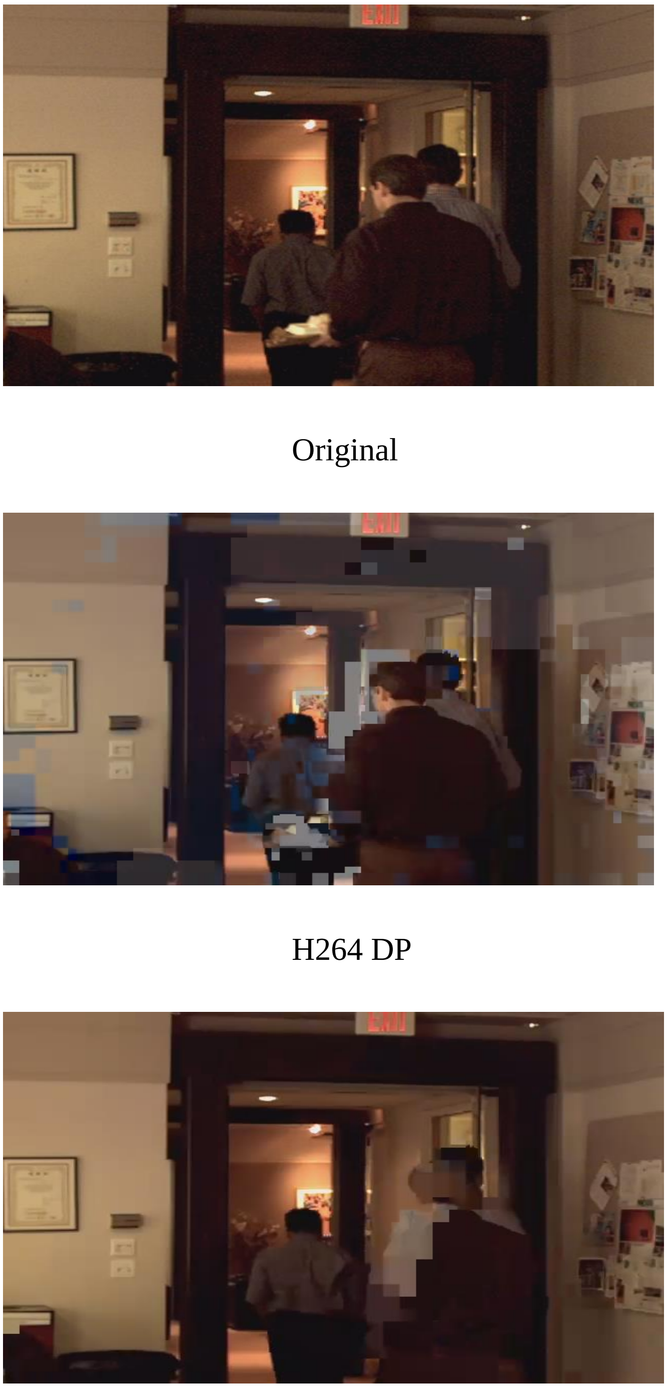


Original

H264 DP

ML DP

Figure 6.30: Exit subjective comparison for frame 121 of view 1 at 20% PLR

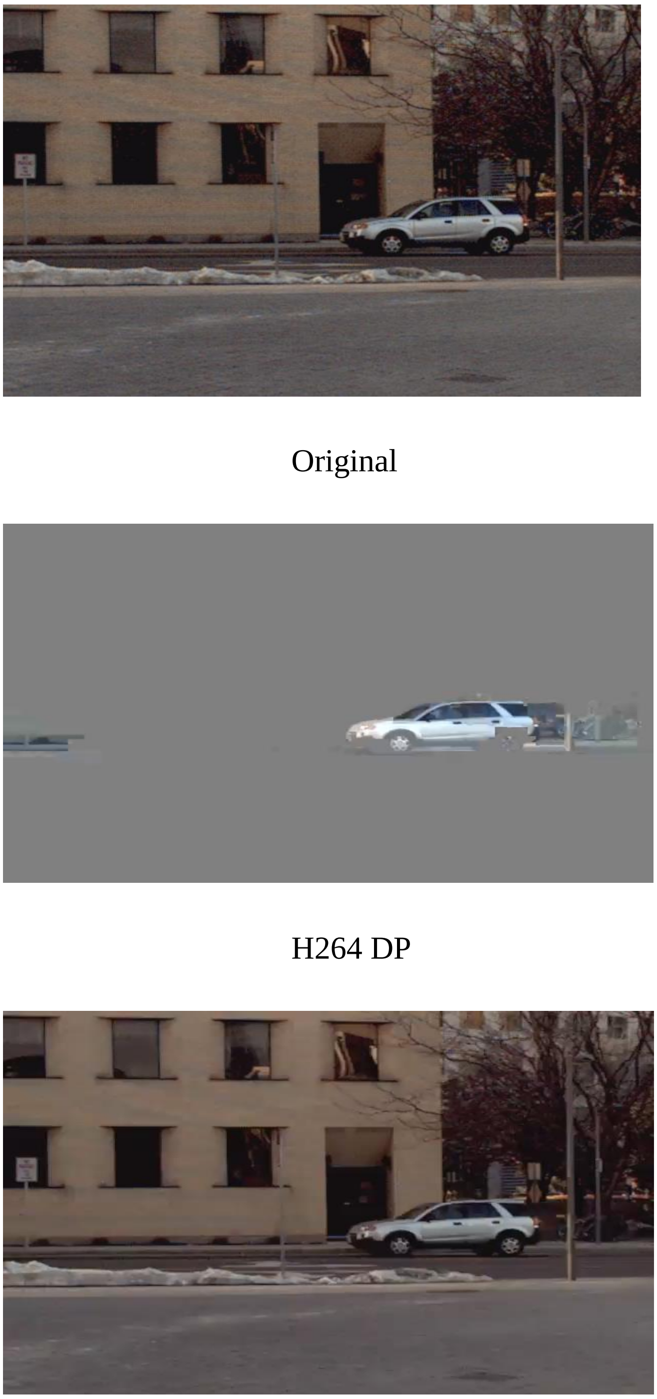


Original

H264 DP

ML DP

Figure 6.31: Exit subjective comparison for frame 121 of view 2 at 20% PLR

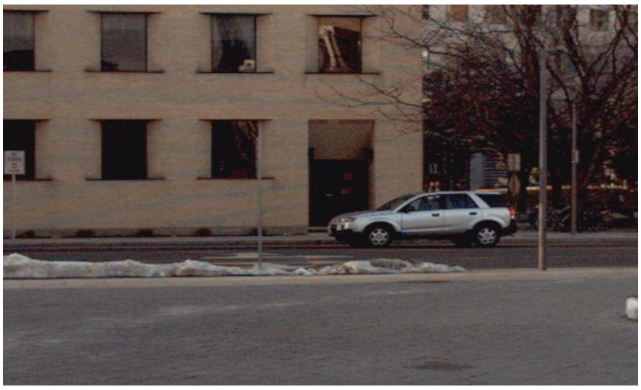


Original

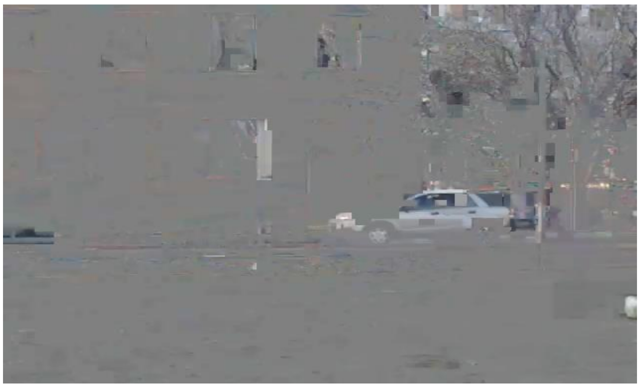
H264 DP

ML DP

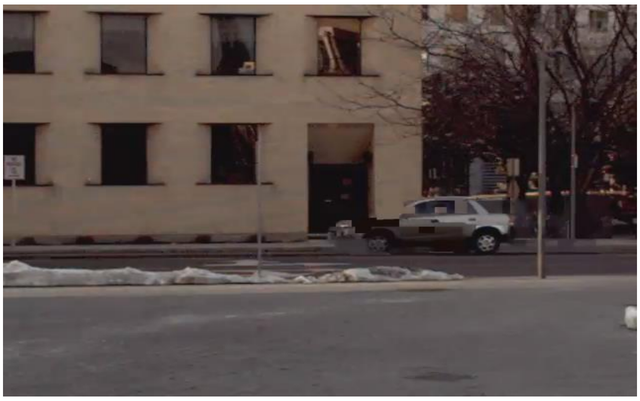
Figure 6.32: Vassar subjective comparison for frame 250 of view 0 at 20% PLR



Original

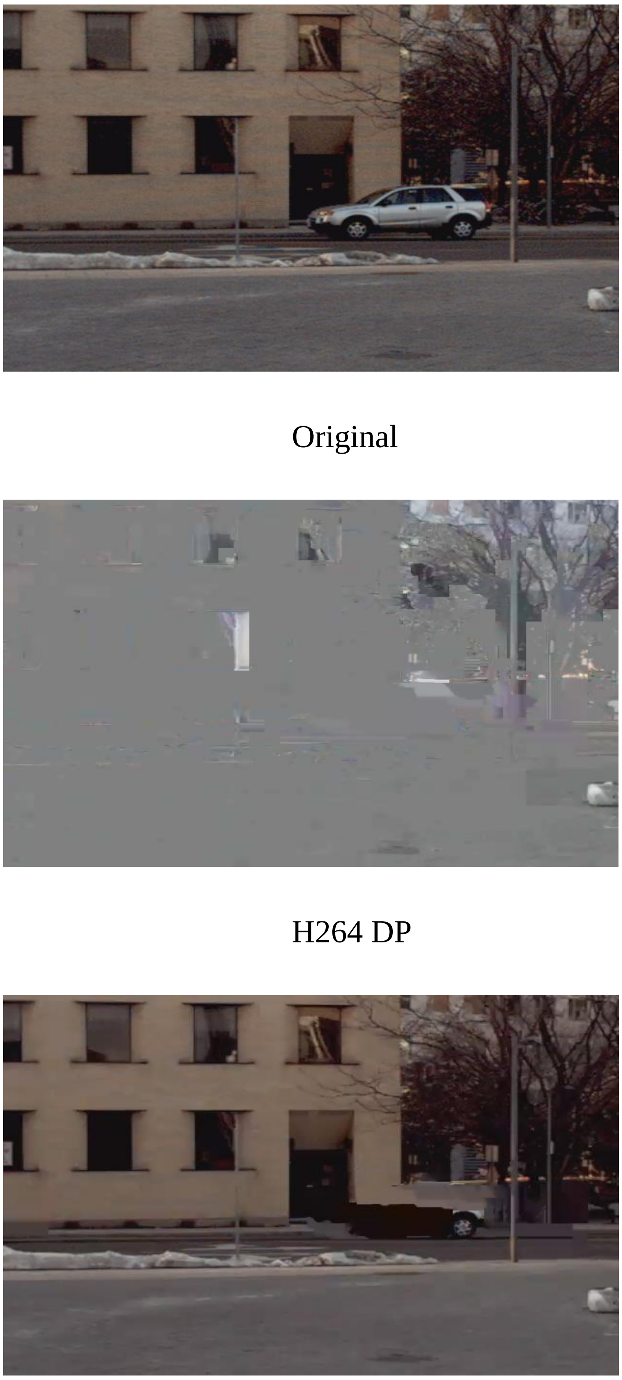


H264 DP



ML DP

Figure 6.33: Vassar subjective comparison for frame 250 of view 1 at 20% PLR



Original

H264 DP

ML DP

Figure 6.34: Vassar subjective comparison for frame 250 of view 2 at 20% PLR

6.5. Analysis and Discussion

At lower error rate of 1% in the ballroom sequence, it is observed that from Fig. 6.17 the H.264 DP has by a small margin outperformed the multi-layer DP for GOP = 8 and error rates of 5% and 10%. Also, from the objective performance in Fig. 6.18, H.264 DP has slightly demonstrated better quality performance than the multi-layer DP for GOP = 8. This is an indication that the H.264 DP in MVC can be effective for very low error rate channels. However, the multi-layer DP technique on average has a better quality performance than the H.264 DP technique in high error rate channel. From Fig. 6.23 and Fig. 6.24, the results of the experiment have revealed that a small number of GOP size means more I-frames. This can have a tendency to consume more bits because of the frequent occurrence of intra frames within the GOP. However, having more I-frames increases the multiview bitstream size. It can also have the tendency of reducing the efficiency of the multiview video coding. Different applications can have different GOP requirements such as real time and offline applications each having a different latency or delay requirement [234]. From the objective result in Fig. 6.25, the results obtained illustrate that lower GOP size can give a slightly better perceptual quality for the multi-layer DP technique. This is because low GOP means more number of intra frames within the GOP with less prediction error which can result in a higher video quality. In video communications over an error-prone environment, trade-off between perceptual quality and bitrate consumption is important and necessary [235]. In most cases, applications requiring a high level of quality in an error-prone network end will have a higher bitrate in order to make the MVV bitstream more resilient to channel noise and that result in visual quality improvement. In our experiment, we can record from the list of simulations a constant bit rate for variable video quality for different GOP size. For different quality and loss rate, bitrate remained constant as observed and recorded in our simulation. Encoding with a constant bit rate means that the reference encoder has an idea of the bit rate range before the encoding of the videos even begin. The same number of bits is used by the encoder to encode the entire videos in order to achieve the target bit rate while varying quality level. Even though, the bitrate is not exactly constant at some points and slightly varies but remains close to the average value.

However, this explains why the quality of the reconstructed video degrades are not uniform. This is recommended for bandwidth constraint application and can be predefined by the user by setting a target bitrate. The disadvantage of maintaining a constant bitrate to obtain variable quality is, for example, high motion in a video scene can lead to a bitrate requirement that is higher than the target bitrate. This constraint of maintaining a fixed bitrate may result in quality degradation in the reconstructed video.

6.6. Conclusion

This chapter presents the research contribution and proposes a new error resilient technique for multiview video coding. The H.264/AVC data partitioning technique is first implemented in the JMVC 8.5 reference software. This is followed by the implementation of the multi-layer data partitioning technique for improved robustness against transmission errors and losses. The two techniques are evaluated in terms of factors such as robustness, bitrate consideration and perceptual quality. The performance evaluation is carried out over a simulated network. The network condition of the simulation test bed is defined based on valid and practical networks for different error rates. Experimental results illustrate that the Multi-Layer DP technique can improve the visual perception of reconstructed videos for various loss rates and conditions. From the results obtained, it has been noticed that additional bits are not required to implement Multi-Layer DP. It is noticeable from the results that the proposed technique is capable of improving the visual quality of the MVV bitstream at higher loss rates than the H.264 DP technique at the same bit rate. It can be seen both subjectively and objectively how these losses can affect the reconstructed multiview video through spatial/temporal and inter-view error propagation. The effect of these errors is severe in the H.264 DP compared to the Multi-Layer DP, which has the capability of making the MVV bitstream more resilient to channel errors. Also in this chapter, the frame copy concealment algorithm is introduced, which is employed to support the decoding of erroneous MVV data partition bitstream for both the H.264 and multi-later technique. The frame concealment algorithm was necessary to implement because, without it, decoding of the MVV bitstream would have been impossible when errors are introduced. The algorithm is modified to work with the JMVC reference software and be able to handle the multi-layer DP bitstream. From the result of several experiments and simulations obtained, the modified concealment algorithm has enhanced both the performance of the multi-layer DP technique and the JMVC reference decoder considerable.

Furthermore, this chapter examines and presents an analysis of the effects of GOP size in multiview video coding over error prone channels. From the analysis, we can understand that the GOP within a video sequence is one of the key coding parameters that determine the video quality perception of the viewer, more importantly, the GOP size and the motion within the sequence. Large GOP size improves the compression efficiency, which can allow more or higher video content to be transmitted for a given bitrate. It is necessary to decide wisely what GOP structure and size to support any application such as streaming or transmitting videos. However, the work in this chapter focuses and illustrates the performance of the two algorithms for worst case scenario. The optimal GOP size that can be determined for high coding efficiency and low quality distortion varies for different test sequences. For example, in the ballroom test sequence which contains a very complex background, high motion and large objects are very difficult to encode with low bitrate and constant quality. We can observe from the objective and subjective results that the ballroom sequence is severely affected compared to the exit and the Vassar test sequence. These two sequences have got less complex activities contained within them, which is why they require a lower bitrate to the maintain a better video quality compared to the ballroom. This is demonstrated in our objective and subjective results. The two different techniques namely H264 DP and multi-layer DP are used to show this effect. Our experimental results illustrate that the Multi-Layer DP technique can improve the visual perception of reconstructed videos for higher error rates within allowable compression efficiency and bitrate. From the results obtained, we can assume and suggest that multi-layer DP technique can suitably be utilized for delivering multiview video content over bandwidth constraint and high error rate channel at a GOP size of 16. Please note that the work in this chapter is not claiming to achieve a remarkable visual quality. We are proposing based on the experimental study and simulated results, a different approach that can apparently improve the visual quality of multiview video in a very high error rate channel with a selected GOP size.

7. Chapter 7: Conclusions & Future work

7.1. Research Contributions

The main goal of the research work as described in the thesis was to develop an error resilient technique for 3D multiview video coding, which can improve the error robustness of the coded sequences against transmission error. There are several standard error resilient mechanisms for monoscopic video defined by the video coding standard organizations in the literature. Some of these techniques can be extended to MVC as proposed by many researchers. In this research work, data partitioning as an error resilient technique in video communication is employed. The idea is to minimize the effects of transmission losses in the MVV bitstream and to improve the perceptual quality while considering the cost of bitrate and coding efficiency. The overall key achievements are as follows:

I. The quality performance of the MVC codec is with losses improved.
II. Design and development of MLDP technique in the MVC reference software.
III. Implementation of H.264/AVC DP technique in the MVC reference software.
IV. Decoder optimization for high error performance handling.

However, the thesis is structured into chapters that contributed in one way or the other to the research work and are summarized in this chapter as follows:

A general description and layout of the thesis is provided in chapter one, such as overview of 3D multiview video coding and its importance in the current state of multimedia communication. The chapter also highlights some problems and challenges that are related to MVC such as bandwidth variation and limitation and transmission losses.

A comprehensive review of the H.264/AVC video coding standard including concept, operation and all the key components that are constituted in the design are presented in chapter two. The concept of multiview video coding as an extension of the H.264/AVC is also elaborated such as MVC prediction structure, MVC bitstream and header extension structure and the decoding process. Furthermore, High Efficiency Video Coding (HEVC) standard is reviewed and recent work that has been conducted based on the performance and evaluations in terms of coding efficiency with respect to its predecessor.

The 3D fundamentals and the end-to-end communication pipeline is discussed in chapter three, which include 3D content creation, compression, format representation, and the display systems. Recent and state of the art developments especially in 3D coding are reported and discussed along with the problems and challenges with each component in the communication pipeline and the system technology at large. While 3D quality assessment is an active research field, the chapter also highlighted the challenges and some recent research work that has been done to improve the 3D perceptual quality assessment.

The standard error resilient techniques in H.264/AVC and the extension of some of these techniques in multiview video coding are described in chapter four. The chapter also gives a brief overview of error control techniques such as FEC in H.264/AVC video coding standard. The error concealment scheme in H.264/AVC standard and the MVC decoder operation in the presence of error and losses is also addressed

The experimental set-up and simulation test bed are reported in chapter five. The chapter describes the simulation approach, necessary conditions and coding parameters involved in the JMVC 8.5 reference software. The effects of key coding parameters such as Group of Pictures (GOP), Quantization Parameter (QP) on the multiview video coding are discussed. Multiview video sequence characteristics and the analysis of MVC bitstream are demonstrated including the layer that supports data partitioning technique and the implementation. The chapter also presents the network simulation test bed that is used to introduce or generate the error pattern in the MVV bitstream based on Gilbert modelling. Relevant software and applications that are used during the implementation, experiment such as CodecVisa bitstream analyser, codec modules and libraries are introduced and discussed. Furthermore, the conditions for error resilient in MVC is addressed, relationship between packet losses and bit errors in relation to the research are all discussed. Lastly, chapter five presents some results and analysis in an experiment that investigates and compares between simulcast video coding and 3D stereo video coding which exploits the interview dependency. The results of the experiment demonstrate that, while there is no considerable quality loss, we are able to record a bitrate saving of about 3% - 24%. This experiment is a confirmation of theoretical fact that considerable amount of bit rate reduction can be obtained when two or more views are encoded while exploiting the redundancies in other views. However, in simulcast coding, the same visual quality can be achieved but with a higher bit rate which is not suitable especially for bandwidth constraints video applications.

The main research contribution of improving the MVC codec performance in terms of error robustness is reported in chapter six. The new proposed multi-layer data partitioning and the H.264/AVC data partitioning technique are both evaluated in terms of network error robustness. The performance of the two techniques is examined when transmitted over a simulated network channel with different error rates. From the average objective analysis, the results obtained demonstrate that the multi-layer data partitioning produce a higher PSNR value in all the test sequences used for the different error rates. From the experimental result, an average of 2.1dB, 1.5dB and 1.3dB quality improvement in multi-layer data partitioning is observed over H.264 data partitioning in the vassar, exit and ballroom sequences respectively. In the subjective quality analysis, it has been found that the multi-layer technique can improve the perceptual quality in all the three multiview video sequences. The evaluation is based on 10% error rate which indicate that at this high rate, the multi-layer DP could reconstruct the views with improved visual quality than the H.264 DP. However, the multi-layer DP could not reconstruct the views to a higher quality as expected due to very high losses in the bitstream, but could still perform better than the H.264 DP technique. Most of the lost information in the H.264 DP technique is recovered by the multi-layer DP technique at such a high error rate of 20%. The multi-layer DP technique is developed and implemented in the JMVC 8.5 reference software with almost no added complexity. Furthermore, the chapter introduces simple error concealment and decoding scheme for MVC multi-layer DP bitstream with losses. Some results and analysis of the effects of GOP in MVC bitstream is presented. Different GOP sizes are experimented and simulated with different error rates and from the results obtained, GOP size of 16 is identified and recommended for optimal performance especially for video transmission or streaming over a high error-prone wireless networks. The cost of bit rate is crucial in error resilient schemes and video transmission over networks. The multi-layer data partitioning technique is implemented at no extra cost, and this has been demonstrated in the bitrate performance figures of this chapter. In general, data partitioning technique is efficient when considering bitrate consumption because the technique does not require many bits for the implementation. It is a common practice in error resilient and video coding system in general to examine two important factors that determine the performance and efficiency of the technique and system. These factors are either varying quality with constant bitrate or constant quality while varying bitrate. The chapter also confirms the concept by achieving constant bitrate for a particular GOP size and error rate with variable losses.

The effect of that is examined while experimenting with the multi-layer and H.264 DP technique for different GOP size and test sequences with different characteristics of motion and complexity. As expected, low motion and complex sequence such as exit sequence is encoded with less bitrate and improved quality when compared to higher motion and complex sequence such as ballroom sequence. Different level of quality both objectively and subjectively has shown that multi-layer DP can improve the quality perception compared to the H.264 DP technique with the same bitrate. The multi-layer DP in additional to the error concealment technique employed, however, could not efficiently reconstruct the views as would be expected with high and acceptable quality. This research attempts to examine the quality performance in multiview video coding of the two techniques over an excessive high error prone channel. In practice, a packet loss rate of 1% can be detrimental on the reconstructed views especially when a random error hits a packet with header information that relates to several other packets both in the same view and other view(s). Sometimes it can be very difficult to recover and reconstruct the views at such loss with a high level of perceptual quality in a real-time video communication application. As the error rate increases and in a consistent way, more annoying perceived quality is observed and the less the techniques become less robust to error and difficult to reconstruct with acceptable visual quality. While pushing the limits to what we call worst case scenario, the performance of multi-layer DP technique demonstrate an improved perceptual quality subjectively far better than the H.264 DP technique.

7.2. Future Work

Error control in 3D multiview video coding applications is an active research area that that will continue to advance especially in combating the inevitable transmission errors and losses in the network. In view of this research work, the concept of data partitioning seems to be the most promising error resilience mechanism. One major setback with the DP technique is the presence of high level of dependencies between the various partitions. We have seen that the loss of one partition can render other correctly received partitions useless. One way to minimize these dependencies in MVC is to export the constraint intra prediction feature of the H.264/AVC as defined in the specification into the JMVC reference software. This feature can eliminate the dependency of the partition containing intra-coded macroblocks on the partition containing inter coded macroblocks.

Furthermore, constrained inter prediction as proposed in the H.264/AVC extension can be applied in the JMVC codec in order to remove the dependencies that exist between data from inter-coded macroblocks in C partition and data from intra-coded macroblocks in B partition. This would mean that, when partition B is lost or corrupted in a packet loss network, partition C can still be used to recover the lost information in partition B. Note that, the research work in this thesis has only examined the effects of transmission losses in MVC bitstream, which opens the doors for further investigation and improvement. The work in this thesis is recommended also to consider the following key points for improvement and high-level performance.

- Data partitioning technique is ideal for supporting channel coding techniques such as Forward Error Correction (FEC). Hence, the multi-layer DP scheme with FEC scheme in the JMVC reference software can provide an improved level of quality performance to the video data or partitions against the transmission errors in the network channel. It is important to investigate and evaluate the cost of bitrate when implementing the channel coding in addition to the data partitioning technique. This is necessary in order not to compromise the overall coding efficiency with negligible quality improvement.
- In the decoder, the frame copy error concealment algorithm can be extended to hybrid error concealment algorithm which exploits the redundancies between adjacent macroblocks in both time and view domain in MVC. The technique, when implemented, can have a tendency to improve the visual quality perception of the reconstructed views.

We anticipate that from our current findings and results, a higher level of robustness and quality performance can be achieved when these recommended measures are considered and fully implemented while considering the cost of bit rate and coding efficiency.

References

[1] *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013–2018 - Cisco Systems, Inc., San Diego, White Paper (2014).*

[2] A. H. Sadka, "Flow Control in Compressed Video Communications," *Compressed Video Communications,* pp. 75-119, 2002.

[3] M. Lu, C. Lin, J. Yao and H. H. Chen, "Multiple description coding with spatial-temporal hybrid interpolation for video streaming in peer-to-peer networks," *Journal of Zhejiang University SCIENCE A,* vol. 7, pp. 894-899, 2006.

[4] Y. Wang, Y. Qu and R. Lv, "Packet dropping schemes and quality evaluation for h. 264 videos at high packet loss rates," in *Web Information Systems and Mining*Anonymous Springer, 2012, pp. 81-89.

[5] L. Liu, R. Cohen, H. Sun, A. Vetro and S. Zhuang, "New techniques for next generation video coding," in *Broadband Multimedia Systems and Broadcasting (BMSB), 2010 IEEE International Symposium on,* 2010, pp. 1-6.

[6] T. Stockhammer and M. M. Hannuksela, "H.264/AVC video for wireless transmission," *Wireless Communications, IEEE,* vol. 12, pp. 6-13, 2005.

[7] *Strongene-Standards and Profession Contribution.*

[8] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer and T. Wedi, "Video coding with H. 264/AVC: tools, performance, and complexity," *Circuits and Systems Magazine, IEEE,* vol. 4, pp. 7-28, 2004.

[9] T. Wiegand and G. J. Sullivan, "The H. 264/AVC video coding standard," *IEEE Signal Process. Mag.,* vol. 24, pp. 148-153, 2007.

[10] D. Marpe, T. Wiegand and G. J. Sullivan, "The H. 264/MPEG4 advanced video coding standard and its applications," *Communications Magazine, IEEE,* vol. 44, pp. 134-143, 2006.

[11] S. Winkler, *Digital Video Quality: Vision Models and Metrics.* John Wiley & Sons, 2005.

[12] T. Wiegand, G. J. Sullivan, G. Bjontegaard and A. Luthra, "Overview of the H. 264/AVC video coding standard," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 13, pp. 560-576, 2003.

[13] I. E. Richardson, *The H. 264 Advanced Video Compression Standard.* John Wiley & Sons, 2011.

[14] S. Branguolo, N. Tizon, B. Popescu and B. Lehembre, "Video transmission over UMTS networks using UDP/IP," in *Proc. of 14th European Signal Processing Conference (EUSIPCO), Florence, Italy,* 2006, .

[15] R. Schäfer, T. Wiegand and H. Schwarz, "The emerging H. 264/AVC standard," *EBU Technical Review,* vol. 293, 2003.

[16] T. Wedi and H. G. Musmann, "Motion-and aliasing-compensated prediction for hybrid video coding," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 13, pp. 577-586, 2003.

[17] A. Vetro, T. Wiegand and G. J. Sullivan, "Overview of the stereo and multiview video coding extensions of the H. 264/MPEG-4 AVC standard," *Proc IEEE,* vol. 99, pp. 626-642, 2011.

[18] M. Gerndt and D. Kranzlmüller, *High Performance Computing and Communications: Second International Conference, HPCC 2006, Munich, Germany, September 13-15, 2006, Proceedings.* Springer, 2006.

[19] G. J. Sullivan and T. Wiegand, "Video compression-from concepts to the H. 264/AVC standard," *Proc IEEE,* vol. 93, pp. 18-31, 2005.

[20] S. Kwon, A. Tamhankar and K. Rao, "Overview of H. 264/MPEG-4 part 10," *Journal of Visual Communication and Image Representation,* vol. 17, pp. 186-216, 2006.

[21] C. Wong, O. C. Au and R. Wong, "Advanced macro-block entropy coding in h. 264," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on,* 2007, pp. I-1169-I-1172.

[22] D. Marpe, H. Schwarz and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H. 264/AVC video compression standard," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 13, pp. 620-636, 2003.

[23] S. P. AMEE, "Design of High Speed CAVLC encoder for H.264 codec," *JOURNAL OF INFORMATION, KNOWLEDGE AND RESEARCH IN ELECTRONICS AND,* vol. 02, pp. 695-698, 2013.

[24] I. E. Richardson, *H. 264 and MPEG-4 Video Compression: Video Coding for Next-Generation Multimedia.* John Wiley & Sons, 2004.

[25] T. Wiegand, "Draft ITU-T recommendation and final draft international standard of joint video specification," *ITU-T Rec.H.264| ISO/IEC 14496-10 AVC,* 2003.

[26] I. Rec, "H. 264 & ISO/IEC 14496-10 AVC," *Advanced Video Coding for Generic Audiovisual Services.ITU-T,* 2003.

[27] W. Ye-Kui, M. M. Hannuksela, S. Pateux, A. Eleftheriadis and S. Wenger, "System and transport interface of SVC," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 17, pp. 1149-1163, 2007.

[28] H. Malepati, *Digital Media Processing: DSP Algorithms using C.* Newnes, 2010.

[29] L. Onural, T. Sikora and A. Smolic, "An overview of a new european consortium: Integrated three-dimensional television-capture, transmission and display (3DTV)." in *EWIMT,* 2004, .

[30] A. Smolic, K. Mueller, P. Merkle, C. Fehn, P. Kauff, P. Eisert and T. Wiegand, "3d video and free viewpoint video-technologies, applications and mpeg standards," in *Multimedia and Expo, 2006 IEEE International Conference on,* 2006, pp. 2161-2164.

[31] P. Merkle, A. Smolic, K. Muller and T. Wiegand, "Efficient prediction structures for multiview video coding," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 17, pp. 1461-1473, 2007.

[32] Y. Chen, M. M. Hannuksela, L. Zhu, A. Hallapuro, M. Gabbouj and H. Li, "Coding techniques in multiview video coding and joint multiview video model," in *Picture Coding Symposium, 2009. PCS 2009,* 2009, pp. 1-4.

[33] O. Neṁić, S. Rimac-Drlje and M. Vranjeˇs, "Multiview video coding extension of the H. 264/AVC standard," in *ELMAR, 2010 PROCEEDINGS,* 2010, pp. 73-76.

[34] Y. Chen, Y. Wang, K. Ugur, M. M. Hannuksela, J. Lainema and M. Gabbouj, "The emerging MVC standard for 3D video services," *EURASIP Journal on Applied Signal Processing,* vol. 2009, pp. 8, 2009.

[35] L. Shen, Z. Liu, T. Yan, Z. Zhang and P. An, "View-adaptive motion estimation and disparity estimation for low complexity multiview video coding," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 20, pp. 925-930, 2010.

[36] Y. Kim, J. Kim and K. Sohn, "Fast disparity and motion estimation for multi-view video coding," *Consumer Electronics, IEEE Transactions on,* vol. 53, pp. 712-719, 2007.

[37] X. Li, D. Zhao, X. Ji, Q. Wang and W. Gao, "A fast inter frame prediction algorithm for multi-view video coding," in *Image Processing, 2007. ICIP 2007. IEEE International Conference on,* 2007, pp. III-417-III-420.

[38] X. Guo, Y. Lu, F. Wu and W. Gao, "Inter-view direct mode for multiview video coding," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 16, pp. 1527-1532, 2006.

[39] G. Triantafyllidis, N. Grammalidis and M. Strintzis, "Coding of Stereoscopic and Three Dimensional Images and Video," *Encyclopedia of Multimedia,* pp. 63-67, 2008.

[40] A. Vetro, P. Pandit, H. Kimata, A. Smolic and Y. Wang, "Joint draft 8.0 on multiview video coding," *Joint Video Team, Doc.JVT-AB204,* 2008.

[41] T. Schierl and S. Narasimhan, "Transport and storage systems for 3-D video using MPEG-2 systems, RTP, and ISO file format," *Proc IEEE,* vol. 99, pp. 671-683, 2011.

[42] R. Skupin, P. Yue, T. Schierl and Y. Wang, "RTP Payload Format for MVC Video," 2012.

[43] I. Ali, S. Moiron, M. Fleury and M. Ghanbari, "Congestion resiliency of data partitioned H. 264/AVC video over wireless networks," in *Mobile Multimedia Communications*, 2012, pp. 1-14.

[44] *High Efficiency Video Coding | MPEG* . Available: http://mpeg.chiariglione.org/standards/mpeg-h/high-efficiency-video-coding [Last accessed 9/21/2014].

[45] *TVTechnology: HEVC to have a revolutionary impact on OTT video and UHDTV* . Available: http://www.tvtechnology.com/cable-satellite-iptv/0149/hevc-to-have-a-revolutionary-impact-on-ott-video-and-uhdtv/267459; [Last accessed 9/21/2014].

[46] G. J. Sullivan, J. Ohm, W. Han and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 22, pp. 1649-1668, 2012.

[47] *Next generation video compression - Ericsson* . Available: http://www.ericsson.com/news/130424-next-generation-video-compression_244129228_c [Last accessed 9/21/2014].

[48] *High Efficiency Video Coding (HEVC) Compression Standard | Harmonic Inc* [http://www.harmonicinc.com/content/hevc-focus [Last accessed 9/11/2014]].

[49] E. Peixoto, T. Shanableh and E. Izquierdo, "H. 264/AVC to HEVC Video Transcoder based on Dynamic Thresholding and Content Modeling," 2014.

[50] *HEVC: The next generation video compression standard | Ericsson Time to Play* . Available: http://www.ericsson.com/televisionary/blog/hevc-next-generation-video-compression-standard/ [Last accessed 9/11/2014].

[51] J. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan and T. Wiegand, "Comparison of the coding efficiency of video coding standards—including high efficiency video coding (HEVC)," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 22, pp. 1669-1684, 2012.

[52] T. Tan, M. Mrak, R. Weerakkody, N. Ramzan, V. Baroncini, G. Sullivan, J. Ohm and K. McCann, "HEVC subjective video quality test results," 2014.

[53] A. Vetro and D. Tian, "Analysis of 3D and multiview extensions of the emerging HEVC standard," in *SPIE Optical Engineering Applications,* 2012, pp. 84990Y-84990Y-7.

[54] M. Winken, H. Schwarz and T. Wiegand, "Motion vector inheritance for high efficiency 3D video plus depth coding," in *Picture Coding Symposium (PCS), 2012,* 2012, pp. 53-56.

[55] *3D HEVC Extension - Fraunhofer HHI* . Available: http://www.hhi.fraunhofer.de/fields-of-competence/image-processing/research-groups/image-video-coding/3d-hevc-extension.html [Last accessed 9/11/2014].

[56] P. G. Bhaktavathsalam, R. Adireddy and N. Palanisamy, "Novel approach limiting non-square prediction unit (PU) estimation in HEVC," in *Communications (NCC), 2014 Twentieth National Conference on,* 2014, pp. 1-5.

[57] J. Jiang, L. B. Guo, W. Mo and F. K. Fan, "Block-based parallel intra prediction scheme for HEVC," *Journal of Multimedia,* vol. 7, pp. 289-294, 2012.

[58] P. Merkle, K. Müller and T. Wiegand, "3D video: acquisition, coding, and display," *Consumer Electronics, IEEE Transactions on*, vol. 56, pp. 946-950, 2010.

[59] W. Schroeder, K. Martin and B. Lorensen, *An Object-Oriented Approach to 3D Graphics.* Prentice hall, 1997.

[60] B. A. Wandell, *Foundations of Vision.* Sinauer Associates, 1995.

[61] G. Su, Y. Lai, A. Kwasinski and H. Wang, "3D video communications: Challenges and opportunities," *International Journal of Communication Systems,* vol. 24, pp. 1261-1281, 2011.

[62] A. Vetro, S. Yea and A. Smolic, "Toward a 3D video format for auto-stereoscopic displays," in *Optical Engineering Applications*, 2008, pp. 70730F-70730F-10.

[63] A. Vetro, A. M. Tourapis, K. Muller and T. Chen, "3D-TV content storage and transmission," *Broadcasting, IEEE Transactions on,* vol. 57, pp. 384-394, 2011.

[64] J. Seo, D. Park, H. Wey, S. Lee and K. Sohn, "Motion information sharing mode for depth video coding," in *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2010,* 2010, pp. 1-4.

[65] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision,* vol. 47, pp. 7-42, 2002.

[66] A. Smolic, K. Mueller, P. Merkle, P. Kauff and T. Wiegand, "An overview of available and emerging 3D video formats and depth enhanced stereo as efficient generic solution," in *Picture Coding Symposium, 2009. PCS 2009,* 2009, pp. 1-4.

[67] K. Muller, P. Merkle and T. Wiegand, "3-D video representation using depth maps," *Proc IEEE,* vol. 99, pp. 643-656, 2011.

[68] C. Fehn, P. Kauff, M. O. De Beeck, F. Ernst, W. Ijsselsteijn, M. Pollefeys, L. Van Gool, E. Ofek and I. Sexton, "An evolutionary and optimised approach on 3D-TV," in *Proc. of IBC,* 2002, pp. 357-365.

[69] P. Merkle, Y. Wang, K. Muller, A. Smolic and T. Wiegand, "Video plus depth compression for mobile 3D services," in *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video, 2009,* 2009, pp. 1-4.

[70] A. Smolic, K. Mueller, N. Stefanoski, J. Ostermann, A. Gotchev, G. B. Akar, G. Triantafyllidis and A. Koz, "Coding algorithms for 3DTV—a survey," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 17, pp. 1606-1621, 2007.

[71] P. Merkle, A. Smolic, K. Muller and T. Wiegand, "Multi-view video plus depth representation and coding," in *Image Processing, 2007. ICIP 2007. IEEE International Conference on,* 2007, pp. I-201-I-204.

[72] J. Zhang, M. M. Hannuksela and H. Li, "Joint multiview video plus depth coding," in *Image Processing (ICIP), 2010 17th IEEE International Conference on,* 2010, pp. 2865-2868.

[73] F. Shao, G. Jiang, M. Yu, K. Chen and Y. Ho, "Asymmetric coding of multi-view video plus depth based 3-D video for view rendering," *Multimedia, IEEE Transactions on,* vol. 14, pp. 157-167, 2012.

[74] Q. Zhang, P. An, Y. Zhang, Q. Zhang and Z. Zhang, "Reduced resolution depth compression for multiview video plus depth coding," in *Signal Processing (ICSP), 2010 IEEE 10th International Conference on,* 2010, pp. 1145-1148.

[75] S. Yea and A. Vetro, "View synthesis prediction for multiview video coding," *Signal Process Image Commun,* vol. 24, pp. 89-100, 2009.

[76] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Muller, P. De With and T. Wiegand, "The effect of depth compression on multiview rendering quality," in *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video, 2008,* 2008, pp. 245-248.

[77] J. Shade, S. Gortler, L. He and R. Szeliski, "Layered depth images," in *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques,* 1998, pp. 231-242.

[78] K. Muller, A. Smolic, K. Dix, P. Kauff and T. Wiegand, "Reliability-based generation and view synthesis in layered depth video," in *Multimedia Signal Processing, 2008 IEEE 10th Workshop on,* 2008, pp. 34-39.

[79] K. R. Vijayanagar and J. Kim, "Compression of residual layers of layered depth video using hierarchical block truncation coding," in *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2012,* 2012, pp. 1-4.

[80] V. Jantet, L. Morin and C. Guillemot, "Incremental-ldi for multi-view coding," in *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video, 2009,* 2009, pp. 1-4.

[81] I. Daribo and H. Saito, "A novel inpainting-based layered depth video for 3DTV," *Broadcasting, IEEE Transactions on,* vol. 57, pp. 533-541, 2011.

[82] J. Ohm, "Overview of 3D video coding standardization," *Proceedings of 3DSA2013, Keynote Speech,* vol. 2, 2013.

[83] G. Balota, M. Saldanha, G. Sanchez, B. Zatt, M. Porto and L. Agostini, "Overview and quality analysis in 3D-HEVC emergent video coding standard," in *Circuits and Systems (LASCAS), 2014 IEEE 5th Latin American Symposium on,* 2014, pp. 1-4.

[84] K. Muller, H. Schwarz, D. Marpe, C. Bartnik, S. Bosse, H. Brust, T. Hinz, H. Lakshman, P. Merkle and F. H. Rhee, "3D high-efficiency video coding for multi-view video and depth data," *Image Processing, IEEE Transactions on,* vol. 22, pp. 3366-3378, 2013.

[85] F. Bossen, D. Flynn and K. Sühring, "HEVC reference software manual," *JCTVC-D404, Daegu, Korea,* 2011.

[86] B. Lee, J. Park and S. Min, "Three-dimensional display and information processing based on integral imaging," in *Digital Holography and Three-Dimensional Display*Anonymous Springer, 2006, pp. 333-378.

[87] G. Lawton, "3D displays without glasses: coming to a screen near you," *Computer,* vol. 44, pp. 17-19, 2011.

[88] C. Conti, P. Nunes and L. D. Soares, "New HEVC prediction modes for 3D holoscopic video coding," in *Image Processing (ICIP), 2012 19th IEEE International Conference on,* 2012, pp. 1325-1328.

[89] A. Aggoun, E. Tsekleves, M. R. Swash, D. Zarpalas, A. Dimou, P. Daras, P. Nunes and L. D. Soares, "Immersive 3D holoscopic video system," *MultiMedia, IEEE,* vol. 20, pp. 28-37, 2013.

[90] A. Agooun, O. A. Fatah, J. C. Fernandez, C. Conti, P. Nunes and L. D. Soares, "Acquisition, processing and coding of 3D holoscopic content for immersive video systems," in *3DTV-Conference: The True Vision-Capture, Transmission and Dispaly of 3D Video (3DTV-CON), 2013,* 2013, pp. 1-4.

[91] C. Conti, J. Lino, P. Nunes and L. D. Soares, "Spatial and temporal prediction scheme for 3D holoscopic video coding based on H. 264/AVC," in *Packet Video Workshop (PV), 2012 19th International,* 2012, pp. 143-148.

[92] C. Conti, L. Ducla Soares and P. Nunes, "Influence of self-similarity on 3D holoscopic video coding performance," in *Proceedings of the 18th Brazilian Symposium on Multimedia and the Web,* 2012, pp. 131-134.

[93] C. Conti, P. Nunes and L. Soares, "3D holoscopic video coding," *IEEE Trans Circuits Syst Video Technol,* .

[94] C. Conti, P. Nunes and L. D. Soares, "Impact of packet losses in scalable 3D holoscopic video coding," in *SPIE Photonics Europe,* 2014, pp. 91380E-91380E-15.

[95] B. Javidi and F. Okano, *Three-Dimensional Television, Video, and Display Technologies.* Springer, 2002.

[96] J. Konrad and M. Halle, "3-D displays and signal processing," *Signal Processing Magazine, IEEE,* vol. 24, pp. 97-111, 2007.

[97] N. Holliman, "3D display systems," *To Appear,* pp. 0-7503, 2005.

[98] H. Mohib, "End-to-end 3D video communication over heterogeneous networks," 2014.

[99] J. E. Caviedes, "The Evolution of Video Processing Technology and Its Main Drivers," *Proc IEEE,* vol. 100, pp. 872-877, 2012.

[100] H. Urey, K. V. Chellappan, E. Erden and P. Surman, "State of the art in stereoscopic and autostereoscopic displays," *Proc IEEE,* vol. 99, pp. 540-555, 2011.

[101] L. M. Meesters, W. A. IJsselsteijn and P. J. Seuntiens, "A survey of perceptual evaluations and requirements of three-dimensional TV," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 14, pp. 381-391, 2004.

[102] P. Boher, T. Leroux, T. Bignon and V. Collomb-Patton, "Multispectral polarization viewing angle analysis of circular polarized stereoscopic 3D displays," *Electronic Imaging, San Jose, Proc.SPIE,* vol. 7524, pp. 26, 2010.

[103] B. S. Santos, P. Dias, A. Pimentel, J. Baggerman, C. Ferreira, S. Silva and J. Madeira, "Head-mounted display versus desktop for 3D navigation in virtual reality: a user study," *Multimedia Tools Appl,* vol. 41, pp. 161-181, 2009.

[104] I. N. Kompanets and S. A. Gonchukov, "Volumetric 3D Liquid Crystal Displays," *Inf Disp,* vol. 20, pp. 24-26, 2004.

[105] T. Grossman and R. Balakrishnan, "The design and evaluation of selection techniques for 3D volumetric displays," in *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology,* 2006, pp. 3-12.

[106] F. Yaraş, H. Kang and L. Onural, "State of the art in holographic displays: A survey," *Journal of Display Technology,* vol. 6, pp. 443-454, 2010.

[107] N. Peyghambarian, S. Tay, P. Blanche, R. Norwood and M. Yamamoto, "Rewritable holographic 3D displays," *Opt. Photonics News,* vol. 19, pp. 22-27, 2008.

[108] W. Matusik and H. Pfister, "3D TV: A scalable system for real-time acquisition, transmission, and autostereoscopic display of dynamic scenes," in *ACM Transactions on Graphics (TOG),* 2004, pp. 814-824.

[109] M. Swash, O. Abdulfatah, E. Alazawi, T. Kalganova and J. Cosmas, "Adopting multiview pixel mapping for enhancing quality of holoscopic 3D scene in parallax barriers based holoscopic 3D displays," in *Broadband Multimedia Systems and Broadcasting (BMSB), 2014 IEEE International Symposium on,* 2014, pp. 1-4.

[110] C. Conti, J. Lino, P. Nunes, L. D. Soares and F. Pereira, "Non-scalable 3D holoscopic video," .

[111] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz and P. Hanrahan, "Light field photography with a hand-held plenoptic camera," *Computer Science Technical Report CSTR,* vol. 2, 2005.

[112] A. Smolic, "3D video and free viewpoint video—from capture to display," *Pattern Recognit,* vol. 44, pp. 1958-1968, 2011.

[113] W. Funk, "History of autostereoscopic cinema," in *IS&T/SPIE Electronic Imaging,* 2012, pp. 82880R-82880R-25.

[114] Y. Zhu and T. Zhen, "3D multi-view autostereoscopic display and its key technologie," in *Information Processing, 2009. APCIP 2009. Asia-Pacific Conference on,* 2009, pp. 31-35.

[115] N. A. Dodgson, J. Moore and S. Lang, "Multi-view autostereoscopic 3D display," in *International Broadcasting Convention,* 1999, .

[116] V. V. Saveljev, "Characteristics of moiré spectra in autostereoscopic three-dimensional displays," *Display Technology, Journal of,* vol. 7, pp. 259-266, 2011.

[117] J. Son, B. Javidi and K. Kwack, "Methods for displaying three-dimensional images," *Proc IEEE,* vol. 94, pp. 502-523, 2006.

[118] C. Gurler, B. Gorkemli, G. Saygili and A. M. Tekalp, "Flexible transport of 3-D video over networks," *Proc IEEE,* vol. 99, pp. 694-707, 2011.

[119] O. Schreer, P. Kauff and T. Sikora, *3D Videocommunication.* Wiley Online Library, 2005.

[120] L. Onural, T. Sikora, J. Ostermann, A. Smolic, M. R. Civanlar and J. Watson, "An assessment of 3DTV technologies," in *Proceedings of the NAB Broadcast Engineering Conference,* 2006, pp. 456-467.

[121] N. Hur, H. Lee, G. S. Lee, S. J. Lee, A. Gotchev and S. Park, "3DTV broadcasting and distribution systems," *Broadcasting, IEEE Transactions on,* vol. 57, pp. 395-407, 2011.

[122] U. Reimers, *DVB: The Family of International Standards for Digital Video Broadcasting.* Springer, 2005.

[123] G. Saygili, C. G. Gurler and A. M. Tekalp, "Evaluation of asymmetric stereo video coding and rate scaling for adaptive 3D video streaming," *Broadcasting, IEEE Transactions on,* vol. 57, pp. 593-601, 2011.

[124] C. T. Hewage and M. G. Martini, "Quality of experience for 3D video streaming," *Communications Magazine, IEEE,* vol. 51, pp. 101-107, 2013.

[125] S. Sedef Savas, C. Göktuğ Gürler, A. Murat Tekalp, E. Ekmekcioglu, S. Worrall and A. Kondoz, "Adaptive streaming of multi-view video over P2P networks," *Signal Process Image Commun,* vol. 27, pp. 522-531, 2012.

[126] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder and R. Szeliski, "High-quality video view interpolation using a layered representation," in *ACM Transactions on Graphics (TOG),* 2004, pp. 600-608.

[127] D. Tian, P. Lai, P. Lopez and C. Gomila, "View synthesis techniques for 3D video," *Applications of Digital Image Processing XXXII, Proceedings of the SPIE,* vol. 7443, pp. 74430T-74430T, 2009.

[128] A. Gotchev, "Computer technologies for 3d video delivery for home entertainment," in *Proceedings of the 9th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing,* 2008, pp. 1.

[129] A. Aggoun, P. Amon, I. Arbel, A. Chernilov, J. Cosmas, G. Garcia, A. Jari, S. Keller, M. Mattavelli and C. Kontopoulos, "Multimedia delivery in the future internet," 2008.

[130] P. Joveluro, H. Malekmohamadi, W. Fernando and A. Kondoz, "Perceptual video quality metric for 3D video quality assessment," in *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2010,* 2010, pp. 1-4.

[131] J. Klaue, B. Rathke and A. Wolisz, "EvalVid-A Video Quality Evaluation Tool-set," *Telecommunication Networks,* 2011.

[132] S. Wolf and M. Pinson, "Video quality measurement techniques," *2002.,* 2002.

[133] B. W. Micallef and C. J. Debono, "An analysis on the effect of transmission errors in real-time H. 264-MVC bit-streams," in *MELECON 2010-2010 15th IEEE Mediterranean Electrotechnical Conference,* 2010, pp. 1215-1220.

[134] M. H. Pinson, M. Barkowsky and P. Le Callet, "Selecting scenes for 2D and 3D subjective video quality tests," *EURASIP Journal on Image and Video Processing,* vol. 2013, pp. 1-12, 2013.

[135] I. R. Assembly, *Methodology for the Subjective Assessment of the Quality of Television Pictures.* International Telecommunication Union, 2003.

[136] P. ITU-T RECOMMENDATION, "Subjective video quality assessment methods for multimedia applications," 1999.

[137] F. Dufaux, B. Pesquet-Popescu and M. Cagnazzo, *Emerging Technologies for 3D Video: Creation, Coding, Transmission and Rendering.* John Wiley & Sons, 2013.

[138] S. Yasakethu, C. T. Hewage, W. A. C. Fernando and A. M. Kondoz, "Quality analysis for 3D video using 2D video quality models," *Consumer Electronics, IEEE Transactions on,* vol. 54, pp. 1969-1976, 2008.

[139] Video Quality Experts Group, "Final report from the Video Quality Experts Group on the validation of objective models of video quality assessment, Phase II (FR_TV2)," *Ftp://ftp.its.Bldrdoc.gov/dist/ituvidq/Boulder_VQEG_jan_04/VQEG_PhaseII_FRTV_Final_Report_SG9060E.Doc, 2003,* 2003.

[140] P. Seuntiens, L. Meesters and W. Ijsselsteijn, "Perceived quality of compressed stereoscopic images: Effects of symmetric and asymmetric JPEG coding and camera separation," *ACM Transactions on Applied Perception (TAP),* vol. 3, pp. 95-109, 2006.

[141] B. Alexandre, L. C. Patrick, C. Patrizio and C. Romain, "Quality assessment of stereoscopic images," *EURASIP Journal on Image and Video Processing,* vol. 2008, 2009.

[142] A. Tikanmaki, A. Gotchev, A. Smolic and K. Miller, "Quality assessment of 3D video in rate allocation experiments," in *Consumer Electronics, 2008. ISCE 2008. IEEE International Symposium on,* 2008, pp. 1-4.

[143] A. Kordelas, T. Dagiuklas and I. Politis, "On the performance of H. 264/MVC over lossy IP-based networks," in *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European,* 2012, pp. 1149-1153.

[144] O. H. Salim and W. Xiang, "A novel unequal error protection scheme for 3-D video transmission over cooperative MIMO-OFDM systems," *EURASIP Journal on Wireless Communications and Networking,* vol. 2012, pp. 269, 2012.

[145] T. Stockhammer, M. M. Hannuksela and T. Wiegand, "H. 264/AVC in wireless environments," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 13, pp. 657-673, 2003.

[146] H. Wang, L. Kondi, A. Luthra and S. Ci, *4G Wireless Video Communications.* John Wiley & Sons, 2009.

[147] S. Z. Azami, P. Duhamel and O. Rioul, "Joint source-channel coding: Panorama of methods," in *Proceedings of CNES Workshop on Data Compression,* 1996, pp. 1232-1254.

[148] Y. Wang and Q. Zhu, "Error control and concealment for video communication: A review," *Proc IEEE,* vol. 86, pp. 974-997, 1998.

[149] J. G. Apostolopoulos, "Reliable video communication over lossy packet networks using multiple state encoding and path diversity," in *Photonics West 2001-Electronic Imaging,* 2000, pp. 392-409.

[150] I. Moccagatta, S. Soudagar, J. Liang and H. Chen, "Error-resilient coding in JPEG-2000 and MPEG-4," *Selected Areas in Communications, IEEE Journal on,* vol. 18, pp. 899-914, 2000.

[151] G. Tech, H. Brust, K. Müller, A. Aksay and D. Bugdayci, "Development and optimization of coding algorithms for mobile 3DTV," *Tampere University of Technology, Tech.Rep,* 2009.

[152] S. Wenger, "H. 264/avc over ip," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 13, pp. 645-656, 2003.

[153] R. Razavi, M. Fleury, M. Altaf, H. Sammak and M. Ghanbari, "H. 264 video streaming with data-partitioning and growth codes," in *Image Processing (ICIP), 2009 16th IEEE International Conference on,* 2009, pp. 909-912.

[154] C. Zhu, Y. Li and X. Niu, *Streaming Media Architectures, Techniques and Applications: Recent Advances.* IGI Global, 2011.

[155] X. Zhang, X. Peng, S. Fowler and D. Wu, "Robust h. 264/avc video transmission using data partitioning and unequal loss protection," in *Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on,* 2010, pp. 2471-2477.

[156] A. Li, "RTP payload format for generic forward error correction," 2007.

[157] M. Van der Schaar and P. A. Chou, *Multimedia Over IP and Wireless Networks: Compression, Networking, and Systems.* Academic Press, 2011.

[158] A. B. Ibrahim and A. H. Sadka, "Implementation of error resilience technique in multiview video coding," in *IEEE Southwest Symposium on Image Analysis and Interpretation,* San Diego, California, 2014, pp. 1-4.

[159] B. Kamolrat, W. A. C. Fernando, M. Mrak and A. Kondoz, "Joint source and channel coding for 3D video with depth image-based rendering," *Consumer Electronics, IEEE Transactions on,* vol. 54, pp. 887-894, 2008.

[160] S. Kumar, L. Xu, M. K. Mandal and S. Panchanathan, "Error resiliency schemes in H. 264/AVC standard," *Journal of Visual Communication and Image Representation,* vol. 17, pp. 425-450, 2006.

[161] I. Ismaeil, S. Shirani, F. Kossentini and R. Ward, "An efficient, similarity-based error concealment method for block-based coded images," in *Image Processing, 2000. Proceedings. 2000 International Conference on,* 2000, pp. 388-391.

[162] W. Chu and J. Leou, "Detection and concealment of transmission errors in H. 261 images," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 8, pp. 74-84, 1998.

[163] Y. Dhondt, S. Mys, K. Vermeirsch and R. Van de Walle, "Constrained inter prediction: Removing dependencies between different data partitions," in *Advanced Concepts for Intelligent Vision Systems,* 2007, pp. 720-731.

[164] A. Kondoz, *Visual Media Coding and Transmission.* John Wiley & Sons, 2009.

[165] S. Yang, C. Chu and C. Chang, "An H. 264/AVC error concealment technique enhanced by depth correlation," in *Proceedings of the International Multi Conference of Engineers and Computer Scientists, IMECS,* 2012, .

[166] S. Argyropoulos, A. Tan, N. Thomos, E. Arikan and M. G. Strintzis, "Robust transmission of multi-view video streams using flexible macroblock ordering and systematic LT codes," in *3DTV Conference, 2007,* 2007, pp. 1-4.

[167] W. El-Shafai, "Improved 3D Multi-View Stereoscopic Video Decoding through Dispersed Flexible Macro-block Ordering and Multi-Dimensional Error Concealment," *International Journal of Advanced Science & Technology,* vol. 57, 2013.

[168] B. W. Micallef and C. J. Debono, "Error concealment techniques for multi-view video," in *Wireless Days (WD), 2010 IFIP,* 2010, pp. 1-5.

[169] K. Seo, J. Kim, S. Jung and J. Yoo, "A practical RTP packetization scheme for SVC video transport over IP networks," *ETRI J.,* vol. 32, pp. 281-291, 2010.

[170] A. Sood, N. K. Chilamkurti and B. Soh, "Analysis of error resilience in h. 264 video using slice interleaving technique," in *Information Networking. Advances in Data Communications and Wireless Networks*Anonymous Springer, 2006, pp. 912-921.

[171] D. Bugdayci, M. O. Bici, A. Aksay, M. Demirtas, G. B. Akar, A. Tikanmäki and A. Gotchev, *Stereo DVB-H Broadcasting System with Error Resilient Tools,* 2009.

[172] R. A. Farrugia and C. J. Debono, "Resilient Digital Video Transmission over Wireless Channels using Pixel-Level Artefact Detection Mechanisms," 2010.

[173] C. Lin, T. Tillo, Y. Zhao and B. Jeon, "Multiple description coding for H. 264/AVC with redundancy allocation at macro block level," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 21, pp. 589-600, 2011.

[174] C. Zhu, Y. Wang, M. M. Hannuksela and H. Li, "Error resilient video coding using redundant pictures," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 19, pp. 3-14, 2009.

[175] I. Radulovic, P. Frossard, Y. Wang, M. M. Hannuksela and A. Hallapuro, "Multiple description video coding with H. 264/AVC redundant pictures," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 20, pp. 144-148, 2010.

[176] V. A. Vaishampayan, "Design of multiple description scalar quantizers," *Information Theory, IEEE Transactions on,* vol. 39, pp. 821-834, 1993.

[177] T. Tillo, M. Grangetto and G. Olmo, "Redundant slice optimal allocation for H. 264 multiple description coding," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 18, pp. 59-70, 2008.

[178] C. Su, J. J. Yao and H. H. Chen, "H. 264/AVC-based multiple description coding scheme," in *Image Processing, 2007. ICIP 2007. IEEE International Conference on,* 2007, pp. IV-265-IV-268.

[179] M. B. Dissanayake, "A novel error robust video coding concept using motion vectors and parity bits," in *Industrial and Information Systems (ICIIS), 2012 7th IEEE International Conference on,* 2012, pp. 1-6.

[180] M. Dissanayake, S. Worrall and W. Fernando, "Error resilience for multi-view video using redundant macroblock coding," in *Industrial and Information Systems (ICIIS), 2011 6th IEEE International Conference on,* 2011, pp. 472-476.

[181] F. Boulos, W. Chen, B. Parrein and P. Le Callet, "Region-of-interest intra prediction for H. 264/AVC error resilience," in *Image Processing (ICIP), 2009 16th IEEE International Conference on,* 2009, pp. 3109-3112.

[182] I. A. Ali, S. Moiron, M. Fleury and M. Ghanbari, "Packet prioritization for H. 264/AVC video with cyclic intra-refresh line," *Journal of Visual Communication and Image Representation,* vol. 24, pp. 486-498, 2013.

[183] T. Stockhammer and W. Zia, "Error-resilient coding and decoding strategies for video communication," *Multimedia Over IP and Wireless Networks,* pp. 13-58, 2011.

[184] R. M. Schreier, A. T. I. Rahman, G. Krishnamurthy and A. Rothermel, "Architecture analysis for low-delay video coding," in *Multimedia and Expo, 2006 IEEE International Conference on,* 2006, pp. 2053-2056.

[185] X. Wang, C. Kodikara, A. Sadka and A. Kondoz, "Robust GOB intra refresh scheme for H. 264/AVC video over UMTS," in *3G and Beyond, 2005 6th IEE International Conference on,* 2005, pp. 1-4.

[186] P. Nunes, L. D. Soares and F. Pereira, "Automatic and adaptive network-aware macroblock intra refresh for error-resilient H. 264/AVC video coding," in *Image Processing (ICIP), 2009 16th IEEE International Conference on,* 2009, pp. 3073-3076.

[187] Y. Wang, M. M. Hannuksela and M. Gabbouj, "Error-robust inter/intra macroblock mode selection using isolated regions," *Proc.13th Packet Video Wksp,* 2003.

[188] J. Y. Liao and J. D. Villasenor, "Adaptive intra update for video coding over noisy channels," in *Image Processing, 1996. Proceedings., International Conference on,* 1996, pp. 763-766.

[189] T. Wiegand, N. Farber, K. Stuhlmuller and B. Girod, "Error-resilient video transmission using long-term memory motion-compensated prediction," *Selected Areas in Communications, IEEE Journal on,* vol. 18, pp. 1050-1062, 2000.

[190] P. Gao, W. Xiang, J. Billingsley and Y. Zhang, "Error-resilient multi-view video coding for next generation 3-D video broadcasting," in *ICT Convergence (ICTC), 2013 International Conference on,* 2013, pp. 1022-1026.

[191] P. Gao and W. Xiang, "Rate-distortion optimized mode switching for error-resilient multi-view video plus depth based 3-D video coding," 2014.

[192] Y. Wang, S. Wenger, J. Wen and A. K. Katsaggelos, "Error resilient video coding techniques," *Signal Processing Magazine, IEEE,* vol. 17, pp. 61-82, 2000.

[193] A. H. Sadka, *Compressed Video Communications.* Halsted Press, 2002.

[194] F. Luo, *Mobile Multimedia Broadcasting Standards: Technology and Practice.* springer, 2008.

[195] Y. Zhang, S. Kwong, G. Jiang and H. Wang, "Efficient multi-reference frame selection algorithm for hierarchical B pictures in multiview video coding," *Broadcasting, IEEE Transactions on,* vol. 57, pp. 15-23, 2011.

[196] M. Kitahara, H. Kimata, S. Shimizu, K. Kamikura, Y. Yashima, K. Yamamoto, T. Yendo, T. Fujii and M. Tanimoto, "Multi-view video coding using view interpolation and reference picture selection." in *ICME,* 2006, pp. 97-100.

[197] B. Macchiavello, C. Dorea, E. M. Hung, G. Cheung and W. Tan, "Reference frame selection for loss-resilient texture & depth map coding in multiview video conferencing," in *Image Processing (ICIP), 2012 19th IEEE International Conference on,* 2012, pp. 1653-1656.

[198] Y. Si, M. Yu, Z. Peng and G. Jiang, "A Fast Multi-reference Frame Selection Algorithm for Multiview Video Coding," *Journal of Multimedia,* vol. 5, pp. 369-376, 2010.

[199] M. Karczewicz and R. Kurceren, "The SP-and SI-frames design for H. 264/AVC," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 13, pp. 637-644, 2003.

[200] T. Stockhammer, G. Liebl and M. Walter, "Optimized H. 264/AVC-based bit stream switching for mobile video streaming," *EURASIP Journal on Applied Signal Processing,* vol. 2006, pp. 127-127, 2006.

[201] M. Altaf, E. Khan, M. Ghanbari and N. N. Qadri, "Efficient bitstream switching for streaming of H. 264/AVC coded video," *EURASIP Journal on Image and Video Processing,* vol. 2011, pp. 1-12, 2011.

[202] X. Guo, Y. Lu, W. Gao and Q. Huang, "Viewpoint switching in multiview video streaming," in *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on,* 2005, pp. 3471-3474.

[203] L. Al-Jobouri, M. Fleury and M. Ghanbari, "Protecting H. 264/AVC data-partitioned video streams over broadband WiMAX," *Advances in Multimedia,* vol. 2012, pp. 10, 2012.

[204] R. Talluri, "Error-resilient video coding in the ISO MPEG-4 standard," *Communications Magazine, IEEE,* vol. 36, pp. 112-119, 1998.

[205] M. G. Martini, M. Mazzotti, C. Lamy-Bergot, J. Huusko and P. Amon, "Content adaptive network aware joint optimization of wireless video transmission," *Communications Magazine, IEEE,* vol. 45, pp. 84-90, 2007.

[206] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Computing and Communications Review,* vol. 5, pp. 3-55, 2001.

[207] P. Yip, J. Malcolm, W. Fernando, K. Loo and H. K. Arachchi, "Joint source and channel coding for H. 264 compliant stereoscopic video transmission," in *Electrical and Computer Engineering, 2005. Canadian Conference on,* 2005, pp. 188-191.

[208] Y. Wang, M. M. Hannuksela, V. Varsa, A. Hourunranta and M. Gabbouj, "The error concealment feature in the H. 26L test model," in *Image Processing. 2002. Proceedings. 2002 International Conference on,* 2002, pp. II-729-II-732 vol. 2.

[209] W. Lam, A. R. Reibman and B. Liu, "Recovery of lost or erroneously received motion vectors," in *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on,* 1993, pp. 417-420.

[210] B. W. Micallef, C. J. Debono and R. A. Farrugia, "Performance of enhanced error concealment techniques in multi-view video coding systems," in *Systems, Signals and Image Processing (IWSSIP), 2011 18th International Conference on,* 2011, pp. 1-4.

[211] Y. Chen, "Advances on Coding and Transmission of Scalable Video and Multiview Video," *Tampereen Teknillinen Yliopisto.Julkaisu-Tampere University of Technology.Publication; 871,* 2010.

[212] Z. Wu and J. M. Boyce, "An error concealment scheme for entire frame losses based on H. 264/AVC," in *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on,* 2006, pp. 4 pp.

[213] G. J. Sullivan, P. N. Topiwala and A. Luthra, "The H. 264/AVC advanced video coding standard: Overview and introduction to the fidelity range extensions," in *Optical Science and Technology, the SPIE 49th Annual Meeting,* 2004, pp. 454-474.

[214] L. Casadesus, J. Fernández-Navajas, L. Sequeira, I. Quintana, J. Saldana and J. Ruiz-Mas, "IPTV quality assessment system," in *Proceedings of the 7th Latin American Networking Conference,* 2012, pp. 52-58.

[215] H. A. Sanneck and G. Carle, "Framework model for packet loss metrics based on loss runlengths," in *Electronic Imaging,* 1999, pp. 177-187.

[216] N. Staelens, I. Sedano, M. Barkowsky, L. Janowski, K. Brunnstrom and P. Le Callet, "Standardized toolchain and model development for video quality assessment—the mission of the joint effort group in VQEG," in *Quality of Multimedia Experience (QoMEX), 2011 Third International Workshop on,* 2011, pp. 61-66.

[217] O. Modeler, *OPNET Technologies Inc,* 2009.

[218] E. N. Gilbert, "Capacity of a Burst-Noise Channel," *Bell System Technical Journal,* vol. 39, pp. 1253-1265, 1960.

[219] C. Jiao, L. Schwiebert and B. Xu, "On modeling the packet error statistics in bursty channels," in *Local Computer Networks, 2002. Proceedings. LCN 2002. 27th Annual IEEE Conference on,* 2002, pp. 534-541.

[220] E. Elliott, "Estimates of Error Rates for Codes on Burst-Noise Channels," *Bell System Technical Journal,* vol. 42, pp. 1977-1997, 1963.

[221] A. Vetro, M. McGuire, W. Matusik, A. Behrens, J. Lee and H. Pfister, "Multiview video test sequences from MERL," *ISO/IEC JTC1/SG29/WG11, Document MPEG2005/M12077,* 2005.

[222] Y. Chen, P. Pandit and S. Yea, "WD 4 reference software for MVC," *ISO/IEC JTC/ISC29/WG11 and ITU,* 2009.

[223] T. Stockhammer and M. Bystrom, "H. 264/AVC data partitioning for mobile video communication," in *Image Processing, 2004. ICIP'04. 2004 International Conference on,* 2004, pp. 545-548.

[224] Y. Su, A. Vetro and A. Smolic, *Common Conditions for MVC.JVT-U211,* 2006.

[225] (http://www.pixeltools.com/ [Last accessed 9/25/2014]). *PixelTools - The MPEG Experts .*

[226] ([Last accessed 9/10/2014] http://www.codecian.com/index.html). *Powerful video codec analyzer for professionals & researchers .*

[227] (http://h264bitstream.sourceforge.net/ [Last accessed 9/25/2014]). *h264bitstream - a library to read and write H264 files .*

[228] C. Zhu and Y. Li, *Advanced Video Communications Over Wireless Networks.* CRC Press, 2013.

[229] B. Bing, *3D and HD Broadband Video Networking.* Artech House, 2010.

[230] G. Cote, F. Kossentini and S. Wenger, "Error resilience coding," *Compressed Video Over Networks,* pp. 309, 2000.

[231] D. Soldani, M. Li and R. Cuny, *QoS and QoE Management in UMTS Cellular Systems.* John Wiley & Sons, 2007.

[232] O. Hohlfeld, "Stochastic packet loss model to evaluate QoE impairments," *PIK-Praxis Der Informationsverarbeitung Und Kommunikation,* vol. 32, pp. 53-56, 2009.

[233] M. Ebian, M. El-Sharkawy and S. El-Ramly, "Enhanced dynamic error concealment algorithm for multiview coding based on lost MBs sizes and adaptively selected candidates MBs," in *Proceedings of the Fourth International Conference on Signal and Image Processing 2012 (ICSIP 2012),* 2013, pp. 435-443.

[234] D. Wu, Y. T. Hou and Y. Zhang, "Transporting real-time video over the Internet: Challenges and approaches," *Proc IEEE,* vol. 88, pp. 1855-1877, 2000.

[235] M. Flierl and B. Girod, "Generalized B pictures and the draft H. 264/AVC video-compression standard," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 13, pp. 587-597, 2003.

Appendices

Appendix – A: List of Publications

Below is the list of publications that are related to the research work in this thesis:

7.2.1. Journal Publications

I. Abdulkareem Bebeji Ibrahim and Abdul H. Sadka, “Multi-Layer Data Partitioning for Multiview Video Coding” Signal and Image Processing: An International Journal. ISSN: 0976 – 710X; 2229 – 3922 (print), September 2014.

II. Abdulkareem Bebeji Ibrahim and Abdul H. Sadka, “Error Resilient for Multiview Video Transmissions with GOP Analysis” International Journal of Multimedia and Its Applications. Volume 6, Number 6, December 2014.

7.2.2. Conference Papers

I. Abdulkareem Bebeji Ibrahim and Abdul H. Sadka, “Implementation of Error Resilient Technique for Multiview Video Coding.” In Southwest Symposium on Image Analysis and Interpretation (ISSIAI), 2014 IEEE International Symposium on, pp. 25-28. IEEE, 2014.

II. Abdulkareem Bebeji Ibrahim and Abdul H. Sadka, “Error Resilience and Concealment for Multiview Video Coding.” In Broadband Multimedia Systems and Broadcasting (BMBS), International Symposium. IEEE, 2014.

III. Abdulkareem Bebeji Ibrahim and Abdul H. Sadka, “Effects of GOP on Multiview Video Coding over Error Prone Channels.” In Third International Conference on Advanced Information Technologies and Applications (ICAITA), International Conference. ICAITA, 2014.

IV. Sadka Abdul, Hamdullah Mohib, Abdulkareem Bebeji Ibrahim, and Mohd. M. Salzali, “Compression of 3D Stereoscopic Video Using ITU-T H.264/AVC.” In Second Abu Dhabi University Annual Research Conference, 2013.

V. Abdulkareem Bebeji Ibrahim and Abdul H. Sadka, “Error Resilience in 3D Multiview Video over UMTS Network.” In Research Student Conference, School of Engineering and Design, ResCon13, Brunel University, June 2013.

VI. Abdulkareem Bebeji Ibrahim and Abdul H. Sadka, “Implementation of Error Resilience and Concealment Technique for Multiview Video Coding.” In In Research Student Conference, School of Engineering and Design, ResCon13, Brunel University, June 2014.

Appendix – B1

Encoder Configuration File coding parameters and settings

```
# JMVM Configuration File in MVC mode
#=========GENERAL ================================================
InputFile  ballroom
OutputFile output
ReconFile  recon_ballroom
SourceWidth           640       # input  frame width
SourceHeight          480       # input  frame height
FrameRate            25.0       # frame rate [Hz]
FramesToBeEncoded       250       # number of frames

#========CODING ================================================
SymbolMode            0         # 0=CAVLC, 1=CABAC
FRExt               0         # 8x8 transform (0:off, 1:on)
BasisQP              31        # Quantization parameters

#========STRUCTURE ================================================
GOPSize              12   # GOP Size (at maximum frame rate)
IntraPeriod           12   # Anchor Period
NumberReferenceFrames   2         # Number of reference pictures
InterPredPicsFirst      1         # 1 Inter Pics; 0 Inter-view
```

```
DeltaLayer0Quant        0          # differential QP for layer 0
DeltaLayer1Quant        3          # differential QP for layer 1
DeltaLayer2Quant        4          # differential QP for layer 2
DeltaLayer3Quant        5          # differential QP for layer 3
DeltaLayer4Quant        6          # differential QP for layer 4
DeltaLayer5Quant        7          # differential QP for layer 5
#=============== MOTION SEARCH ====================================
SearchMode              4          # Search mode (0:BlockSearch, 4:FastSearch)
SearchFuncFullPel       3          # Search function full pel
                               #   (0:SAD, 1:SSE, 2:HADAMARD, 3:SAD-YUV)
SearchFuncSubPel        2          # Search function sub pel
                               #   (0:SAD, 1:SSE, 2:HADAMARD)
SearchRange             16         # Search range (Full Pel)
BiPredIter              4          # Max iterations for bi-pred search
IterSearchRange         8          # Search range for iterations (0: normal)

#=================LOOP FILTER =====================================
LoopFilterDisable       0          # Loop filter idc (0: on, 1: off, 2:
                               #   on except for slice boundaries)
LoopFilterAlphaC0Offset 0          # AlphaOffset(-6..+6): valid range
LoopFilterBetaOffset    0          # BetaOffset (-6..+6): valid range

#================WEIGHTED PREDICTION ==============================
WeightedPrediction      0          # Weighting IP Slice (0:disable, 1:enable)
WeightedBiprediction    0          # Weighting B  Slice (0:disable, 1:explicit,
                                                  2:implicit)

#=====PARALLEL DECODING INFORMATION SEI Message ===================
PDISEIMessage           0          # PDI SEI message enable (0: disable, 1:enable)
PDIInitialDelayAnc      2          # PDI initial delay for anchor pictures
PDIInitialDelayNonAnc   2          # PDI initial delay for non-anchor pictures
#=============SEQUENCE PARAMETER SET ==========================
NumViewsMinusOne 2        # (Number of view to be coded minus 1)
ViewOrder             0-2-1   # (Order in which view_ids are coded)
View_ID                        0         # (view_id of a view 0 - 1024)



Fwd_NumAnchorRefs          0         # (number of list_0 references for anchor)
Bwd_NumAnchorRefs          0         # (number of list 1 references for anchor)
Fwd_NumNonAnchorRefs    0         # (number of list 1 references for non-anchor)
Bwd_NumNonAnchorRefs    0         # (number of list 1 references for non-anchor)

View_ID                1
Fwd_NumAnchorRefs           1
Bwd_NumAnchorRefs           1
Fwd_NumNonAnchorRefs     1
Bwd_NumNonAnchorRefs     1
Fwd_AnchorRefs                  0 0
Bwd_AnchorRefs              0 2
Fwd_NonAnchorRefs  0 0
Bwd_NonAnchorRefs 0 2

View_ID                2
Fwd_NumAnchorRefs           1
Bwd_NumAnchorRefs           0
Fwd_NumNonAnchorRefs     0
Bwd_NumNonAnchorRefs     0
Fwd_AnchorRefs              0 0

#================Assembler: View Encode order ========================
OutputFile            ballroom1.264
NumberOfViews          3
InputFile0            output_0.264
InputFile1            output_2.264
InputFile2            output_1.264
```

Multi-Layer Data Partitioning Application

```
#define NALU_SIZE 1024*1024

typedef struct nal_data_s
{
      uint8_t           *nal_buf;
      uint32_t    size;
} nal_data;

void alloc_nal_queue(nal_data **nal_queue, int num_views)
{
      int i;

      *nal_queue = (nal_data*) malloc(num_views*3*sizeof(nal_data));

      for(i = 0; i < num_views*3; i++)
            (*nal_queue)[i].nal_buf = (uint8_t*) malloc(NALU_SIZE);
}

void init_nal_queue(nal_data *nal_queue, int num_views)
{
      int i;

      for(i = 0; i < num_views*3; i++)
            nal_queue[i].size = 0;
}

void destroy_nal_queue(nal_data *nal_queue, int num_views)
{
      int i;

      for(i = 0; i < num_views*3; i++)
            free(nal_queue[i].nal_buf);

      free(nal_queue);
}

void main(int argc, char *argv[])
{
      int nalu_count = 0;
      file_handle_t* file_handle;
      uint8_t* in_nalu_buffer = (uint8_t*) malloc(NALU_SIZE);
      uint8_t* out_nalu_buffer[3];

      int32_t in_nalu_size, out_nalu_size[3];
      FILE *outputFile = NULL;
      h264_data_partitioner_t *data_partitioner;
      nalu_type_t nalu_type;
      nal_data *nal_queue = NULL;
      int num_views = 0;
      int au_slice_count = 0;

      out_nalu_buffer[0] = (uint8_t*) malloc(NALU_SIZE);
      out_nalu_buffer[1] = (uint8_t*) malloc(NALU_SIZE);
      out_nalu_buffer[2] = (uint8_t*) malloc(NALU_SIZE);
      if(argc < 3)
      {
            printf("Insufficient arguments.\n");
            printf("Usage:\n");
            printf("\t H264DataPartitioner <input_file> <output_file>\n");
            return;
      }

      file_handle = file_handle_open(argv[1]);

      if(!file_handle)
      {
            printf("Unable to open input file for reading: %s", argv[1]);
            goto cleanup;
      }

      outputFile = fopen(argv[2], "wb");

      if(!outputFile)
      {
            printf("Unable to open output file for writing: %s", argv[2]);
            goto cleanup;
      }

      data_partitioner = h264_data_partitioner_init();

      if(!data_partitioner)
      {
            printf("Unable to initialize data_partitioner");
            goto cleanup;
      }

      while(file_handle_read_nalu(file_handle,              in_nalu_buffer,
&in_nalu_size) != 0 || in_nalu_size != 0)
      {
            printf("<----------------------------------------------->\n");
            printf("Input: %d\n", nalu_count);
            h264_data_partitioner_process(data_partitioner, in_nalu_buffer,
in_nalu_size, out_nalu_buffer, out_nalu_size, &nalu_type);
            printf("<---------------------------------------------->\n\n");

            switch(nalu_type)
            {
            case NAL_UNIT_SUBSET_SPS:
                  num_views              =              data_partitioner-
>SeqParSet[1].num_views_minus_1+1;
                  alloc_nal_queue(&nal_queue, num_views);
            case NAL_UNIT_SPS:
            case NAL_UNIT_PPS:
                  fwrite(out_nalu_buffer[0],      1,      out_nalu_size[0],
outputFile);
                  break;
case NAL_UNIT_CODED_SLICE_PREFIX:
                  if(au_slice_count)
                  {
                        int i;

                        for(i = 0; i < num_views*3; i++)
                              fwrite(nal_queue[i].nal_buf,              1,
nal_queue[i].size, outputFile);
                  }
                  init_nal_queue(nal_queue, num_views);
                  fwrite(out_nalu_buffer[0],      1,      out_nalu_size[0],
outputFile);
                  au_slice_count = 0;
                  break;

            case NAL_UNIT_CODED_SLICE_IDR:
            case NAL_UNIT_CODED_SLICE:
            case NAL_UNIT_CODED_SLICE_SCALABLE:
                  nal_queue[au_slice_count].size = out_nalu_size[0];
                  memcpy(nal_queue[au_slice_count].nal_buf,
out_nalu_buffer[0], out_nalu_size[0]);
                  au_slice_count++;
                  break;

            case NAL_UNIT_CODED_SLICE_DATAPART_A:
            case NAL_UNIT_CODED_SLICE_SCALABLE_DATAPART_A:
                  //copy Data Partition A
                  nal_queue[au_slice_count].size = out_nalu_size[0];
                  memcpy(nal_queue[au_slice_count].nal_buf,
out_nalu_buffer[0], out_nalu_size[0]);

                  //copy Data Partition B
                  nal_queue[au_slice_count     +     num_views].size     =
out_nalu_size[1];
                  memcpy(nal_queue[au_slice_count   +   num_views].nal_buf,
out_nalu_buffer[1], out_nalu_size[1]);

                  //copy Data Partition C
                  nal_queue[au_slice_count    +    num_views*2].size    =
out_nalu_size[2];
                  memcpy(nal_queue[au_slice_count  +  num_views*2].nal_buf,
out_nalu_buffer[2], out_nalu_size[2]);

                  au_slice_count++;
                  break;
            }

            //fwrite(out_nalu_buffer, 1, out_nalu_size, outputFile);
            in_nalu_size = 0;
            nalu_count++;
      }

      if(au_slice_count)
      {
            int i;
            for(i = 0; i < num_views*3; i++)
                  fwrite(nal_queue[i].nal_buf,    1,    nal_queue[i].size,
outputFile);
      }
```

```
cleanup:

      if(nal_queue)
            destroy_nal_queue(nal_queue, num_views);

      if(in_nalu_buffer)
            free(in_nalu_buffer);

      free(out_nalu_buffer[0]);
      free(out_nalu_buffer[1]);
      free(out_nalu_buffer[2]);

      if(data_partitioner)
            h264_data_partitioner_close(data_partitioner);

      if(file_handle)
            file_handle_close( file_handle );

      if(outputFile)
            fclose(outputFile);
}
```

Multi-Layer Data Partitioned bitstream (First 11 NALUs)

```
<------------------------------------------------------------------------->
Input: 0
        NAL Unit type: NAL_UNIT_SPS
        NAL Unit size: 15
Output:
        Total Output Size: 15
<------------------------------------------------------------------------->
Input: 1
        NAL Unit type: NAL_UNIT_SUBSET_SPS
        NAL Unit size: 24
Output:
        Total Output Size: 24
<------------------------------------------------------------------------->
Input: 2
        NAL Unit type: NAL_UNIT_PPS
        NAL Unit size: 9
```

```
Output:
        Total Output Size: 9
<---------------------------------------------------------------------->
Input: 3
        NAL Unit type: NAL_UNIT_PPS
        NAL Unit size: 9
Output:
        Total Output Size: 9
<---------------------------------------------------------------------->
Input: 4
        NAL Unit type: NAL_UNIT_CODED_SLICE_PREFIX
        NAL Unit size: 8
Output:
        Total Output Size: 8
<---------------------------------------------------------------------->
Input: 5
        NAL Unit type: NAL_UNIT_CODED_SLICE_IDR
        NAL Unit size: 19314
Output:
        Total Output Size: 19314
<---------------------------------------------------------------------->
Input: 6
        NAL Unit type: NAL_UNIT_CODED_SLICE_SCALABLE
        Anchor Pic Flag: 1
        NAL Unit size: 11204
Output:
        Total Output Size: 11204
<---------------------------------------------------------------------->
```

```
Input: 7
        NAL Unit type: NAL_UNIT_CODED_SLICE_SCALABLE
        Anchor Pic Flag: 1
        NAL Unit size: 9148
Output:
        Total Output Size: 9148
<-------------------------------------------------------------------->
Input: 8
        NAL Unit type: NAL_UNIT_CODED_SLICE_PREFIX
        NAL Unit size: 8
Output:
        Total Output Size: 8
<-------------------------------------------------------------------->
Input: 9
        NAL Unit type: NAL_UNIT_CODED_SLICE
        NAL Unit size: 17810
Output:
        NAL Unit type: NAL_UNIT_CODED_SLICE_DATAPART_A
        NAL Unit size: 5196
        NAL Unit type: NAL_UNIT_CODED_SLICE_DATAPART_B
        NAL Unit size: 12621
        NAL Unit type: NAL_UNIT_CODED_SLICE_DATAPART_C
        NAL Unit size: 6
        Total Output Size: 5196
<-------------------------------------------------------------------->
Input: 10
        NAL Unit type: NAL_UNIT_CODED_SLICE_SCALABLE
        Anchor Pic Flag: 1
        NAL Unit size: 11168
Output:
        Total Output Size: 11168
<-------------------------------------------------------------------->
```

MAP for first 55 NALUs of Multi-Layer partitioned bitstream

0. NAL_UNIT_SPS

1. NAL_UNIT_SUBSET_SPS

2. NAL_UNIT_PPS

3. NAL_UNIT_PPS

4. NAL_UNIT_CODED_SLICE_PREFIX

5. NAL_UNIT_CODED_SLICE_IDR

6. NAL_UNIT_CODED_SLICE_SCALABLE

7. NAL_UNIT_CODED_SLICE_SCALABLE

8. NAL_UNIT_CODED_SLICE_PREFIX

9. NAL_UNIT_CODED_SLICE_DATAPART_A

10. NAL_UNIT_CODED_SLICE_SCALABLE

11. NAL_UNIT_CODED_SLICE_SCALABLE

12. NAL_UNIT_CODED_SLICE_DATAPART_B

13. NAL_UNIT_CODED_SLICE_DATAPART_C

14. NAL_UNIT_CODED_SLICE_PREFIX

15. NAL_UNIT_CODED_SLICE_DATAPART_A

16. NAL_UNIT_CODED_SLICE_SCALABLE_DATAPART_A

17. NAL_UNIT_CODED_SLICE_SCALABLE_DATAPART_A

18. NAL_UNIT_CODED_SLICE_DATAPART_B

19. NAL_UNIT_CODED_SLICE_SCALABLE_DATAPART_B

20. NAL_UNIT_CODED_SLICE_SCALABLE_DATAPART_B

21. NAL_UNIT_CODED_SLICE_DATAPART_C

22. NAL_UNIT_CODED_SLICE_SCALABLE_DATAPART_C

23. NAL_UNIT_CODED_SLICE_SCALABLE_DATAPART_C

24. NAL_UNIT_CODED_SLICE_PREFIX

25. NAL_UNIT_CODED_SLICE_DATAPART_A

26. NAL_UNIT_CODED_SLICE_SCALABLE_DATAPART_A

27. NAL_UNIT_CODED_SLICE_SCALABLE_DATAPART_A

28. NAL_UNIT_CODED_SLICE_DATAPART_B

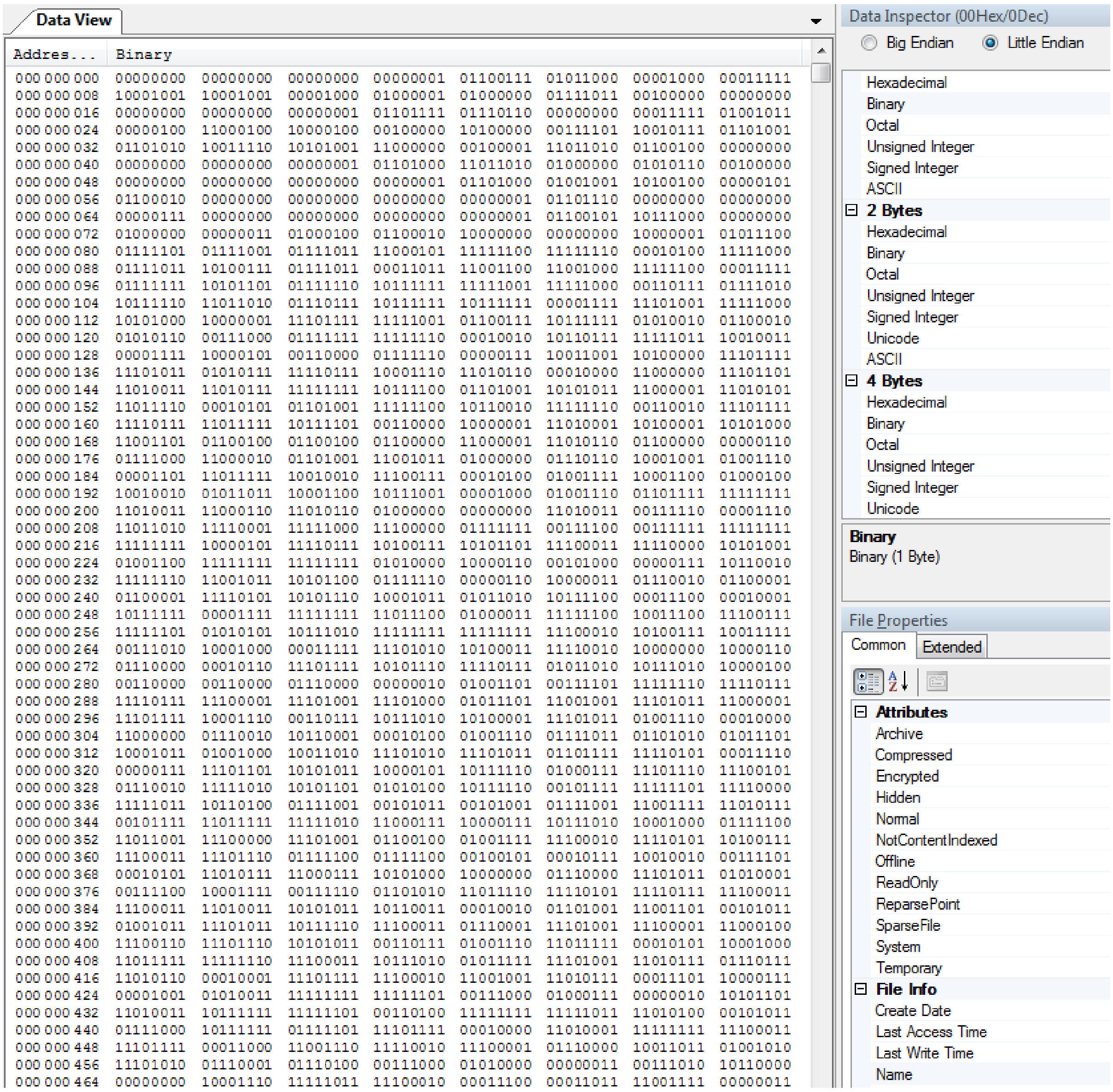
29. NAL_UNIT_CODED_SLICE_SCALABLE_DATAPART_B

30. NAL_UNIT_CODED_SLICE_SCALABLE_DATAPART_B

31. NAL_UNIT_CODED_SLICE_DATAPART_C

32. NAL_UNIT_CODED_SLICE_SCALABLE_DATAPART_C
33. NAL_UNIT_CODED_SLICE_SCALABLE_DATAPART_C
34. NAL_UNIT_CODED_SLICE_PREFIX
35. NAL_UNIT_CODED_SLICE_DATAPART_A
36. NAL_UNIT_CODED_SLICE_SCALABLE_DATAPART_A
37. NAL_UNIT_CODED_SLICE_SCALABLE_DATAPART_A
38. NAL_UNIT_CODED_SLICE_DATAPART_B
39. NAL_UNIT_CODED_SLICE_SCALABLE_DATAPART_B
40. NAL_UNIT_CODED_SLICE_SCALABLE_DATAPART_B
41. NAL_UNIT_CODED_SLICE_DATAPART_C
42. NAL_UNIT_CODED_SLICE_SCALABLE_DATAPART_C
43. NAL_UNIT_CODED_SLICE_SCALABLE_DATAPART_C
44. NAL_UNIT_CODED_SLICE_PREFIX
45. NAL_UNIT_CODED_SLICE_DATAPART_A
46. NAL_UNIT_CODED_SLICE_SCALABLE_DATAPART_A
47. NAL_UNIT_CODED_SLICE_SCALABLE_DATAPART_A
48. NAL_UNIT_CODED_SLICE_DATAPART_B
49. NAL_UNIT_CODED_SLICE_SCALABLE_DATAPART_B
50. NAL_UNIT_CODED_SLICE_SCALABLE_DATAPART_B
51. NAL_UNIT_CODED_SLICE_DATAPART_C
52. NAL_UNIT_CODED_SLICE_SCALABLE_DATAPART_C
53. NAL_UNIT_CODED_SLICE_SCALABLE_DATAPART_C
54. NAL_UNIT_CODED_SLICE_PREFIX

Binary view of the H.264 video file



```
Addres...    Binary
000 000 000  00000000 00000000 00000000 00000001 01100111 01011000 00001000 00011111
000 000 008  10001001 10001001 00001000 01000001 01000000 01111011 00100000 00000000
000 000 016  00000000 00000000 00000001 01101111 01110110 00000000 00011111 01001011
000 000 024  00000100 11000100 10000100 00100000 10100000 00111101 10010111 01101001
000 000 032  01101010 10011110 10101001 11000000 00100001 11011010 01100100 00000000
000 000 040  00000000 00000000 00000001 01101000 11011010 01000000 01010110 00100000
000 000 048  00000000 00000000 00000000 00000001 01101000 01001001 10100100 00000101
000 000 056  01100010 00000000 00000000 00000000 00000001 01101110 00000000 00000000
000 000 064  00000111 00000000 00000000 00000000 00000001 01100101 10111000 00000000
000 000 072  01000000 00000011 01000100 01100010 10000000 00000000 10000001 01011100
000 000 080  01111101 01111001 01111011 11000101 11111100 11111110 00010100 11111000
000 000 088  01111011 10100111 01111011 00011011 11001100 11001000 11111100 00011111
000 000 096  01111111 10101101 01111110 10111111 11111001 11111000 00110111 01111010
000 000 104  10111110 11011010 01110111 10111111 10111111 00001111 11101001 11111000
000 000 112  10101000 10000001 11101111 11111001 01100111 10111111 01010010 01100010
000 000 120  01010110 00111000 01111111 11111110 00010010 10110111 11111011 10010011
000 000 128  00001111 10000101 00110000 01111110 00000111 10011001 10100000 11101111
000 000 136  11101011 01010111 11110111 10001110 11010110 00010000 11000000 11101101
000 000 144  11010011 11010111 11111111 10111100 01101001 10101011 11000001 11010101
000 000 152  11011110 00010101 01101001 11111100 10110010 11111110 00110010 11101111
000 000 160  11110111 11011111 10111101 00110000 10000001 11010001 10100001 10101000
000 000 168  11001101 01100100 01100100 01100000 11000001 11010110 01100000 00000110
000 000 176  01111000 11000010 01101001 11001011 01000000 01110110 10001001 01001110
000 000 184  00001101 11011111 10010010 11100111 00010100 01001111 10001100 01000100
000 000 192  10010010 01011011 10001100 10111001 00001000 01001110 01101111 11111111
000 000 200  11010011 11000110 11010110 01000000 00000000 11010011 00111110 00001110
000 000 208  11011010 11110001 11111000 11100000 01111111 00111100 00111111 11111111
000 000 216  11111111 10000101 11110111 10100111 10101101 11100011 11110000 10101001
000 000 224  01001100 11111111 11111111 01010000 10000110 00101000 00000111 10110010
000 000 232  11111110 11001011 10101100 01111110 00000110 10000011 01110010 01100001
000 000 240  01100001 11110101 10101110 10001011 01011010 10111100 00011100 00010001
000 000 248  10111111 00001111 11111111 11011100 01000011 11111100 10011100 11100111
000 000 256  11111101 01010101 10111010 11111111 11111111 11100010 10100111 10011111
000 000 264  00111010 10001000 00011111 11101010 10100011 11110010 10000000 10000110
000 000 272  01110000 00010110 11101111 10101110 11110111 01011010 10111010 10000100
000 000 280  00110000 00110000 01110000 00000010 01001101 00111101 11111110 11110111
000 000 288  11110111 11100001 11101001 11101000 01011101 11001001 11101011 11000001
000 000 296  11101111 10001110 00110111 10111010 10100001 11101011 01001110 00010000
000 000 304  11000000 01110010 10110001 00010100 01001110 01111011 01101010 01011101
000 000 312  10001011 01001000 10011010 11101010 11101011 01101111 11110101 00011110
000 000 320  00000111 11101101 10101011 10000101 10111110 01000111 11101110 11100101
000 000 328  01110010 11111010 10101101 01010100 10111110 00101111 11111101 11110000
000 000 336  11111011 10110100 01111001 00101011 00101001 01111001 11001111 11010111
000 000 344  00101111 11011111 11111010 11000111 10000111 10111010 10001000 01111100
000 000 352  11011001 11100000 11101001 01100100 01001111 11100010 11110101 10100111
000 000 360  11100011 11101110 01111100 01111100 00100101 00010111 10010010 00111101
000 000 368  00010101 11010111 11000111 10101000 10000000 01110000 11101011 01010001
000 000 376  00111100 10001111 00111110 01101010 11011110 11110101 11110111 11100011
000 000 384  11100011 11010011 10101011 10110011 00010010 01101001 11001101 00101011
000 000 392  01001011 11101011 10111110 11100011 01110001 11101001 11100001 11000100
000 000 400  11100110 11101110 10101011 00110111 01001110 11011111 00010101 10001000
000 000 408  11011111 11111110 11100011 10111010 01011111 11101001 11010111 01110111
000 000 416  11010110 00010001 11101111 11100010 11001001 11010111 00011101 10000111
000 000 424  00001001 01010011 11111111 11111101 00111000 01000111 00000010 10101101
000 000 432  11010011 10111111 11111101 00110100 11111111 11111011 11010100 00101011
000 000 440  01111000 10111111 01111101 11101111 00010000 11010001 11111111 11100011
000 000 448  11101111 00011000 11001110 11110010 11100001 01110000 10011011 01001010
000 000 456  11101010 01110001 01110100 00111000 01010000 00000011 00111010 10110000
000 000 464  00000000 10001110 11111011 11100010 00011100 00011011 11001111 00000011
```

Figure – B1 1

Figure – B1 1 depicts a view of the H.264 video file in binary mode. This piece of software allows the users to find and view hexadecimal values and their addresses of the content of a video file in addition to equivalent Big/Endian and ASCII values. Some versions of the software could be used to modify or edit the video file for further analysis such as flipping bits in order to impair a bitstream manually. The video file can be viewed in different modes such as hexadecimal (1, 2, or 4 Byte mode), unsigned integer (1 or 2 Byte mode) etc.

Appendix – B2

Modified decoder for Multi-layer data partitioned bitstream

```
DPErrorCode   H264AVCDecoder::dpCheck(NalUnitType   prevNalu,   NalUnitType   currNalu,
NalUnitType& eExpectedNalu)
{
       if(prevNalu == NAL_UNIT_CODED_SLICE_DATAPART_A)
       {
              if(currNalu == NAL_UNIT_CODED_SLICE_DATAPART_B)
              {
                     // Everything is fine
                     m_dpPresent[DATA_PART_B] = true;
                     return DP_OK;
              }
              else if(currNalu == NAL_UNIT_CODED_SLICE_DATAPART_C)
              {
                     // DP B is missing. But decoding should continue.
                     m_dpPresent[DATA_PART_B] = false;
                     m_dpPresent[DATA_PART_C] = true;
                     return DP_CONTINUE;
              }
              else
              {
                     // DP B & C are missing. But the decoding should continue
                     // the bitstream should be rewind to put the current NALU
                     // back into the stream for reading again.
                     m_dpPresent[DATA_PART_B] = false;
                     m_dpPresent[DATA_PART_C] = false;
                     eExpectedNalu = NAL_UNIT_CODED_SLICE_DATAPART_C;
                     return DP_REWIND_AND_CONTINUE;
              }
       }
       else if(prevNalu == NAL_UNIT_CODED_SLICE_DATAPART_B)
       {
              if(currNalu == NAL_UNIT_CODED_SLICE_DATAPART_C)
              {
                     // Everything is fine.
                     m_dpPresent[DATA_PART_C] = true;
                     return DP_OK;
              }
              else
              {
                     // DP C is missing. But the decoding should continue
                     // the bitstream should be rewind to put the current NALU
                     // back into the stream for reading again.
                     m_dpPresent[DATA_PART_C] = false;
                     eExpectedNalu = NAL_UNIT_CODED_SLICE_DATAPART_C;
                     return DP_REWIND_AND_CONTINUE;
              }
       }
       else   if(   (currNalu   ==   NAL_UNIT_CODED_SLICE_DATAPART_B   ||   currNalu   ==
NAL_UNIT_CODED_SLICE_DATAPART_C) &&
                      m_dpPresent[DATA_PART_A] == false )
       {
              // DP A is missing
              m_dpPresent[DATA_PART_A] = false;
              m_dpPresent[NalUnitType2DataPart[currNalu]] = true;
              return DP_MISSING_A;
       }
```

```
        else if( currNalu == NAL_UNIT_CODED_SLICE_DATAPART_A )
        {
                m_dpPresent[DATA_PART_A] = true;
                return DP_OK;
        }
        else if(prevNalu == NAL_UNIT_CODED_SLICE_SCALABLE_DATAPART_A)
        {
                if(currNalu == NAL_UNIT_CODED_SLICE_SCALABLE_DATAPART_B)
                {
                        // Everything is fine
                        m_dpPresent[DATA_PART_B] = true;
                        return DP_OK;
                }
                else if(currNalu == NAL_UNIT_CODED_SLICE_SCALABLE_DATAPART_C)
                {
                        // DP B is missing. But decoding should continue.
                        m_dpPresent[DATA_PART_B] = false;
                        m_dpPresent[DATA_PART_C] = true;
                        return DP_CONTINUE;
                }
                else
                {
                        // DP B & C are missing. But the decoding should continue
                        // the bitstream should be rewind to put the current NALU
                        // back into the stream for reading again.
                        m_dpPresent[DATA_PART_B] = false;
                        m_dpPresent[DATA_PART_C] = false;
                        eExpectedNalu = NAL_UNIT_CODED_SLICE_SCALABLE_DATAPART_C;
                        return DP_REWIND_AND_CONTINUE;
                }
        }
        else if(prevNalu == NAL_UNIT_CODED_SLICE_SCALABLE_DATAPART_B)
        {
                if(currNalu == NAL_UNIT_CODED_SLICE_SCALABLE_DATAPART_C)
                {
                        // Everything is fine.
                        m_dpPresent[DATA_PART_C] = true;
                        return DP_OK;
                }
                else
                {
                        // DP C is missing. But the decoding should continue
                        // the bitstream should be rewind to put the current NALU
                        // back into the stream for reading again.
                        m_dpPresent[DATA_PART_C] = false;
                        eExpectedNalu = NAL_UNIT_CODED_SLICE_SCALABLE_DATAPART_C;
                        return DP_REWIND_AND_CONTINUE;
                }
        }
        else if( (currNalu == NAL_UNIT_CODED_SLICE_SCALABLE_DATAPART_B || currNalu ==
NAL_UNIT_CODED_SLICE_SCALABLE_DATAPART_C) &&
                         m_dpPresent[DATA_PART_A] == false )
        {
                // DP A is missing
                m_dpPresent[DATA_PART_A] = false;
                m_dpPresent[NalUnitType2DataPart[currNalu]] = true;
                return DP_MISSING_A;
        }
        else if( currNalu == NAL_UNIT_CODED_SLICE_SCALABLE_DATAPART_A )
        {
                m_dpPresent[DATA_PART_A] = true;
                return DP_OK;
        }

        m_dpPresent[NalUnitType2DataPart[currNalu]] = true;

        return DP_OK;
}
```

Frame copy Error Concealment

```
{
                m_OpViewId[uiOp] = NULL;
                m_uiNumViews[uiOp] = 0;
        }
        //SEI }
        //~JVT-P031
        //      TMM     EC {{
        m_uiNextFrameNum     =      0;
        m_uiNextLayerId             =      0;
        m_uiNextPoc                        =      0;
        m_uiNumLayers               =      1;
        m_uiMaxGopSize              =      16;
        m_uiMaxDecompositionStages  =      4;
        m_uiMaxLayerId       =      0;
        UInt ui;
        for ( ui=0; ui<MAX_LAYERS; ui++)
        {
                m_pauiPocInGOP         [ui]  =     NULL;
                m_pauiFrameNumInGOP    [ui]  =     NULL;
                m_pauiTempLevelInGOP   [ui]  =     NULL;
                m_uiDecompositionStages[ui]  =     4;
                m_uiFrameIdx           [ui]  =     0;
                m_uiGopSize            [ui]         =     16;
        }

        m_eErrorConceal   =   EC_NONE;
        m_bNotSupport =      false;

        if(m_eErrorConceal==EC_RECONSTRUCTION_UPSAMPLE)
                m_eErrorConceal=EC_FRAME_COPY;
        //  TMM_EC }}
}
```

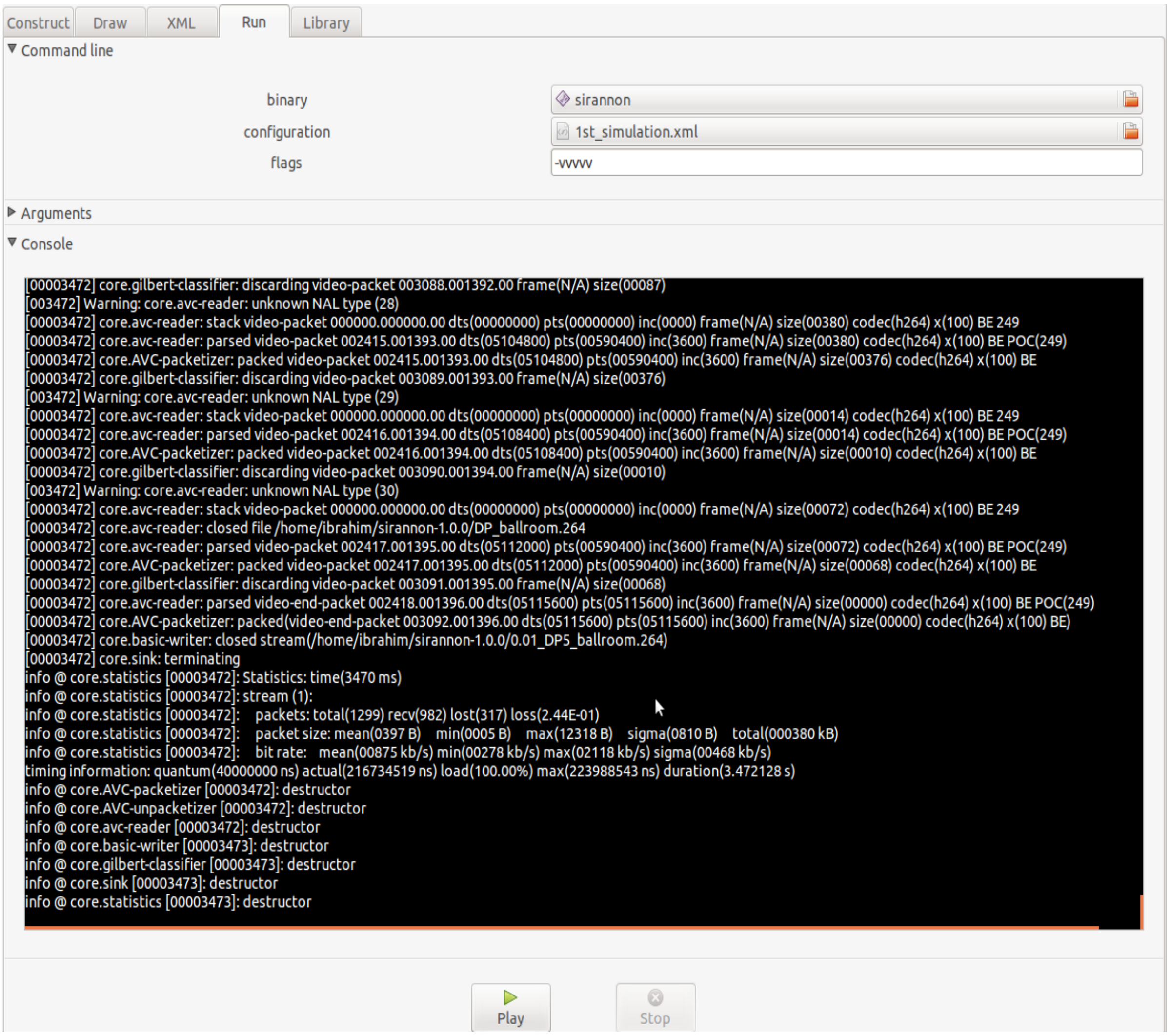
Error Concealment for non-key pictures

```
        //***** NOTE: Motion-compensated prediction for non-key pictures is done in
xReconstructLastFGS()
        bReconstruct   = (bReconstruct && bKeyPicture) || ! bConstrainedIP;

        RNOK( m_pcControlMng  ->initSlice ( rcSH, DECODE_PROCESS ) );

        if ( m_eErrorConceal == EC_BLSKIP || m_eErrorConceal == EC_TEMPORAL_DIRECT)

        {
                RNOK( m_pcSliceDecoder->processVirtual( rcSH, bReconstruct, uiMbRead )
);
```

```
       }
       else
       {
              Frame  *frame =      (Frame*)(rcSH.getRefPicList(       rcSH.getPicType()
,LIST_0 ).get(0).getFrame());
              m_pcFrameMng->getCurrentFrameUnit()->getFrame().getFullPelYuvBuffer()-
>loadBuffer( frame->getFullPelYuvBuffer());
       }

       Bool bPicDone;
       RNOK( m_pcControlMng->finishSlice( rcSH, bPicDone, m_bFrameDone ) );

       bPicDone      =      true;
       m_bFrameDone  =      true;

       if (IsSliceEndOfPic())
       {
              if ( m_eErrorConceal == EC_RECONSTRUCTION_UPSAMPLE || m_eErrorConceal ==
EC_FRAME_COPY)
              {
                     //                   rcSH.getFrameUnit()->getFGSIntFrame()->copy(
&m_pcFrameMng->getCurrentFrameUnit()->getFrame());
                     // memory
              }
              // copy intra and inter prediction signal
              //         m_pcFrameMng->getPredictionIntFrame()->getFullPelYuvBuffer()-
>copy( rcSH.getFrameUnit()->getFGSIntFrame()->getFullPelYuvBuffer());
              // memory
              // delete intra prediction
```

Appendix – B3

Sirannon Network Simulator

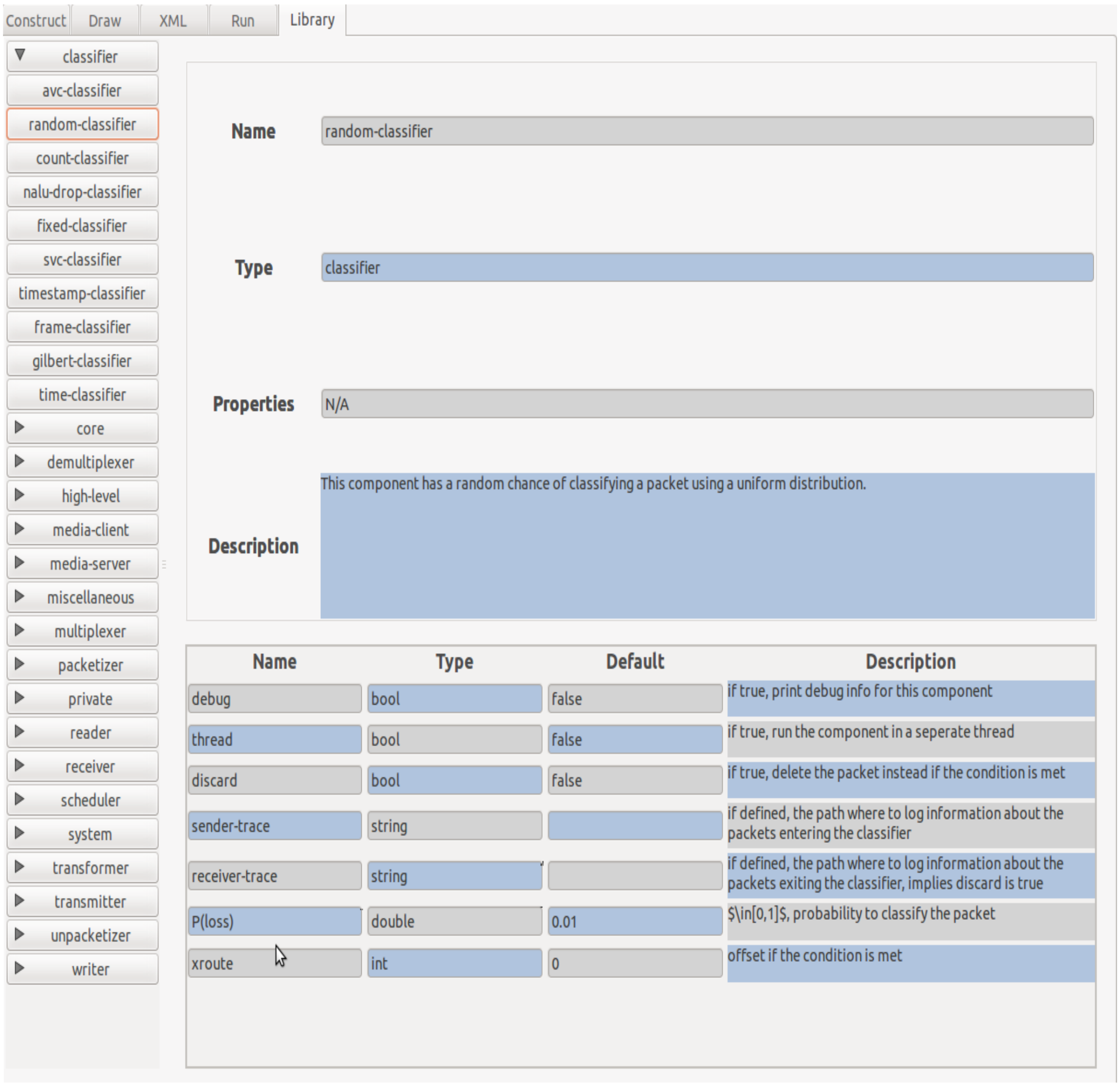


binary sirannon
configuration 1st_simulation.xml
flags -vvvv

```
[00003472] core.gilbert-classifier: discarding video-packet 003088.001392.00 frame(N/A) size(00087)
[003472] Warning: core.avc-reader: unknown NAL type (28)
[00003472] core.avc-reader: stack video-packet 000000.000000.00 dts(00000000) pts(00000000) inc(0000) frame(N/A) size(00380) codec(h264) x(100) BE 249
[00003472] core.avc-reader: parsed video-packet 002415.001393.00 dts(05104800) pts(00590400) inc(3600) frame(N/A) size(00380) codec(h264) x(100) BE POC(249)
[00003472] core.AVC-packetizer: packed video-packet 002415.001393.00 dts(05104800) pts(00590400) inc(3600) frame(N/A) size(00376) codec(h264) x(100) BE
[00003472] core.gilbert-classifier: discarding video-packet 003089.001393.00 frame(N/A) size(00376)
[003472] Warning: core.avc-reader: unknown NAL type (29)
[00003472] core.avc-reader: stack video-packet 000000.000000.00 dts(00000000) pts(00000000) inc(0000) frame(N/A) size(00014) codec(h264) x(100) BE 249
[00003472] core.avc-reader: parsed video-packet 002416.001394.00 dts(05108400) pts(00590400) inc(3600) frame(N/A) size(00014) codec(h264) x(100) BE POC(249)
[00003472] core.AVC-packetizer: packed video-packet 002416.001394.00 dts(05108400) pts(00590400) inc(3600) frame(N/A) size(00010) codec(h264) x(100) BE
[00003472] core.gilbert-classifier: discarding video-packet 003090.001394.00 frame(N/A) size(00010)
[003472] Warning: core.avc-reader: unknown NAL type (30)
[00003472] core.avc-reader: stack video-packet 000000.000000.00 dts(00000000) pts(00000000) inc(0000) frame(N/A) size(00072) codec(h264) x(100) BE 249
[00003472] core.avc-reader: closed file /home/ibrahim/sirannon-1.0.0/DP_ballroom.264
[00003472] core.avc-reader: parsed video-packet 002417.001395.00 dts(05112000) pts(00590400) inc(3600) frame(N/A) size(00072) codec(h264) x(100) BE POC(249)
[00003472] core.AVC-packetizer: packed video-packet 002417.001395.00 dts(05112000) pts(00590400) inc(3600) frame(N/A) size(00068) codec(h264) x(100) BE
[00003472] core.gilbert-classifier: discarding video-packet 003091.001395.00 frame(N/A) size(00068)
[00003472] core.avc-reader: parsed video-end-packet 002418.001396.00 dts(05115600) pts(05115600) inc(3600) frame(N/A) size(00000) codec(h264) x(100) BE POC(249)
[00003472] core.AVC-packetizer: packed(video-end-packet 003092.001396.00 dts(05115600) pts(05115600) inc(3600) frame(N/A) size(00000) codec(h264) x(100) BE)
[00003472] core.basic-writer: closed stream(/home/ibrahim/sirannon-1.0.0/0.01_DP5_ballroom.264)
[00003472] core.sink: terminating
info @ core.statistics [00003472]: Statistics: time(3470 ms)
info @ core.statistics [00003472]: stream (1):
info @ core.statistics [00003472]:    packets: total(1299) recv(982) lost(317) loss(2.44E-01)
info @ core.statistics [00003472]:    packet size: mean(0397 B)    min(0005 B)    max(12318 B)    sigma(0810 B)    total(000380 kB)
info @ core.statistics [00003472]:    bit rate:    mean(00875 kb/s) min(00278 kb/s) max(02118 kb/s) sigma(00468 kb/s)
timing information: quantum(40000000 ns) actual(216734519 ns) load(100.00%) max(223988543 ns) duration(3.472128 s)
info @ core.AVC-packetizer [00003472]: destructor
info @ core.AVC-unpacketizer [00003472]: destructor
info @ core.avc-reader [00003472]: destructor
info @ core.basic-writer [00003473]: destructor
info @ core.gilbert-classifier [00003473]: destructor
info @ core.sink [00003473]: destructor
info @ core.statistics [00003473]: destructor
```

Figure – B3 1

Figure – B3 1 is a snapshot of the Sirannon network simulator console window that shows details of the output simulation process, which includes statistics of the total number of packets transmitted, total number of packets received and the total number of lost packets. Also from the console window, other useful information such as the transmission time, minimum and maximum bitrate, delays and the size of each packet in the sequence.



Name: random-classifier

Type: classifier

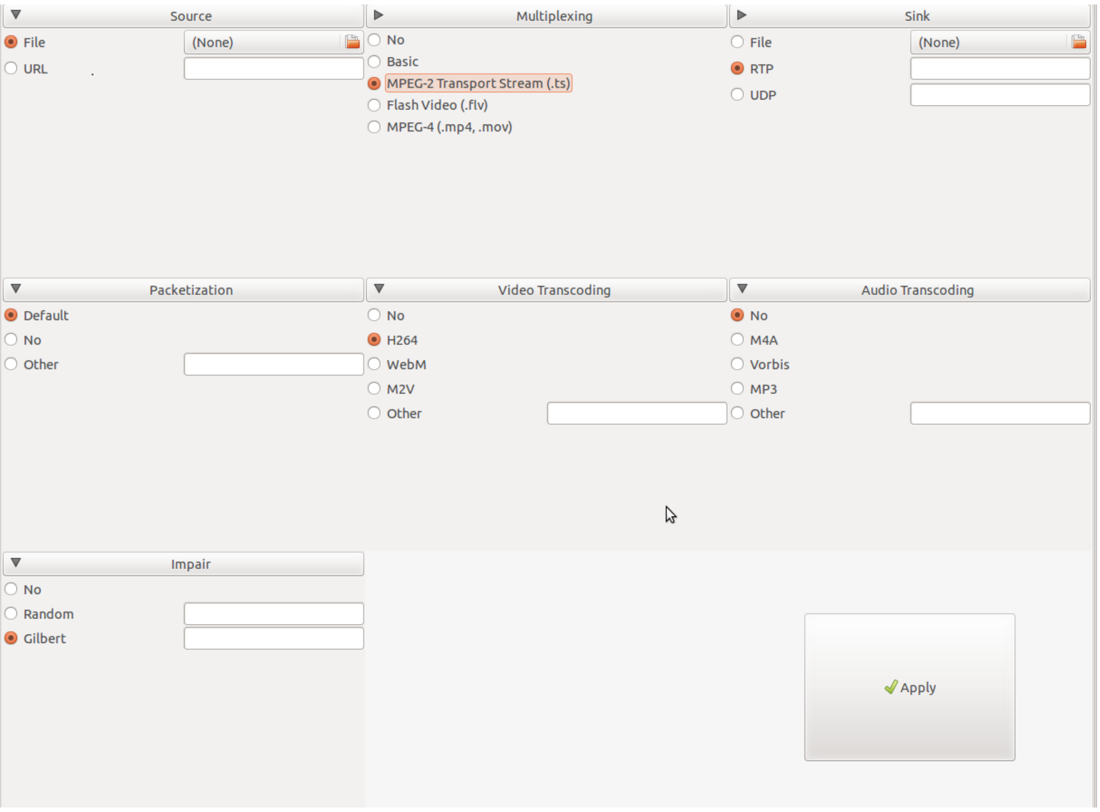
Properties: N/A

Description: This component has a random chance of classifying a packet using a uniform distribution.

Name	Type	Default	Description
debug	bool	false	if true, print debug info for this component
thread	bool	false	if true, run the component in a seperate thread
discard	bool	false	if true, delete the packet instead if the condition is met
sender-trace	string		if defined, the path where to log information about the packets entering the classifier
receiver-trace	string		if defined, the path where to log information about the packets exiting the classifier, implies discard is true
P(loss)	double	0.01	$\in[0,1]$, probability to classify the packet
xroute	int	0	offset if the condition is met

Figure – B3 2

Figure – B3 2 shows the Sirannon network simulation library and specifically an outline of the random error classifier. Details of all the components such as name, type properties and description is provided and embedded within the tool’s library for reference purpose.



Source
● File (None)
○ URL

Multiplexing
○ No
○ Basic
● MPEG-2 Transport Stream (.ts)
○ Flash Video (.flv)
○ MPEG-4 (.mp4, .mov)

Sink
○ File (None)
● RTP
○ UDP

Packetization
● Default
○ No
○ Other

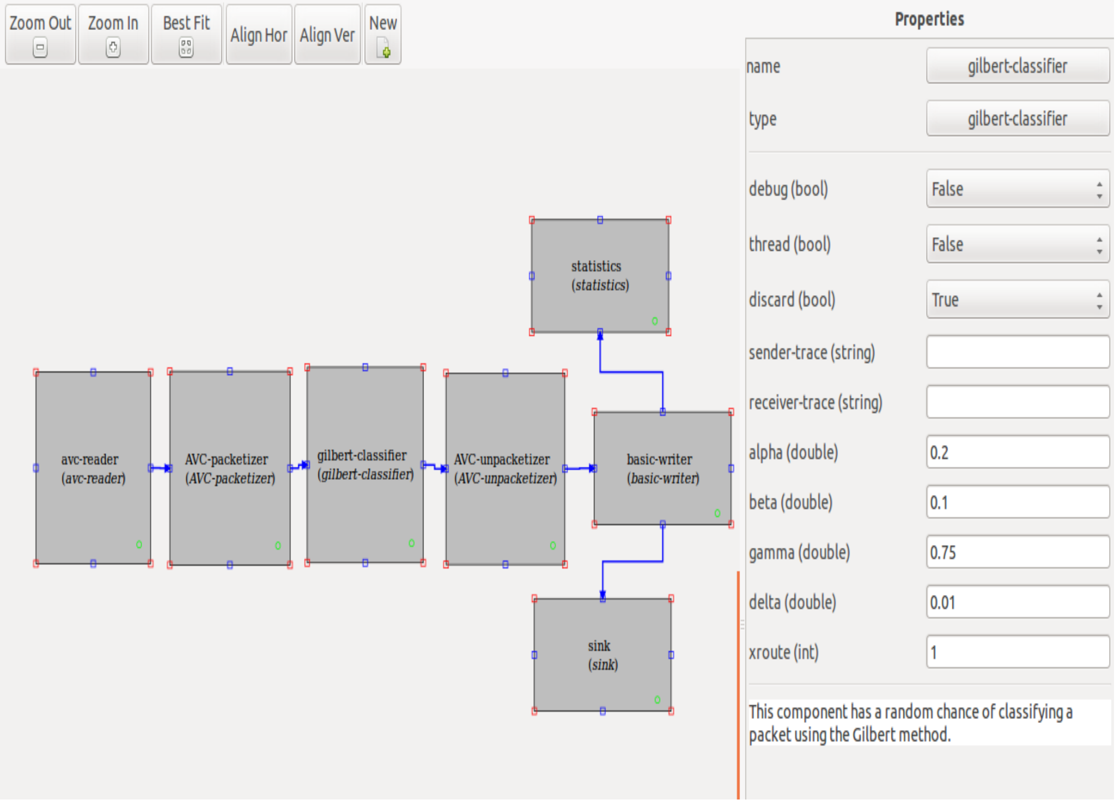
Video Transcoding
○ No
● H264
○ WebM
○ M2V
○ Other

Audio Transcoding
● No
○ M4A
○ Vorbis
○ MP3
○ Other

Impair
○ No
○ Random
● Gilbert

Figure – B3 3

Figure – B3 3 shows how the network is constructed and configured for the test bed used in the network simulator.



Zoom Out
Zoom In
Best Fit
Align Hor
Align Ver
New
statistics (statistics)
avc-reader (avc-reader)
AVC-packetizer (AVC-packetizer)
gilbert-classifier (gilbert-classifier)
AVC-unpacketizer (AVC-unpacketizer)
basic-writer (basic-writer)
sink (sink)

Properties

name	gilbert-classifier
type	gilbert-classifier
debug (bool)	False
thread (bool)	False
discard (bool)	True
sender-trace (string)	
receiver-trace (string)	
alpha (double)	0.2
beta (double)	0.1
gamma (double)	0.75
delta (double)	0.01
xroute (int)	1

This component has a random chance of classifying a packet using the Gilbert method.

Figure – B3 4

Figure – B3 4 shows the schematic test bed used in the thesis. The figure also shows some of the settings used in order to achieve a desired packet loss rate in offline mode. The Gilbert classifier as can be seen from figure - A3 4 is used to generate the error pattern based on the default parameter settings described in the software.

- double alpha: ϵ [0; 1], probability to transit from the GOOD to the BAD state, default: 0.01
- double beta: ϵ [0; 1], probability to transit from the BAD to the GOOD state, default: 0.1
- double gamma: ϵ [0; 1], probability to classify a packet in the BAD state, default: 0.75
- double delta: ϵ [0; 1], probability to classify a packet in the GOOD state, default: 0.01
- int xroute: offset if the condition is met, default: 1

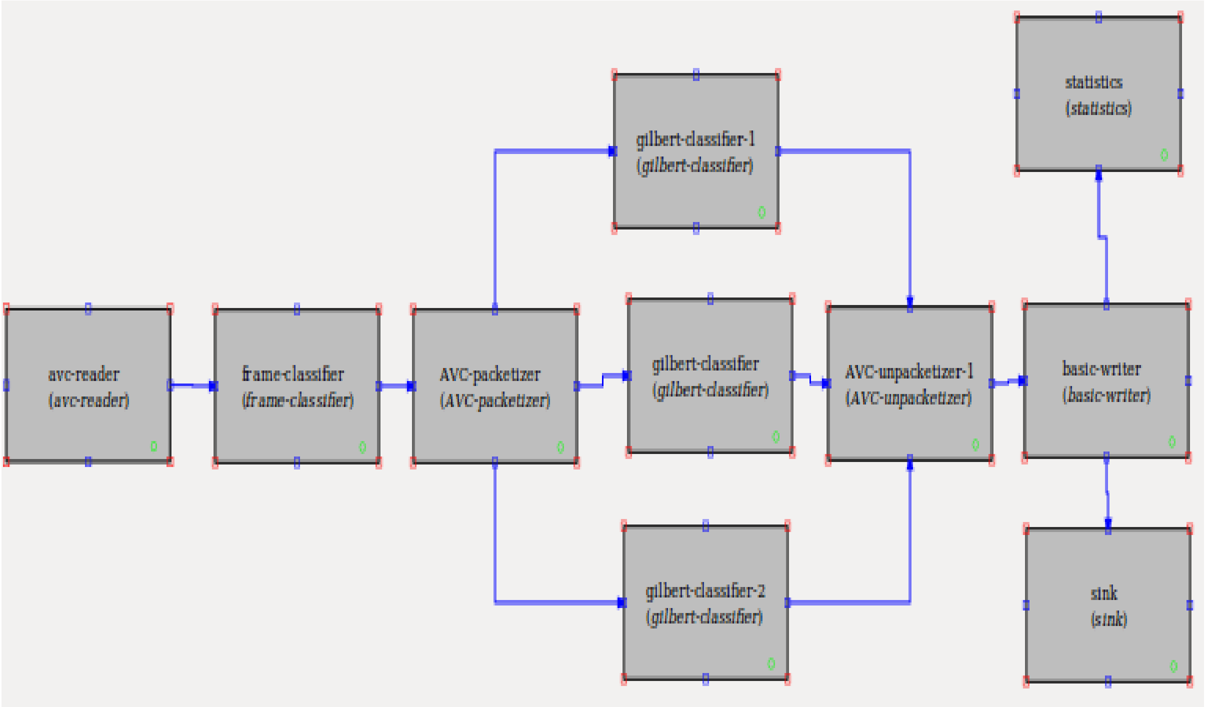


Figure – B3 5

Figure – B3 5 illustrates another test bed that can be used to generate or introduce packet loss. In this test bed, the bitstream is split into I, P, and B packets so that each bitstream can have a specific packet loss rate. Eventually, the damaged bitstream are merged together and the corrupted sequence is written in the writer component.