

ENCODER COMPLEXITY REDUCTION  
WITH SELECTIVE MOTION MERGE IN HEVC

by

ABHISHEK HASSAN THUNGARAJ

Presented to the Faculty of the Graduate School of  
The University of Texas at Arlington in Partial Fulfillment  
of the Requirements  
for the Degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

August 2014

Copyright © by Abhishek Hassan Thungaraj 2014

All Rights Reserved

## Acknowledgements

I would like to thank Dr. K. R. Rao for being a supervisor, mentor and a source of inspiration encouraging me continuously during the course of my thesis. I would like to thank Dr. W. Dillon and Dr. K. Alavi for serving on my thesis committee.

I would also like to thank my MPL lab-mates: Karun Gubbi, Kushal Shah and Tuan Ho for providing valuable insights throughout my research.

Last but not least, I would like to thank my family and friends for supporting me in every way in this undertaking.

July 16, 2014

## Abstract

### ENCODER COMPLEXITY REDUCTION WITH SELECTIVE MOTION MERGE IN HEVC

Abhishek Hassan Thungaraj

The University of Texas at Arlington, 2014

Supervising Professor: K. R. Rao

The High Efficiency Video Coding (HEVC) standard is the latest video coding project developed by the Joint Collaborative Team on Video Coding (JCT-VC) which involves the International Telecommunication Unit (ITU-T) Video Coding Experts Group (VCEG) and the ISO/IEC Moving Pictures Experts Group (MPEG) standardization organizations. The HEVC standard is based on the previous widely used H.264/AVC (Advance Video Coding) standard [10] but includes many new tools and improvements in different key areas which has resulted in achieving a 50% bitrate reduction compared to its predecessor amidst maintaining the same visual quality [11] at a cost of increased complexity.

Among the different vital blocks of a video codec the motion estimation and motion compensation blocks are considered as the key and the most complex section. The calculations involving the derivation of motion estimation followed by predictor picture derivation leading to a residual image which is in turn followed by motion compensation using the previously encoded motion information is responsible for consuming a large part of the encoding time and device resource. Since the video signal largely consists of real world objects which have regions of homogeneous motion, the encoder tries to make use of these regions of common motion to bring about a reduction in bitrate. To achieve this reduction the encoder adapts the technique of motion merging which exploit the redundancies among the motion information data obtained through motion estimation. Although the present algorithm used in HEVC is capable of making use of

these redundancies they still impose a large computational and encoding time overhead on the codecs affecting the device performance. The thesis proposes an algorithm which selectively performs the motion merging allowing the formation of larger blocks of homogenous motion which reduces the bitrate and also utilizes the typical characteristics of video signal to reduce the encoding time at a cost of little loss of quality. The experimental results based on the proposed algorithm on several test sequences suggest a reduction in encoding time by 14-24%, reduction in bitrate by 2-7% at a little loss of quality by 2-6%. Metrics such as BD-PSNR (Bjontegaard Delta Peak Signal to Noise Ratio), BD-bitrate (Bjontegaard Delta bitrate) are used.

## Table of Contents

Acknowledgements.....	iii
Abstract.....	iv
List of Figures.....	ix
List of Tables.....	xii
Chapter 1 Introduction.....	1
1.1 Basics of Video Compression and its Need.....	1
1.2 Video Compression Standards.....	5
1.3 Outline of the Thesis.....	6
Chapter 2 High Efficiency Video Coding, HEVC.....	7
2.1 Video Coding Layer.....	9
2.2 Features of HEVC.....	10
2.2.1 Coding Tree Unit (CTU) and Coding Tree Block (CTB).....	10
2.2.2 Coding Unit (CU) and Coding Block (CB).....	10
2.2.3 Prediction Unit (PU) and Prediction Block (PB).....	11
2.2.4 Transform Units (TU) and Transform Blocks (TB).....	11
2.2.5 Slices and Tiles.....	12
2.2.6 Loop Filters.....	13
2.2.7 Intrapicture Prediction.....	14
2.3 Scalable Video Coding.....	14

2.3.1 Inter-Layer Prediction .....	15
2.4 Summary .....	16
Chapter 3 Interpicture Prediction .....	17
3.1 Motion Vector Prediction .....	17
3.2 Merge Mode in HEVC.....	21
3.2.1 Spatial Merge Candidates .....	23
3.2.2 Temporal Merge Candidates.....	24
3.3 Proposed Method .....	26
3.4 Summary .....	27
Chapter 4 Results .....	28
4.1 Test Conditions .....	28
4.2 Reduction in Encoding Time .....	29
4.3 Reduction in Bitrate .....	32
4.4 Reduction in PSNR.....	35
4.5 BD-PSNR and BD-Bitrate .....	38
4.6 Rate Distortion Plot (RD Plot).....	43
4.7 Summary .....	46
Chapter 5 Conclusions and Future Work.....	47
5.1 Conclusions.....	47
5.2 Future Work .....	48

Appendix A Test Sequences [46] .....	49
A.1 Race Horses.....	50
A.2 BQ Mall.....	51
A.3 Basketball Drill Text.....	52
A.4. Kristen and Sara .....	53
A.5 Basketball Drive.....	54
Appendix B Test Conditions.....	55
Appendix C BD-PSNR and BD-Bitrate [40][41] .....	57
Appendix D Acronyms .....	61
References.....	64
Biographical Information.....	69



## List of Figures

Figure 1.1 Typical order of the Intra and Inter frames [5].....	2
Figure 1.2 4:2:0 sampling [5].....	3
Figure 1.3 4:2:2 sampling [5].....	3
Figure 1.4 4:4:4 sampling [5].....	4
Figure 1.5 Evolution of video compression standards [8] .....	6
Figure 2.1 Typical HEVC video encoder along with decoder modelling [11].....	8
Figure 2.2 Decoder block of the HEVC [15].....	8
Figure 2.3 Modes of splitting a CB into PBs [11] .....	11
Figure 2.4 Division of CTB into CBs (solid line) and TBs (dotted line) [11].....	12
Figure 2.5 Quadtree corresponding to the figure 2.3 [11] .....	12
Figure 2.6 Modes and their directional orientation for intrapicture prediction [11].....	14
Figure 2.7 Scalable encoder with two layers [25].....	15
Figure 3.1 Partition modes in HEVC [10] .....	18
Figure 3.2 Multiple reference pictures for a single current picture [10].....	19
Figure 3.3 Integer and fractional sample luma interpolation in HEVC [11] .....	21
Figure 3.4 Process of obtaining the merge candidates in HEVC [35].....	22
Figure 3.5 Positions of spatial merge candidates [35] .....	23
Figure 3.6 Positions for second PU of Nx2N and 2NxN partitions [35] .....	24
Figure 3.7 Motion vector scaling for temporal merge candidate [35] .....	25
Figure 3.8 Positions of the candidates for temporal merge [35].....	25
Figure 4.1 GOP used in 'random access profile' of HEVC .....	28
Figure 4.2 Encoding time vs quantization parameter for 'Race Horse' sequence.....	29

Figure 4.3 Encoding time vs quantization parameter for 'BQ Mall' sequence .....	30
Figure 4.4 Encoding time vs quantization parameter for 'Basketball Drill Text' sequence.....	30
Figure 4.5 Encoding time vs quantization parameter for 'Kristen and Sara' sequence.....	31
Figure 4.6 Encoding time vs quantization parameter for 'Basketball Drive' sequence.....	31
Figure 4.7 Bitrate vs quantization parameter for 'Race Horse' sequence.....	32
Figure 4.8 Bitrate vs quantization parameter for 'BQ Mall' sequence .....	33
Figure 4.9 Bitrate vs quantization parameter for 'Basketball Drill Text' sequence .....	33
Figure 4.10 Bitrate vs quantization parameter for 'Kristen and Sara' sequence .....	34
Figure 4.11 Bitrate vs quantization parameter for 'Basketball Drive' sequence .....	34
Figure 4.12 PSNR vs quantization parameter for 'Race Horse' sequence .....	35
Figure 4.13 PSNR vs quantization parameter for 'BQ Mall' sequence.....	36
Figure 4.14 PSNR vs quantization parameter for 'Basketball Drill Text' sequence .....	36
Figure 4.15 PSNR vs quantization parameter for 'Kristen and Sara' sequence .....	37
Figure 4.16 PSNR vs quantization parameter for 'Basketball Drive' sequence .....	37
Figure 4.17 BD-PSNR vs quantization parameter for 'Race Horse' sequence.....	38
Figure 4.18 BD-PSNR vs quantization parameter for 'BQ Mall' sequence .....	39
Figure 4.19 BD-PSNR vs quantization parameter for 'Basketball Drill Text' sequence .....	39
Figure 4.20 BD-PSNR vs quantization parameter for 'Kristen and Sara' sequence .....	40
Figure 4.21 BD-PSNR vs quantization parameter for 'Basketball Drive' sequence .....	40
Figure 4.22 BD-bitrate vs quantization parameter for 'Race Horse' sequence .....	41
Figure 4.23 BD-bitrate vs quantization parameter for 'BQ Mall' sequence.....	41
Figure 4.24 BD-bitrate vs quantization parameter for 'Basketball Drill Text' sequence .....	42
Figure 4.25 BD-bitrate vs quantization parameter for 'Kristen and Sara' sequence .....	42

Figure 4.26 BD-bitrate vs quantization parameter for 'Basketball Drive' sequence .....	43
Figure 4.27 PSNR vs bitrate for 'Race Horse' sequence .....	44
Figure 4.28 PSNR vs bitrate for 'BQ Mall' sequence .....	44
Figure 4.29 PSNR vs bitrate for 'Basketball Drill Text' sequence .....	45
Figure 4.30 PSNR vs bitrate for 'Kristen and Sara' sequence .....	45
Figure 4.31 PSNR vs bitrate for 'Basketball Drive' sequence.....	46

## List of Tables

Table 4.1 List of test sequences [46] .....	28
---	----

## Chapter 1

### Introduction

With the invention of smart phones and internet TV technologies the importance of digital video compression and transmission has reached a new level. The new age cellphones and TVs are no longer just capable of performing their basic intended tasks but are developed with the advanced abilities to perform tasks like video conferencing, web browsing, storage of live telecasts for later viewing, navigation applications etc. With all these recent developments we see that video is unquestionably an integral part of development due to its ability to appeal to more audiences. The outcome of this has led to real time multimedia transmission for video conferencing applications and live telecasts and also improvement in the quality of digital video leading to High Definition (HD), Ultra HD, 4K and also 8K resolution video formats and the High Dynamic Range (HDR) videos [1] which provides superior and realistic viewing experience to the users.

#### 1.1 Basics of Video Compression and its Need

The fundamental aspect of any video lies in the images and image as we understand is a projection of a 3 dimensional scene containing depth, texture and illumination onto a 2 dimensional plane consisting of just texture and illumination [2]. Digital form of this image is the representation of the image as a collection of pixels. Digital video on the other hand is collection of these digital images which provide the illusion of motion when played in quick succession [3].

The digital video is generally uncompressed or termed as raw video. The problem with such uncompressed video is that it requires large space for its storage and also large bandwidth for its transmission. The solution for this problem is compression which can be either lossless or

lossy. Compression of an image makes use of the spatial redundancies that exists within an image and has the liberty of undergoing lossless or lossy compression; on a similar line the compression of a video makes use of temporal redundancies which exists among sequence of images which form a continuous scene but incur unavoidable losses in the process.

Effectively video compression exploits both temporal and spatial redundancies. A frame which is compressed by exploiting the spatial redundancies is termed as intra frame and the frames which are compressed by exploiting the temporal redundancies are termed as inter frames. The compression of a inter frame requires a reference frame which will be used to exploit the temporal redundancies. In addition to this the inter frame is of two types namely a P - frame and B – frames. The P - frame makes use of one already encoded/decoded frame which may appear before or after the current picture in the display order i.e. a past or a future frame as its reference, whereas the B - frames make use of two already encoded/decoded frames one of which is a past and the other being the future frame as its reference frames thus providing higher compression but also higher encoding time as it has to use a future frame for encoding [4].

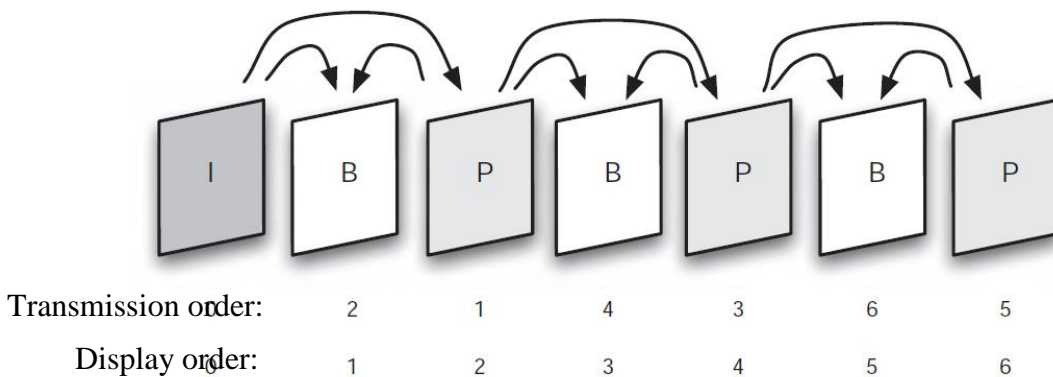


Figure 1.1 Typical order of the Intra and Inter frames [5]

The classical order of frames called as the Group of Pictures or the GOP is given in Figure 1.1 [5]. We observe that the very first frame is always encoded using intra frame encoding

indicating by the letter I, this is because the first frame does not have any frame as a reference. The I frame is followed by a P – frame, as it has the ability to make use of just one reference frame. After the P - frame the B - frame is encoded which makes use of both the I and the P - frames as its references. This pattern is followed with the subsequent frames where the P - frame takes the position of the I - frame.

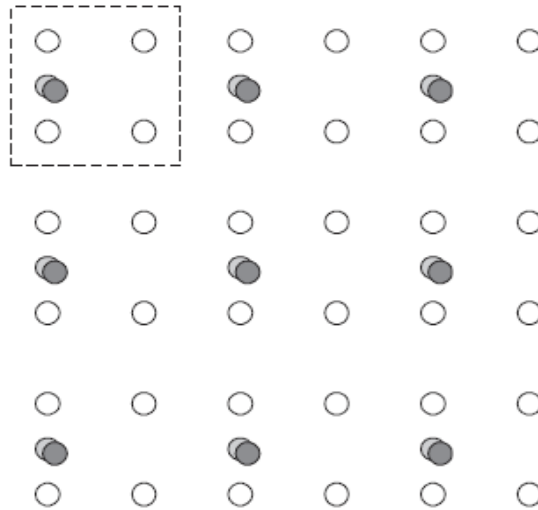


Figure 1.2 4:2:0 sampling [5]

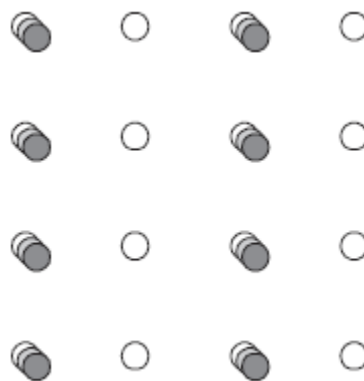


Figure 1.3 4:2:2 sampling [5]

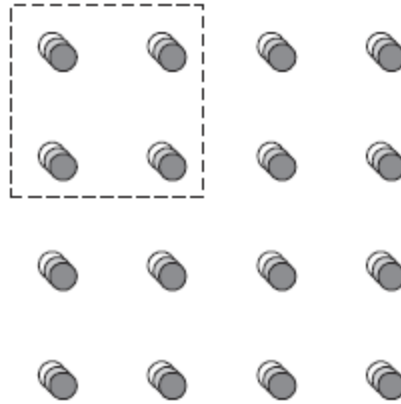


Figure 1.4 4:4:4 sampling [5]

Sampling formats: Although the reduction in size of the video is mainly as a result of exploiting the spatial and temporal redundancies the fundamental bit reduction is obtained by the sampling format enforced [6]. Typically YCrCb format is followed for representing the color space as it exploits the fact that the human visual system (HVS) is less sensitive to color than to luminance [7]. This YCrCb is very similar to the RGB format but represents the luminance also.

$$Y = \text{mean luminance}$$

$$Cr = R - Y$$

$$Cb = B - Y$$

$$Cg = G - Y$$

where, R, G, B indicate the Red, Green and Blue and Cr, Cb and Cg represent the difference between the color intensity and the mean luminance of each image sample. During encoding only the Y, Cr and the Cb will be used to represent the pixels as the Cg can be obtained by simple calculation of the rest thus avoiding a transmission overhead. The typical YCrCb sampling formats are 4:2:0, 4:2:2 and 4:4:4 where the numbers indicate the relative sampling rate of each component in the horizontal direction and the vertical directions, this is as depicted in figures 1.2, 1.3 and 1.4 [5].



The benefit of this sampling format can be observed using an example. In case of a Full HD image whose resolution is 1980 x 1080, if the bit depth of each channel is 8 bits then,

4:4:4 Cr, Cb resolution provides:  $1980 \times 1080 \times 8 \times 3 = 51321600$  bits.

4:2:2 Cr, Cb resolution provides:  $990 \times 1080 \times 8 \times 3 = 25660800$  bits, which is 50% of 4:4:4.

4:2:0 Cr, Cb resolution provides:  $990 \times 540 \times 8 \times 3 = 12830400$  bits, which is 25% of 4:4:4.

From the above example indicates that the sampling format provides significant bit reduction with no perceivable loss of information as the HVS is more sensitive to the intensity i.e. luminance factor rather than the chrominance factor [7].

## 1.2 Video Compression Standards

Since the compression of video is very important for its storage and transmission, it has led to the development of encoders and decoders by different vendors causing the issues with compatibility while they have to function together. To counter this issue the video compression algorithms have been standardized by international bodies namely ISO/IEC (International Organization for Standardization/ International Electrochemical Commission) , Moving Picture Experts Groups (MPEG), International Telecommunication Union-Telecommunication Standardization Sector (ITU-T) and Joint Collaborative Team on Video Coding (JCT-VC). These regulatory bodies have led to the development of many standards with improvements over the predecessors shown in Figure 1.5 [8].

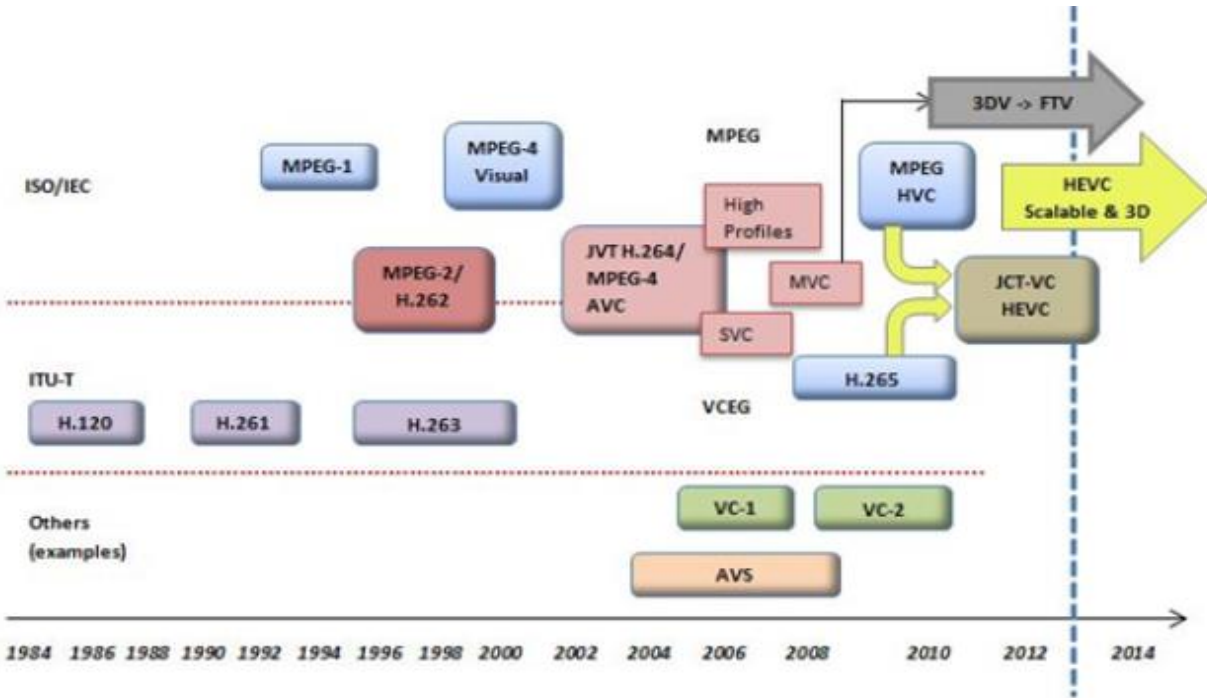


Figure 1.5 Evolution of video compression standards [8]

### 1.3 Outline of the Thesis

The following chapter 2 provides an introduction to the compression algorithms and detailed description of the HEVC standard. Chapter 3 discusses the motion estimation, motion compensation along with the motion merging techniques currently used in the HEVC standard and the motivation for improvement and the proposed algorithm for the same. The chapter 4 discusses the results based on the implementation of the proposed algorithm and provides a comparison against the present algorithm used. Chapter 5 draws a conclusion based on the obtained results and suggests areas of future work.

## Chapter 2

### High Efficiency Video Coding, HEVC

High Efficiency Video Coding is the most recently introduced video compression standard developed by the Joint Collaborative team on Video Coding (JCT-VC) in January, 2013 [9][10]. The standard was developed over a period of 6 years from 2007 to 2013 with a goal to provide lower bitrate than the H.264 standard while still retaining the visual quality [11]. The HEVC standard is composed of three profiles which are a) ‘main profile’ which is capable of handling 8-bit input data. b) ‘main 10 profile’ for handling 10-bit input data and c) ‘Still frame’.

The HEVC standard was developed to address the increased diversity of services like the HD video, beyond HD formats such as 4k X 2k or 8k x 4k resolutions which impose a strong challenge on the present networks [12]. The HEVC standard has been designed to address almost all the applications that existed with the H.264 standard but with special focus on increased video resolution and increased use of parallel processing architectures. It is designed to tackle multiple goals which include increasing the coding efficiency, ease of transport system integration and data loss resilience and also implementability using parallel processing architectures. The major advancements of HEVC over the H.264 standard is the provision for flexible transform block sizes, flexible prediction modes, improved interpolation filters, incorporation of sample and adaptive offset filters for further reducing blocking artifacts and most importantly the ability to exploit parallel processing architectures provided by the Graphics Processing Units (GPUs) [13]. The HEVC extension [14] also include the support for extended formats with much higher bit depths, scalable video coding and also 3D, stereo and multi vision encoding. The basic description of the HEVC encoder is shown in the Figure 2.1 [11].

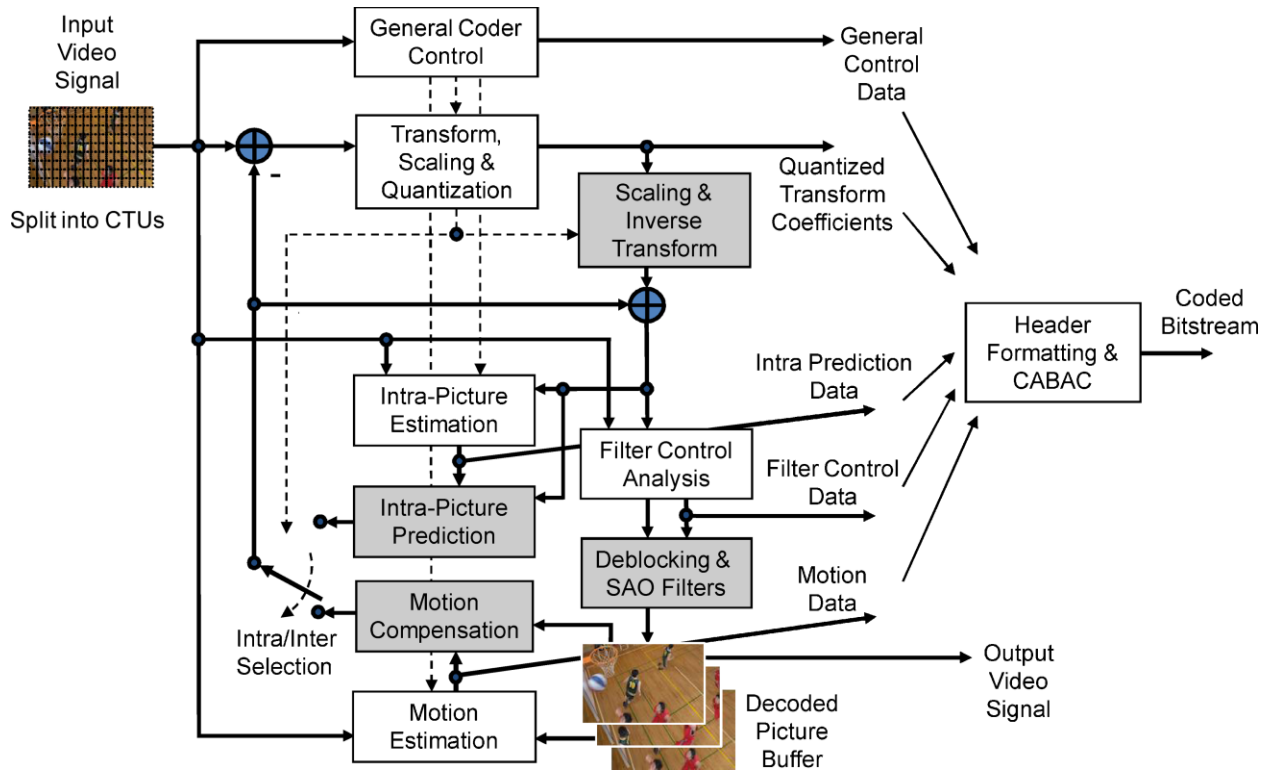


Figure 2.1 Typical HEVC video encoder along with decoder modelling [11]

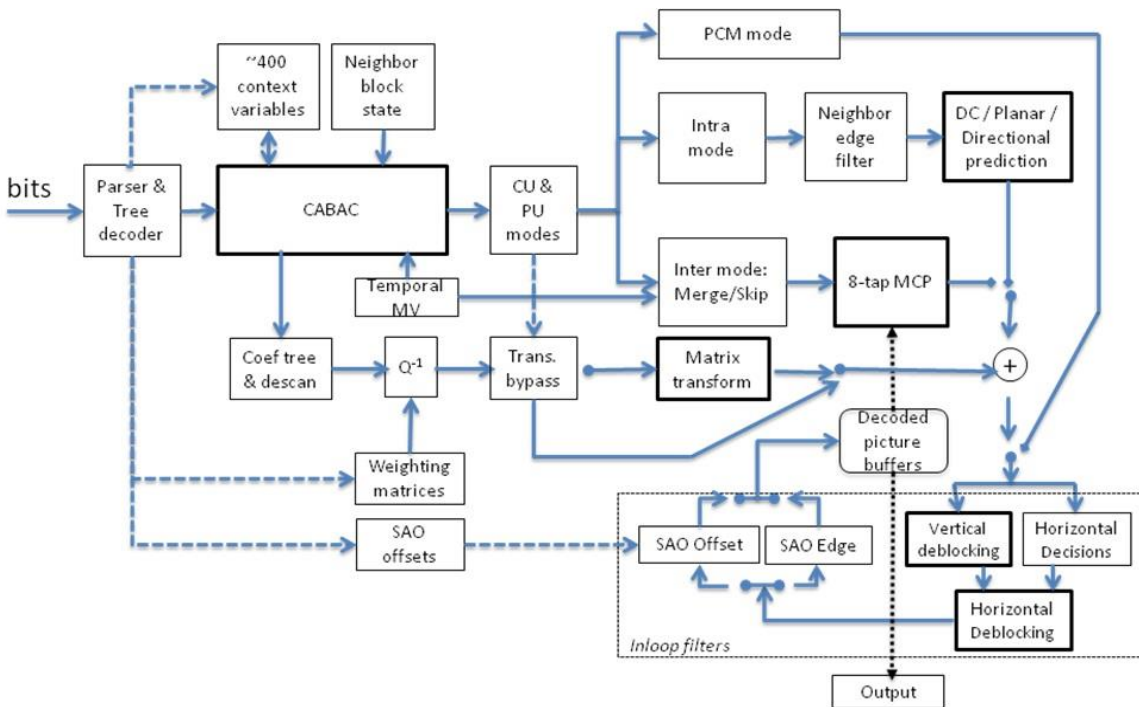


Figure 2.2 Decoder block of the HEVC [15]

The HEVC is composed of many newly incorporated features which aim to support error data loss resilience and parallel processing architectures. It comprises of the following.

## 2.1 Video Coding Layer

HEVC uses the hybrid approach (inter/intra picture prediction and 2D transform coding) as used in H.264/AVC [11]. Each picture is split into block shaped regions and the exact block portioning will be conveyed to the decoder. The first picture of a video sequence will be coded using only ‘intrapicture prediction’ mode which is a spatial prediction within the frame and the remaining pictures are coded using ‘interpicture prediction’ mode which is a temporal prediction between the frames.

The residual signal of the intra- or interpicture prediction which is the difference between the original and its prediction block is transformed by a ‘linear spatial transform’. These transform coefficients will then be scaled, quantized and entropy coded and then transmitted along with the prediction information.

The encoder duplicates the decoder processing loop such that it generates an identical prediction of a decoder. This is done by inverse scaling and inverse transforming of the encoded data to produce the decoder approximation of the residual signal. This residual signal is then added to the prediction signal and the result of this addition will be fed to one or two loop filters which smoothen out the artifacts generally induced by the block-wise processing and quantization step. The final picture representation which is the duplicate of the possible output in the decoder will be stored in a ‘decoded picture buffer’ and will be used for prediction of subsequent pictures.

## 2.2 Features of HEVC

### 2.2.1 Coding Tree Unit (CTU) and Coding Tree Block (CTB)

Unlike the macroblock (MB) of fixed 16x16 size in H.264/AVC the HEVC has the CTU which has a variable size upto 64x64 samples and the size can be selected by the encoder. The CTU is made of one luma CTB two chroma CTBs and syntax elements. The size of such CTBs can vary as 64x64 (1 CTB in the CTU), 32x32 (4 CTBs in the CTU) or 16x16 (8 CTBs in the CTU) and typically larger size gives better compression.

### 2.2.2 Coding Unit (CU) and Coding Block (CB)

CTBs are further partitioned into CU and can be either a) single CU or b) multiple CUs.

HEVC supports this partitioning using a tree structure and quadtree-like signaling. The decision of whether to code a picture area using intrapicture or interpicture prediction is made at these CUs. Each such CUs are made up of one luma CB, one chroma CB and associated syntax. The quadtree syntax of the CTU will specify the size and position of such luma and chroma CBs [16]. The size is variable and since CTU is the root of such a quadtree structure, the maximum size of a luma or a chroma CB can only go up to the size of the luma and chroma CTB respectively and the minimum allowable size is 8x8 or larger. The exact size of the luma and the chroma CBs depends on the decision made on the prediction type (intra or inter prediction). The luma and chroma CBs are predicted from the luma and chroma PBs which will be discussed next.

### 2.2.3 Prediction Unit (PU) and Prediction Block (PB)

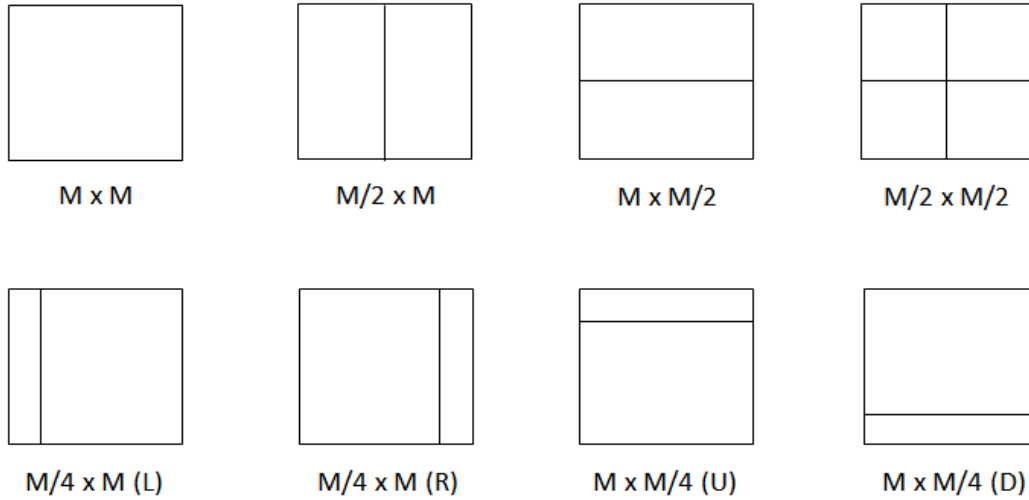


Figure 2.3 Modes of splitting a CB into PBs [11]

Each CU is partitioned into PUs and a tree of transform units (TU). Such PUs are made up of one luma PB, one chroma PB and associated syntax. The size of PBs is variable and sizes can vary from 64x64 down to 4x4. However to avoid worst-case memory bandwidth during motion compensation at the decoding stage, the smallest allowed size of PBs in the case of inter-picture prediction is restricted to 8x4 or 4x8 for uniprediction and 8x8 for biprediction. The modes of splitting a CB into PBs is illustrated in Figure 2.3 [11].

### 2.2.4 Transform Units (TU) and Transform Blocks (TB)

For coding the prediction residue, the CBs within a CTB are recursively partitioned into TBs and such a partitioning is signaled by a residual quadtree. This is illustrated in Figure 2.4 [11]. The luma and the chroma TB together make a transform unit TU. The size of a TU can vary as 4x4, 8x8, 16x16 and 32x32. A 4x4 luma TB that belongs to an intra coded region are transformed using an integer transform derived from the discrete sine transform [17].

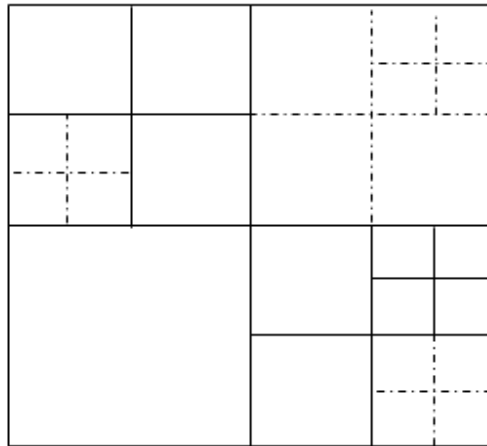


Figure 2.4 Division of CTB into CBs (solid line) and TBs (dotted line) [11]

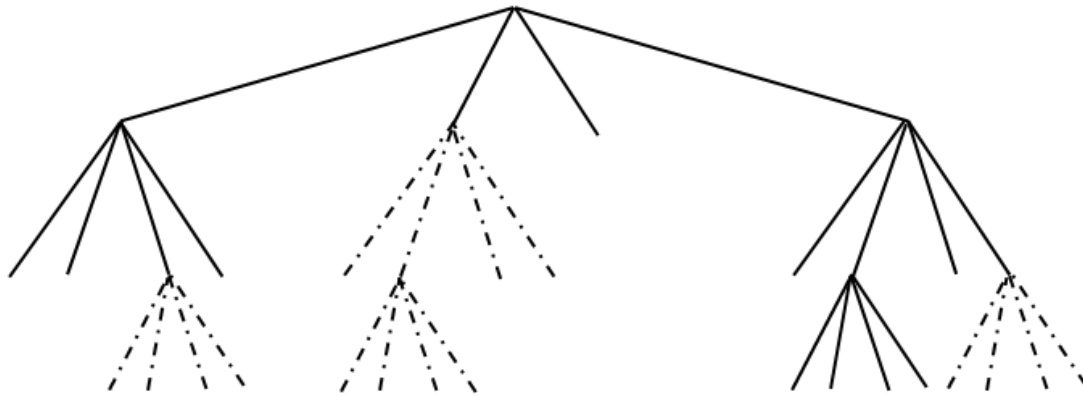


Figure 2.5 Quadtree corresponding to the figure 2.3 [11]

### 2.2.5 Slices and Tiles

A slice is a data structure which can be decoded independently from other slices of the same frame [11]. This slice can either be an entire frame or just a region of the frame. The main purpose of the slice is to provide the ability to resynchronize in case of a data loss. The maximum number of payload bits within a single slice is restricted and also the number of CTUs in each slice is varied in order to minimize the overhead of packetization.



A tile is a self-contained independently decodable rectangular region of the picture. The importance of tile is to enable the use of parallel processing architectures for encoding and decoding of pictures [18][19]. Unlike a slice the tile provides more capability for parallel processing rather than the error resilience. Tiles can also be used for purposes such as spatial random access to local regions of a picture. Typically each tile consists of approximately equal numbers of CTUs.

### *2.2.6 Loop Filters*

HEVC introduces two loop filters namely deblocking filter (DBF) [20] which is applied first and then the sample adaptive offset (SAO) filter which is applied next [21]. These filters are designed to operate during the inter-picture prediction loop. These filters are briefly explained here.

**In-loop deblocking filter:** This filter is very similar to the one designed for H.264/AVC but offers an extended support for parallel processing. The other differences are that, in HEVC the DBF is applied only to 8x8 sample grid while in H.264 it is applied to a 4x4 grid. The filter has also been provided strengths of 0 to 2. The DBF is first applied horizontally for filtering vertical edges and then vertically for filtering the horizontal edge. This feature is processed using multiple parallel threads [22].

**Sample adaptive offset (SAO):** The HEVC introduces a nonlinear amplitude mapping within the interpicture prediction loop after the application of the DBF. This helps in better reconstruction of the original amplitudes of the signal by using a look-up table which is described using a few additional parameters that can be determined by the histogram analysis at the encoder [23].

## 2.2.7 Intrapicture Prediction

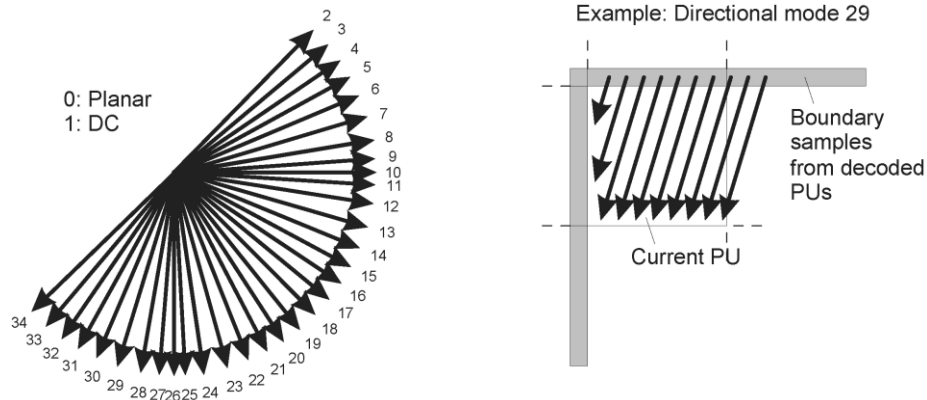


Figure 2.6 Modes and their directional orientation for intrapicture prediction [11]

The intrapicture prediction operates as per the size of the TB. The boundary samples which are previously decoded from spatially neighboring TBs are used to form the prediction signal. The HEVC offers directional prediction with 33 different directional orientations for every TB sizes ranging from 4x4 to 32x32. The prediction directions are shown in the Figure 2.6 [11].

## 2.3 Scalable Video Coding

Scalable video coding enables encoding of a high-quality video bitstream along with one or more subset bitstreams. It allows the adaptation of an encoded bitstream according to the needs of the end user. For example in case the end user is a mobile device such as a smart-phone or a notebook then the high resolution video can be clipped to adapt to the resolution of a mobile display thus increasing the transmission efficiency while providing reasonable display quality. The scalability is of different types namely: Temporal scalability; Spatial scalability and Quality scalability [24][25]. Spatial scalability helps in presenting the source content with a reduced picture size i.e. reduced spatial resolution while the temporal scalability provides the reduced frame rate i.e. the reduced temporal resolution version of the original source content. In contrast,

the quality scalability also referred to as signal-to-noise ratio (SNR) scalability or the fidelity scalability produces a lower reproduction quality of the original content at a lower bit rate amidst maintaining the same spatial and temporal resolutions.

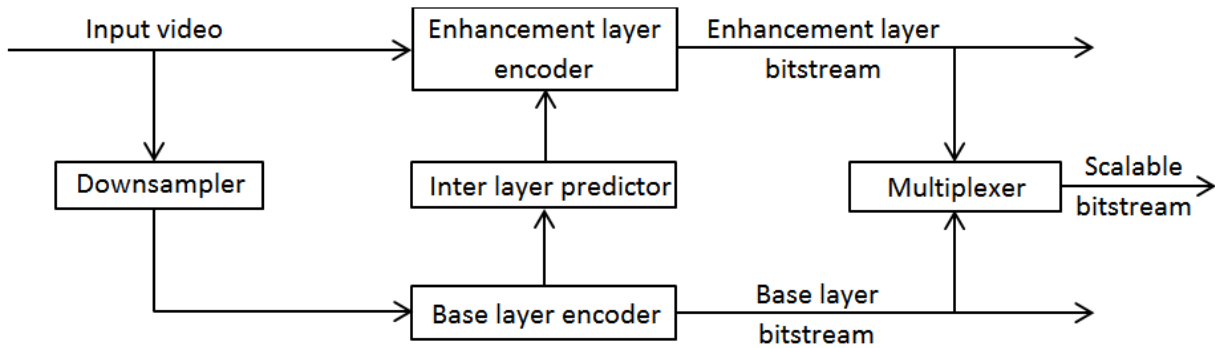


Figure 2.7 Scalable encoder with two layers [25]

The scalable coding is carried out using two layers and each layer is encoded using a separate encoder called as the base layer encoder and an enhancement layer encoder. This is as depicted in Figure 2.7 [25]. The base layer encoder will be just like a normal single-layer video encoder while the enhancement layer encoder will include additional coding features. At the end, the outputs of both the encoders will be multiplexed to form a scalable bitstream. In case of spatial scalable coding, the input video will be downsampled and then encoded by the base layer encoder, meanwhile the original input video will be encoded by the enhancement layer encoder. In case of quality scalable coding, both the encoders will have the same input [26].

### 2.3.1 Inter-Layer Prediction

The inter-layer prediction helps to improve the efficiency of the scalable video coding. It uses the data of one layer to predict the other layer. There are three different kinds of inter-layer prediction namely inter-layer intra prediction, inter-layer motion prediction and inter-layer residual prediction [25].

Inter-layer intra prediction: This method predicts the enhancement layer from the reconstructed and upsampled base layer.

Inter-layer motion prediction: This uses the motion data of the base layer for coding the enhancement layer motion. It also infers the motion data of the enhancement layer completely using the scaled motion data of the co-located base layer blocks [27].

Inter-layer residual prediction: The residual signal of the inter-picture coded block in the enhancement layer is predicted using the reconstructed and upsampled residual signals of the co-located base layer area and the motion compensation is applied using the reference pictures of the enhancement layer [28].

## 2.4 Summary

This chapter outlines the HEVC video coding standard and describes its' components and intra and inter prediction modes. The next chapter describes the inter-prediction mode and the algorithms which it uses in more detail.

## Chapter 3

### Interpicture Prediction

The HEVC standard defines the coding unit (CU) as the most basic processing unit. Unlike a macroblock (MB) in the previous video coding standards [5] whose size is fixed to 16 x 16 samples the coding unit has a variable size ranging from 16, 32 or 64 samples which provides the advantage of better compression performance when a larger CU is used to represent the data. Each CU has one luma coding block (CB) and two chroma coding blocks (CB) and associated syntax. The quadtree syntax of the coding tree unit (CTU) specifies the size and the positions of the luma and the chroma CUs. A coding tree block (CTB) may contain only one CU or can have multiple CUs and also each CU will have associated partitioning into prediction unit (PUs) and a tree of transform units (TUs) [11].

#### 3.1 Motion Vector Prediction

The motion vector prediction in HEVC standard follows a similar basic mechanism of the previous H.264/ AVC standard [10]. The HEVC standard has two reference lists namely L0 and L1 each of which can accommodate 16 references up to a maximum count of 8 unique pictures [29]. The reason for storing a picture more than once is to provide the encoder the ability to predict a picture using different multiple reference pictures according to their weights by using a technique called weighted prediction. Unlike H.264/ AVC the HEVC standard makes use of more complex advanced motion vector prediction (AMVP) for motion vector signaling. This involves the derivation of several most probable candidates based on the data from the adjacent prediction blocks (PBs) and the reference pictures [30]. Apart from the AMVP mode the HEVC standard also makes use of merge mode for motion vector signaling which allows to inherit the motion vectors (MVs) from temporal or spatial neighboring regions of a picture thus providing

significant data rate reduction by avoiding multiple transmission of repeated data. Unlike H.264/AVC the merge mode in HEVC standard has improved skipped and direct motion inference techniques.

The HEVC standard supports more prediction block (PB) partition shapes for interpicture predicted coding blocks (CB) when compared to the intrapicture-predicted coding blocks. The typical partition modes are PART\_2Nx2N, PART\_2NxN and PART\_Nx2N which are formed when a coding block is not split, split into two equal sized prediction blocks horizontally and split into two equal sized prediction blocks vertically respectively. The PART\_NxN is the coding block which is split into four equal sized prediction blocks which is only supported whenever the coding block size is equal to the smallest allowed coding block size. The HEVC standard allows four more partitioning types which supports the coding block to be split into two prediction blocks having different sizes such as PART\_2NxnU, PART\_2NxnD, PART\_nL x2N and PART\_nRx2N which are called as the asymmetric motion partitions as shown in Figure 3.1 [10].

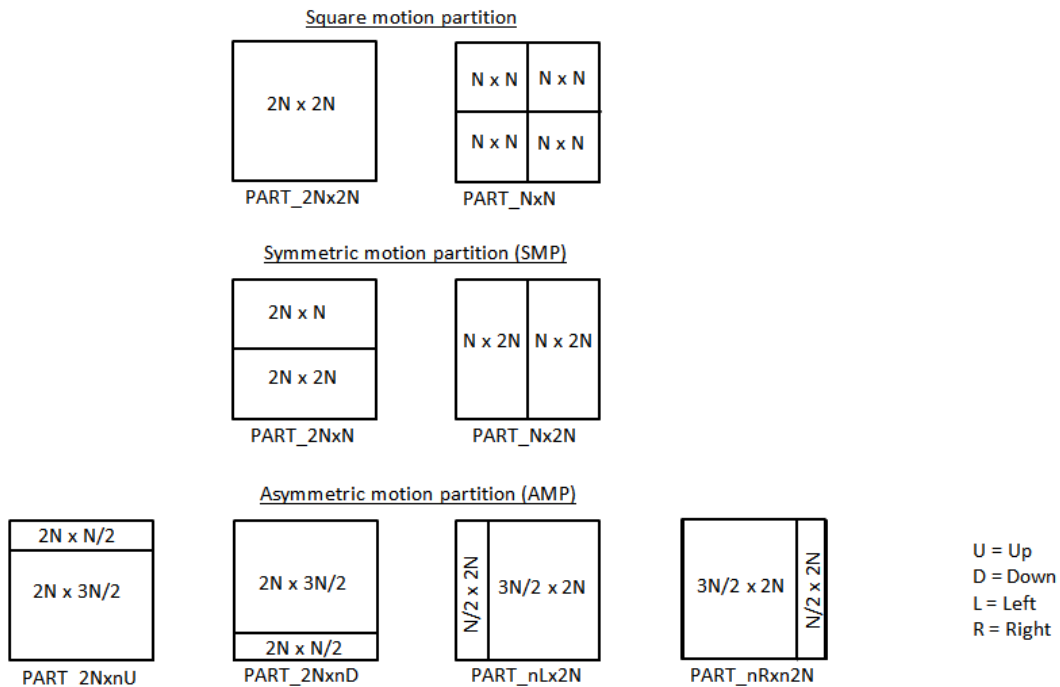


Figure 3.1 Partition modes in HEVC [10]

The advanced motion vector prediction (AMVP) in the HEVC standard makes use of a competition based scheme for selecting the candidates for the spatial and temporal motion vectors. The rate distortion optimization (RDO) process is used to select the best available motion vector from these set of candidates and the index of the selected candidate will be transmitted to the decoder [31] [32]. In this competition scheme, the AMVP has a maximum of two spatial neighboring candidates and one co-located temporal candidate and if the selection of such candidates is less than two then a zero motion vector will be added to the set of candidates. The derivation of these spatial and temporal candidates will be followed by a check for redundancy in order to remove duplicated motion vectors among the selected candidates. For each such inter predicted prediction unit a inter prediction indicator is transmitted which denotes the list used for prediction i.e. whether the reference picture is from list 0 or list 1 (in case of bi-prediction). Also, one or two reference indices will be transmitted when there are multiple reference pictures as shown in Figure 3.2 [10].

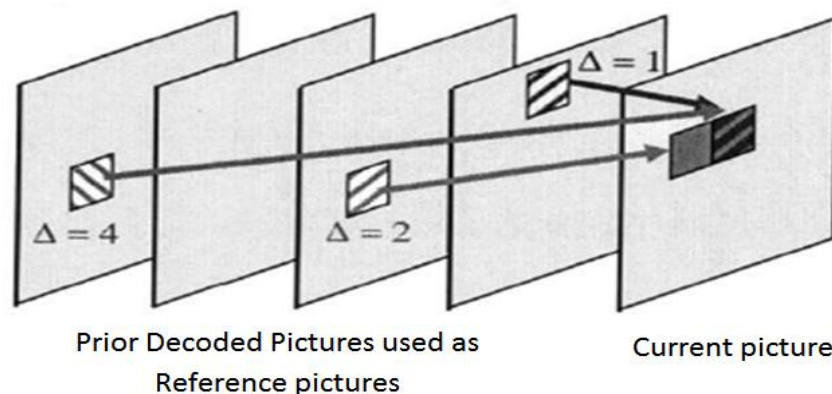


Figure 3.2 Multiple reference pictures for a single current picture [10]

The motion compensation in HEVC supports quarter sample motion vectors just like in H.264/AVC however with some key improvements. The fractional sample interpolation used in HEVC has a separable 8-tap filter (weights: -1, 4, -11, 40, 40, -11, 4, 1) for every half sample

positions and a 7-tap filter (weights: -1, 4, -10, 58, 17, -5, 1) for every quarter sample positions as shown in Figure 3.3[11] whereas the H.254/AVC standard made use of a two stage interpolation filter using six tap filters and rounding their results for integer and half sample positions as follows .

$$\begin{aligned}
a_{0,j} &= \left( \sum_{i=-3..3} A_{i,j} \text{qfilter}[i] \right) \gg (B - 8) \\
b_{0,j} &= \left( \sum_{i=-3..4} A_{i,j} \text{hfilter}[i] \right) \gg (B - 8) \\
c_{0,j} &= \left( \sum_{i=-2..4} A_{i,j} \text{qfilter}[1 - i] \right) \gg (B - 8) \\
d_{0,0} &= \left( \sum_{i=-3..3} A_{0,j} \text{qfilter}[j] \right) \gg (B - 8) \\
h_{0,0} &= \left( \sum_{i=-3..4} A_{0,j} \text{hfilter}[j] \right) \gg (B - 8) \\
n_{0,0} &= \left( \sum_{j=-2..4} A_{0,j} \text{qfilter}[1 - j] \right) \gg (B - 8)
\end{aligned}$$

Where the constant  $B \geq 8$  is the bit depth of the reference samples (which is typically  $B = 8$  for most of the applications). The symbol ‘ $\gg$ ’ indicates a arithmetic right shift operation. The samples labeled  $e_{0,0}$ ,  $f_{0,0}$ ,  $g_{0,0}$ ,  $i_{0,0}$ ,  $j_{0,0}$ ,  $k_{0,0}$ ,  $p_{0,0}$ ,  $q_{0,0}$  and  $r_{0,0}$  can be derived by applying the corresponding filters to the samples located at vertically adjacent  $a_{0,j}$ ,  $b_{0,j}$  and  $c_{0,j}$  positions as follows [11].

$$\begin{aligned}
e_{0,0} &= \left( \sum_{v=-3..3} a_{0,v} \text{qfilter}[v] \right) \gg 6 \\
f_{0,0} &= \left( \sum_{v=-3..3} b_{0,v} \text{qfilter}[v] \right) \gg 6 \\
g_{0,0} &= \left( \sum_{v=-3..3} c_{0,v} \text{qfilter}[v] \right) \gg 6 \\
i_{0,0} &= \left( \sum_{v=-3..4} a_{0,v} \text{hfilter}[v] \right) \gg 6 \\
j_{0,0} &= \left( \sum_{v=-3..4} b_{0,v} \text{hfilter}[v] \right) \gg 6 \\
k_{0,0} &= \left( \sum_{v=-3..4} c_{0,v} \text{hfilter}[v] \right) \gg 6 \\
p_{0,0} &= \left( \sum_{v=-2..4} a_{0,v} \text{qfilter}[1 - v] \right) \gg 6 \\
q_{0,0} &= \left( \sum_{v=-2..4} b_{0,v} \text{qfilter}[1 - v] \right) \gg 6 \\
r_{0,0} &= \left( \sum_{v=-2..4} c_{0,v} \text{qfilter}[1 - v] \right) \gg 6.
\end{aligned}$$

The HEVC makes use of a single consistent separable interpolation for obtaining the fractional samples without the requirement of intermediate rounding operations thus provides improved precision with simplicity. The other advantage of using the longer filters like the 7 and



8-tap filters is that the interpolation precision is improved [33][34]. The 7-tap filters are sufficient for quarter -sample positions since they are much closer to the integer positions.

$A_{-1,-1}$				$A_{0,-1}$	$a_{0,-1}$	$b_{0,-1}$	$c_{0,-1}$	$A_{1,-1}$				$A_{2,-1}$
$A_{-1,0}$				$A_{0,0}$	$a_{0,0}$	$b_{0,0}$	$c_{0,0}$	$A_{1,0}$				$A_{2,0}$
$d_{-1,0}$				$d_{0,0}$	$e_{0,0}$	$f_{0,0}$	$g_{0,0}$	$d_{1,0}$				$d_{2,0}$
$h_{-1,0}$				$h_{0,0}$	$i_{0,0}$	$j_{0,0}$	$k_{0,0}$	$h_{1,0}$				$h_{2,0}$
$n_{-1,0}$				$n_{0,0}$	$p_{0,0}$	$q_{0,0}$	$r_{0,0}$	$n_{1,0}$				$n_{2,0}$
$A_{-1,1}$				$A_{0,1}$	$a_{0,1}$	$b_{0,1}$	$c_{0,1}$	$A_{1,1}$				$A_{2,1}$
$A_{-1,2}$				$A_{0,2}$	$a_{0,2}$	$b_{0,2}$	$c_{0,2}$	$A_{1,2}$				$A_{2,2}$

Figure 3.3 Integer and fractional sample luma interpolation in HEVC [11]

### 3.2 Merge Mode in HEVC.

Each inter coded prediction unit will have a set of motion parameters which consist the motion vector, reference picture index, reference picture list usage flag which must be used during the inter prediction sample generation which are signaled in an explicit or implicit way. However, the motion information analysis of a picture shows that most of the prediction units have identical motion information as they could have resulted due to the movement of a single large object. Thus instead of transmitting the same motion information for every PU in the picture, significant bitrate reduction can be achieved by transmitting all such PUs which have identical motion information as to have a common reference PU called as the base or the seed

PU using which they can obtain the motion information. This method of coding the motion information of the PU is called the merge mode.

The merge mode finds the neighboring inter coded PU whose motion information can be used as the motion information of the current PU in question. This process is carried out by the encoder by investigating the motion information of multiple spatial and temporal neighboring candidate PUs and then transmitting the index of the chosen candidate. This merge mode can be applied to any inter coded PU and is not just limited to skip mode. When a CU is coded as a skip mode it will be represented as a single PU having no significant motion vectors, transform coefficients or reference picture index or reference picture list.

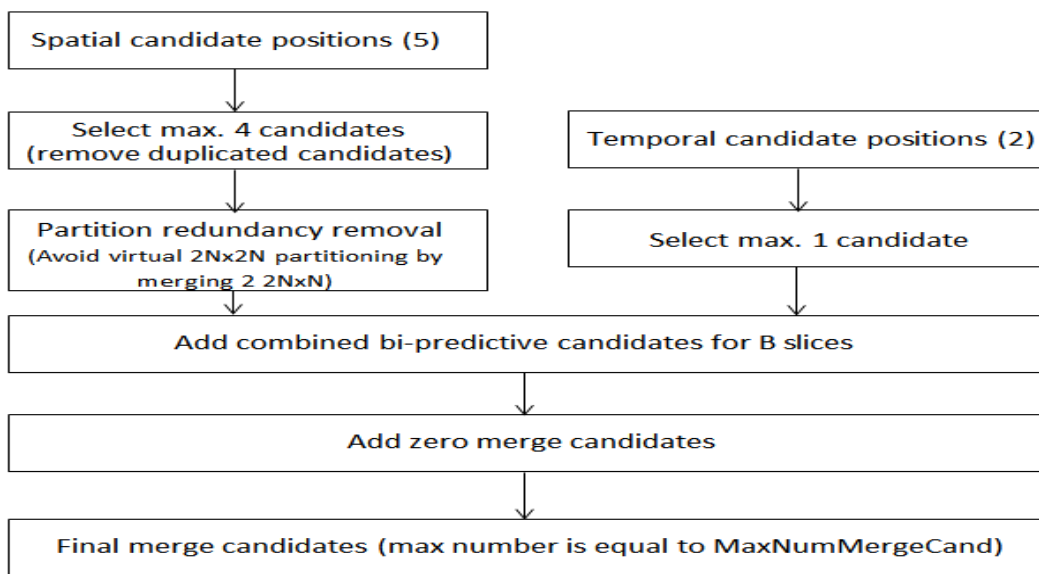


Figure 3.4 Process of obtaining the merge candidates in HEVC [35]

The merge mode in the HEVC standard considers two types of candidates which are spatial and temporal merge candidates. The spatial merge candidate is obtained by selecting a list of four merge candidates by considering five candidates which are located in five different positions. During the process of candidate selection the candidates having the same motion information as the other candidates are removed from the list to avoid duplicates. The candidates

within the same merge estimation region (MER) are also rejected which helps in improving the parallel merge processing.

The temporal merge candidate is obtained by selecting a maximum of one merge candidate by considering two candidates. The number of merge candidates selected in the list is kept constant since it is assumed to be constant at the decoder and avoids additional overhead of transmitting the number of candidates selected. If the number of candidates does not reach the maximum number of merge candidates then additional candidates are generated and if the number of candidates reaches the maximum number of merge candidates then the candidate generation process will be halted. In the case of B predicted slices combined bi-predictive candidates will be generated using the candidates present in the list of spatio-temporal candidates. This process is described in the Figure 3.4 [35].

### 3.2.1 Spatial Merge Candidates

The process of obtaining the spatial merge candidates involves selection of four merge candidates after considering five candidates at different positions as shown in Figure 3.5 [35]. The order used for deriving these candidates is  $A_1 - B_1 - B_0 - A_0 - B_2$ . The last position  $B_2$  is considered only when any of the other position among  $A_1, B_1, B_0, A_0$  is intra coded or is not available.

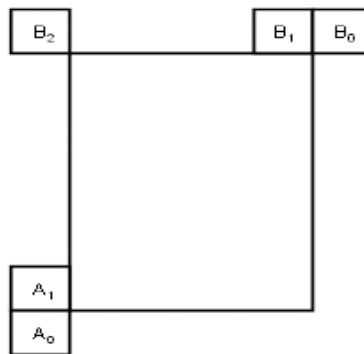


Figure 3.5 Positions of spatial merge candidates [35]

In case of the second PU of an  $N \times 2N$  or  $nL \times 2N$  or  $nR \times 2N$  partitions the position  $A_1$  will be excluded and not be considered in order to prevent  $2N \times 2N$  partition emulation. In this case the order of deriving the candidate will be  $B_1 - B_0 - A_0 - B_2$  as depicted in Figure 3.6 (a) [35] and in the case of second PU of an  $2N \times N$  or  $2N \times nU$  or  $2N \times nD$  partitions the position  $B_1$  will be excluded and will not be considered and the order for deriving candidates will be  $A_1 - B_0 - A_0 - B_2$  as shown in Figure 3.6 (b) [35].

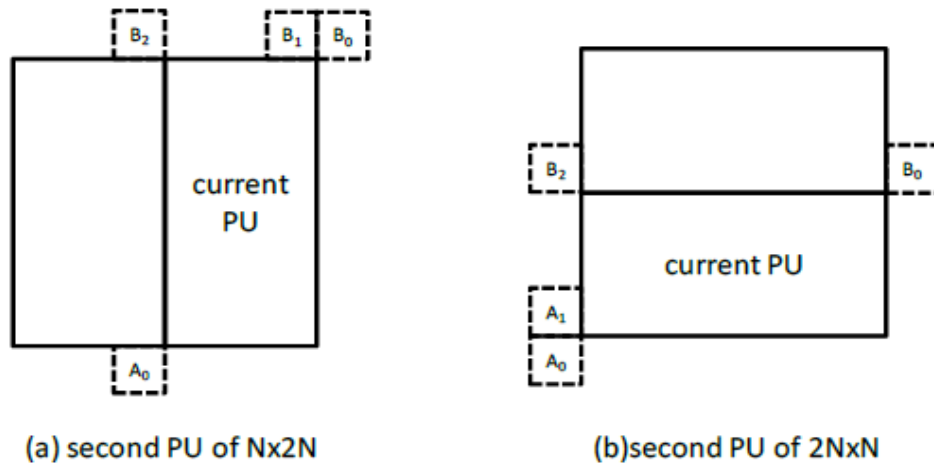


Figure 3.6 Positions for second PU of  $N \times 2N$  and  $2N \times N$  partitions [35]

### 3.2.2 Temporal Merge Candidates

The process of obtaining the temporal merge candidate involves finding a co-located PU which is present in a picture with smallest picture order count (POC) difference with the current picture using which a scaled motion vector will be derived. The reference picture list which must be used to find the picture with co-located PU will be signaled explicitly in the slice header. The process of scaling the motion vector of the co-located PU to the current PU for temporal merge candidate is obtained as shown in the Figure 3.7 [35].

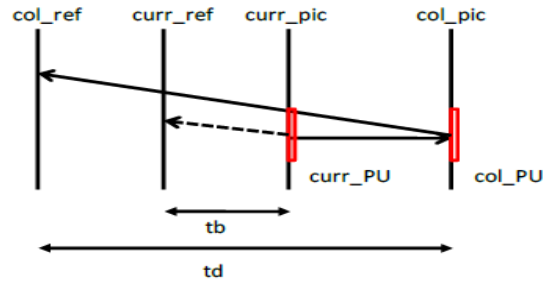


Figure 3.7 Motion vector scaling for temporal merge candidate [35]

The scaled motion vector for the temporal merge candidate is shown in dotted line in Figure 3.7 [35]. This is obtained by scaling the motion vector of the co-located PU using the POC distances  $t_b$  and  $t_d$  where  $t_b$  indicates the POC difference between the current picture and its reference picture and  $t_d$  indicates the POC difference between the co-located picture and its reference picture. The reference picture index of such temporal merge candidates will be set to zero. In case of a B-slice two motion vectors will be combined to make a bi-predictive merge candidate. One of these motion vectors is obtained from reference picture list 0 and the other from reference picture list1. Once the reference picture for obtaining the co-located PU is selected then the position of the co-located Pu will be selected among two candidate positions which are C3 and H as shown in Figure 3.8 [35]. In case the PU at position H is not available or is outside the current coding tree unit (CTU) or is intra coded then the other candidate position C3 will be used. This is as shown in Figure 3.8 [35].

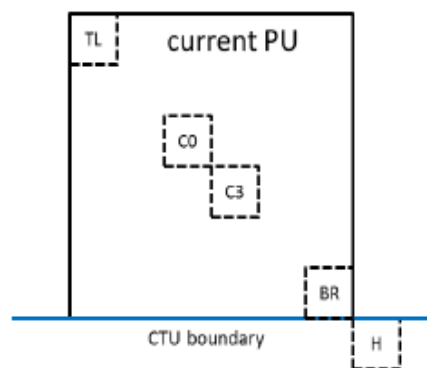


Figure 3.8 Positions of the candidates for temporal merge [35]

As a further enhancement to the currently available methods of obtaining the spatio-temporal merge candidates HEVC standard has recently introduced two more additional types of merge candidates namely the combined bi-predictive merge candidate and zero merge candidate. The combined bi-predictive merge candidates will be generated making use of the available spatio-temporal merge candidates and it is used only in case of B-slice.

### 3.3 Proposed Method

The existing technique for motion merge in the HEVC standard involves deriving the spatial and temporal candidates which requires selecting a list of candidates among the candidates located at different pre-defined positions. The basic requirement for this method is that the candidate PUs must contain motion information i.e. it cannot by itself be a motion merged PU and hence it must obtain the motion information from its base PU and store it for future PU merging. This condition for candidates limits the size of the block which can be assigned to follow a merge mode and since it involves the fetching and storing of motion information for every PUs during the in-loop decoding within the encoder it increases the time overhead, also studies have shown that a video signal typically contains large number of spatially adjacent regions and each such region have homogeneous motion parameters [36] [37]. To avoid the limit on the size of the motion merge block that can be formed and to make use of the inherent property of typical video signal the following technique is proposed for spatial motion merge candidate selection.

For a current  $2N \times 2N$  PU, five candidates located at five different pre-defined positions as shown in the Figure 3.5 [35] are verified for similar motion parameters as that of the current PU in the order  $A_1 - B_1 - B_0 - A_0 - B_2$  to form the list of candidates as before, however in case the candidate PU being considered is coded in merge mode then its base PU will be considered as

the direct candidate for matching the motion parameters of the current PU and in case of a match the direct candidate which is the base PU of the immediate candidate will be considered as the base of the current PU. Also to make use of the inherent property of typical video data that a large number of spatially adjacent regions share a homogenous motion, a threshold size typically of  $16 \times 16$  is considered for CUs and the PUs of a CU of size smaller than this threshold will all be coded as to follow the merge mode where the top left PU of the CU will be considered as the base PU from which the motion parameters can be obtained. This method of skipping the candidate selection process for CUs of size smaller than the threshold provides an advantage of reduction in encoding time with little loss of PSNR for videos capturing normal motion. The same technique will be followed to  $N \times 2N$ ,  $nL \times 2N$  and  $nR \times 2N$  partitioned PUs shown in Figure 3.6 (a) [35] and the candidate  $A_1$  will not be considered to avoid computational complexity and the order of candidate consideration is  $B_1 - B_0 - A_0 - B_2$  and in case of  $2N \times N$ ,  $2N \times nU$  and  $2N \times nD$  partitioned PUs shown in Figure 3.6 (b) [35] the candidate  $B_1$  will not be considered and the order of candidate consideration is  $A_1 - B_0 - A_0 - B_2$ .

### 3.4 Summary

The chapter outlines the inter-prediction process in the present HEVC standard and the existing merge mode candidate selection process along with the motivations for its improvement. Chapter 4 outlines the experimental setup, results and conclusions which are drawn based on the proposed selective motion merge process.

## Chapter 4

### Results

#### 4.1 Test Conditions

To test the performance of the proposed motion merge encoding technique the HEVC reference software HM 13.0 [38] was used. The ‘random access profile’ and group of pictures (GOP) of length 8 was used for conducting the test. The ‘random access profile’ consist of 1 Intra frames (I-frame) followed by 7 inter bi-directional frames (B-frames) and follows a non-sequential approach in choosing the picture order count of the frames as shown in Figure 4.1.

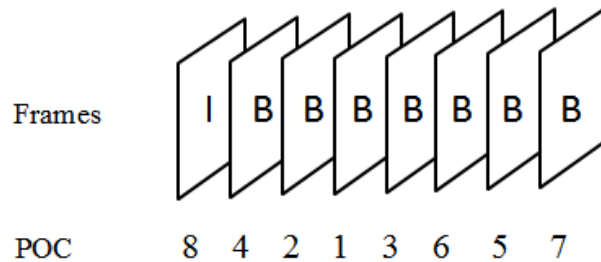


Figure 4.1 GOP used in 'random access profile' of HEVC

The coding tree block (CTB) size used was 64x64 along with a maximum depth of 4 with a minimum coding unit (CU) size of 8x8 pixels for the luma component. The proposed algorithm was tested with four different quantization parameters (QP) of 22, 27, 32, 37 using test sequences recommended by JCT-VC for 50 frames of each sequence [39]. A frame of each of the test sequences is shown in Appendix A.

Table 4.1 List of test sequences [46]

no	Sequence	Resolution	Type	no of frames
1	Race Horses	416x240	WQVGA	50
2	BQ Mall	832x480	WVGA	50
3	Basketball Drill Text	832x480	WVGA	50
4	Kristen and Sara	1280x720	SD	50
5	Basketball Drive	1920x1080	HD	50



## 4.2 Reduction in Encoding Time

The time for encoding the test sequences has reduced by proposed algorithm of selective motion merge 13-24% when compared to the unmodified HM 13.0 reference software. The obtained results indicating the reduction in encoding time is shown in Figure 4.2 through Figure 4.6. The figure show the time taken to encode the test sequences under different quantization parameter values by the original and the proposed algorithm.

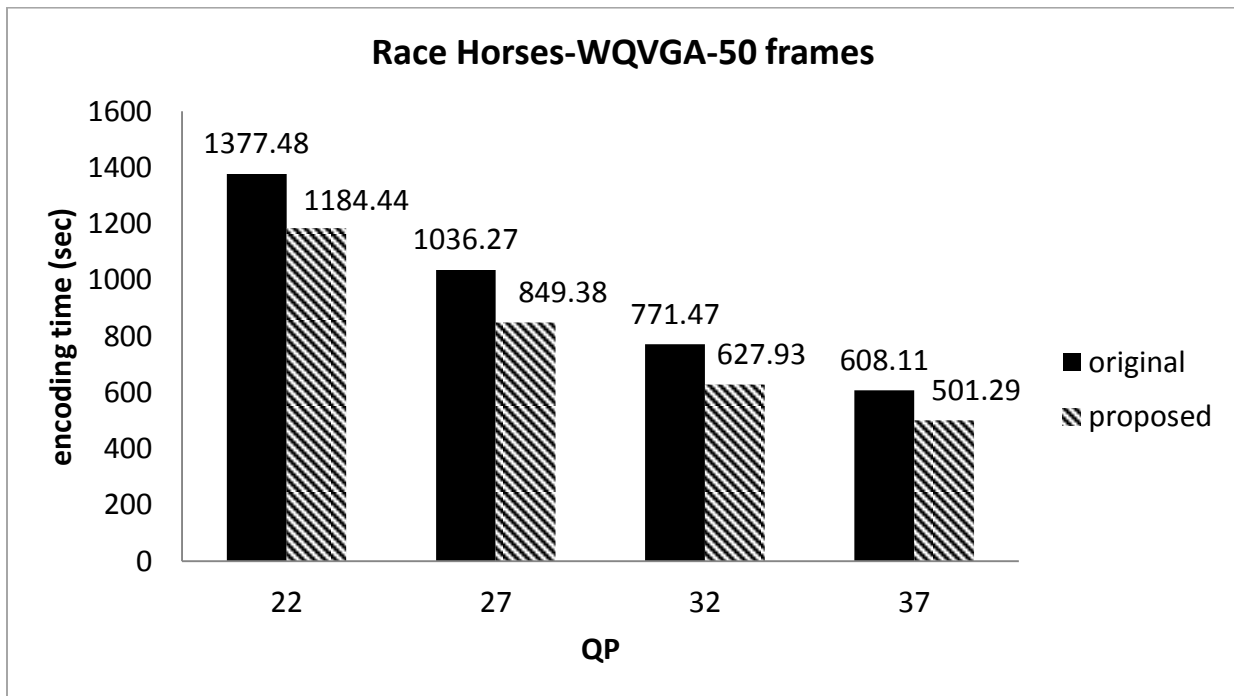


Figure 4.2 Encoding time vs quantization parameter for 'Race Horse' sequence

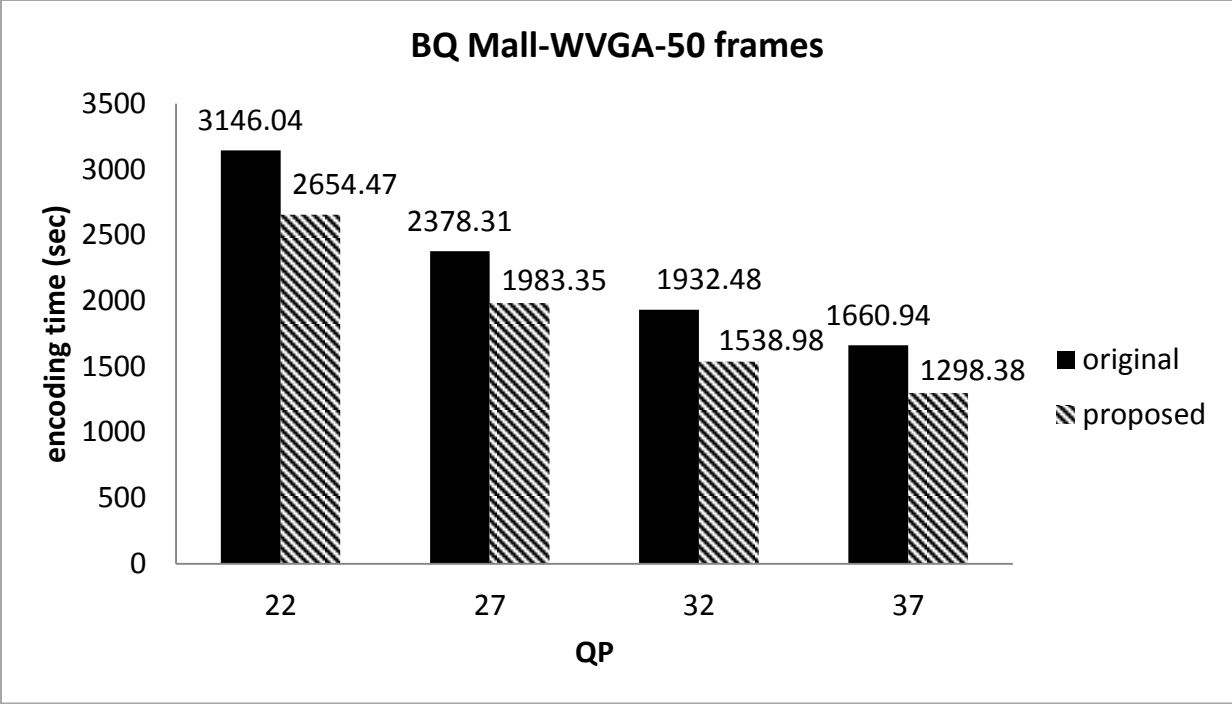


Figure 4.3 Encoding time vs quantization parameter for 'BQ Mall' sequence

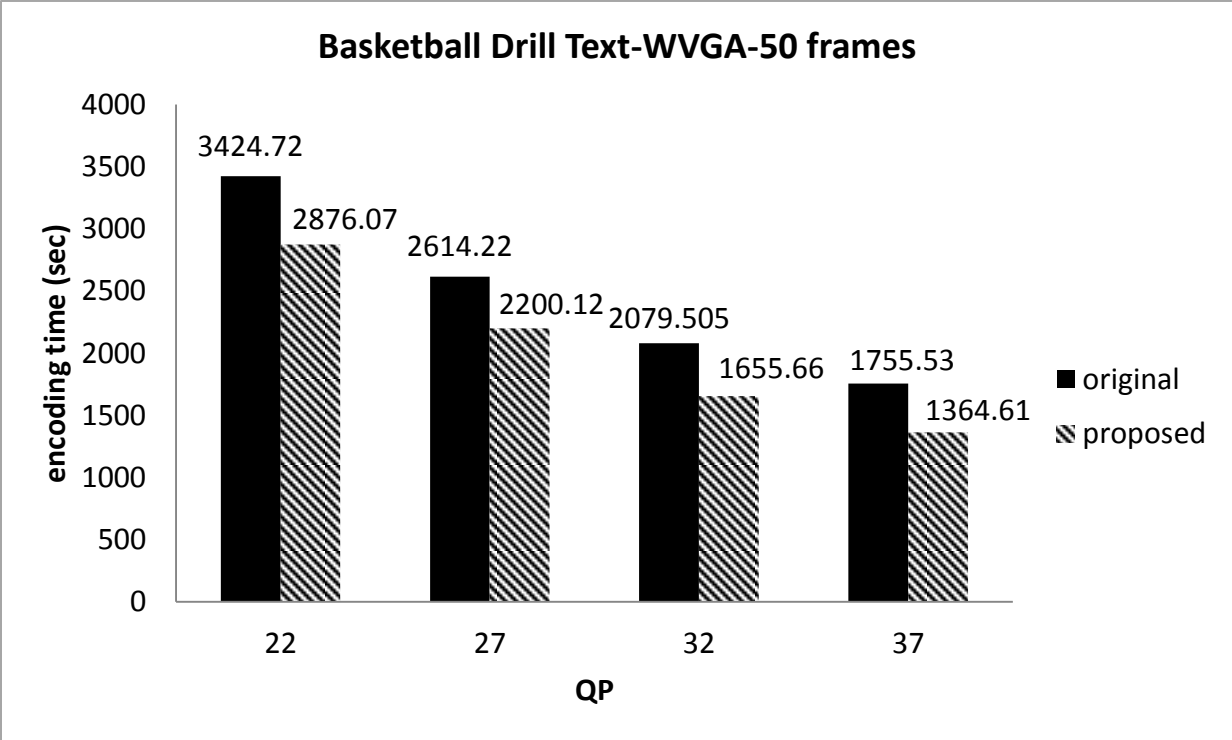


Figure 4.4 Encoding time vs quantization parameter for 'Basketball Drill Text' sequence

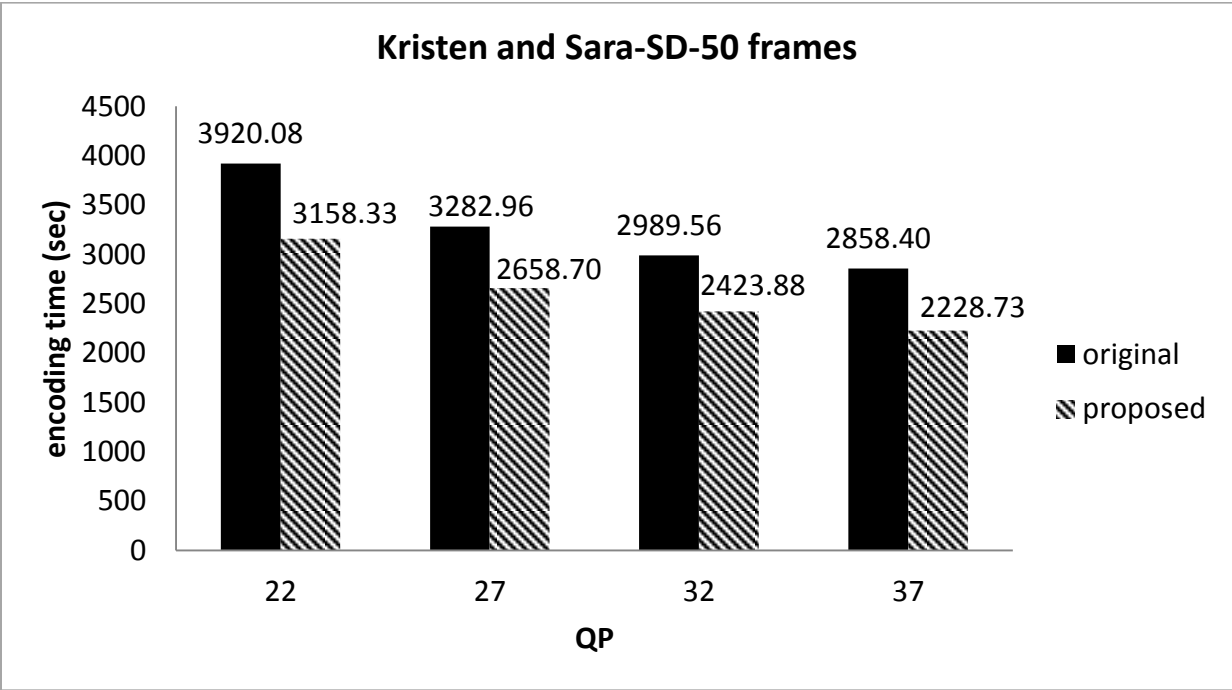


Figure 4.5 Encoding time vs quantization parameter for 'Kristen and Sara' sequence

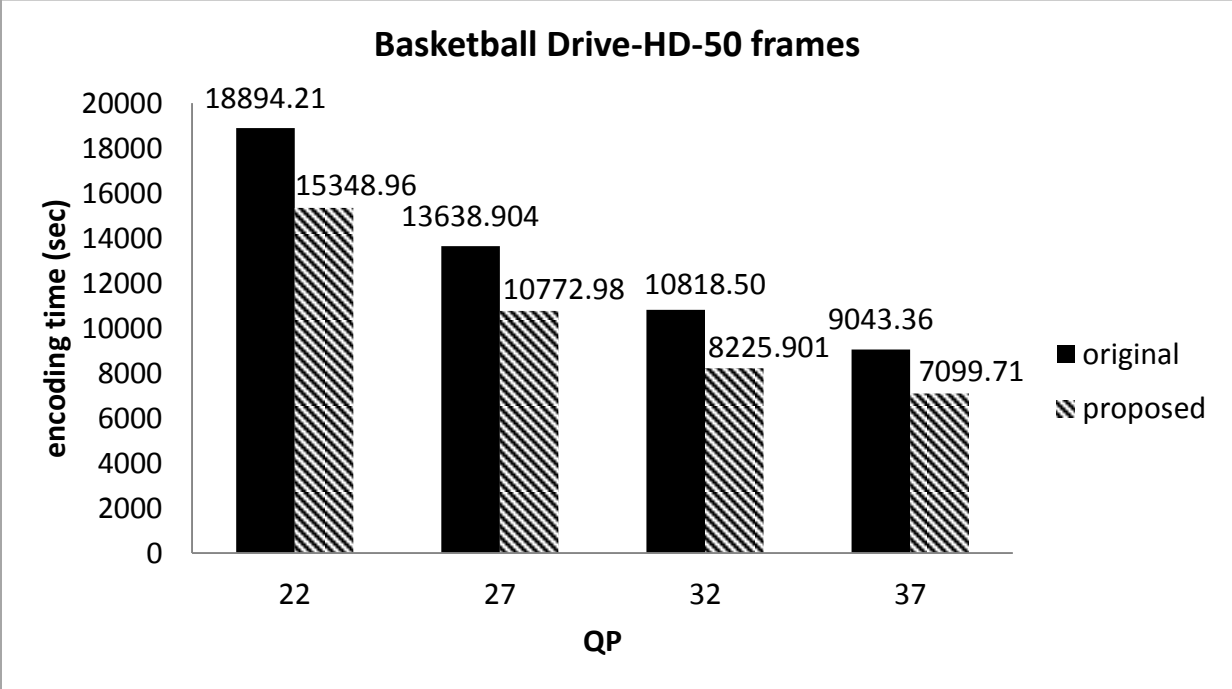


Figure 4.6 Encoding time vs quantization parameter for 'Basketball Drive' sequence

### 4.3 Reduction in Bitrate

The proposed algorithm provides a reduction of 2-7% in bitrate when compared against the HEVC reference software HM 13.0. The results which indicate this reduction in bitrate is shown in Figure 4.7 through Figure 4.11 which was obtained by encoding the test sequence with different quantization parameter values.

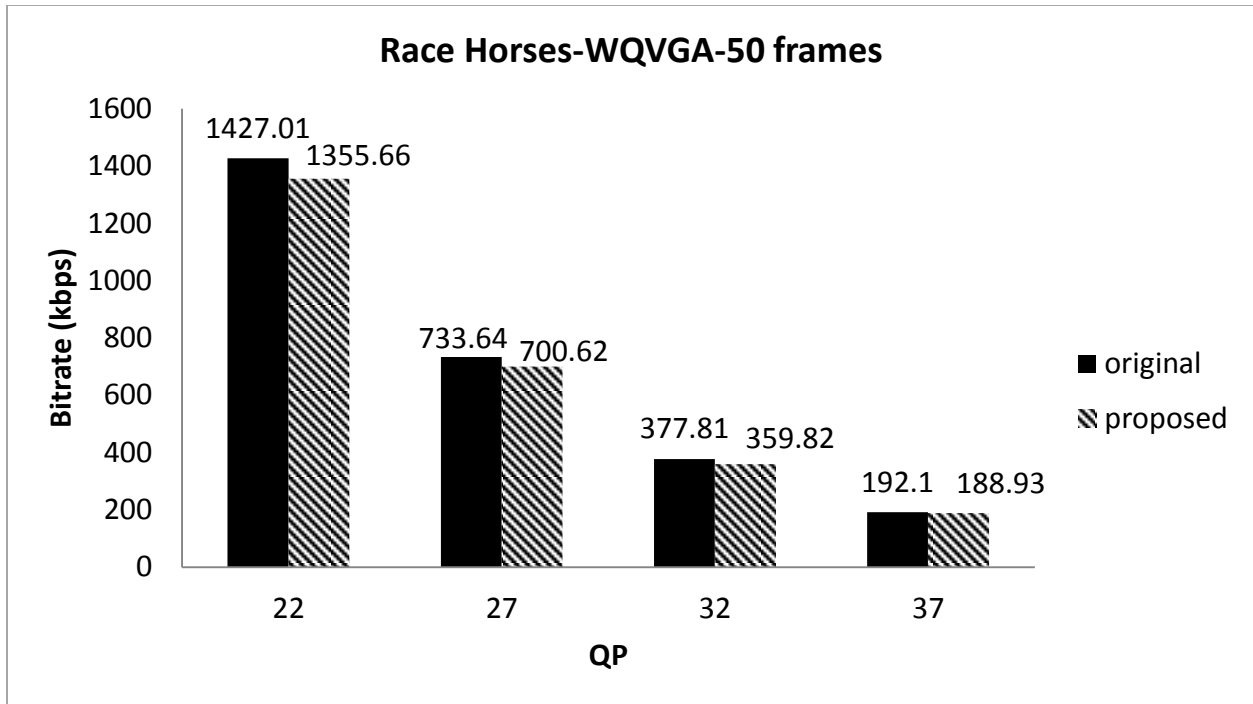


Figure 4.7 Bitrate vs quantization parameter for 'Race Horse' sequence

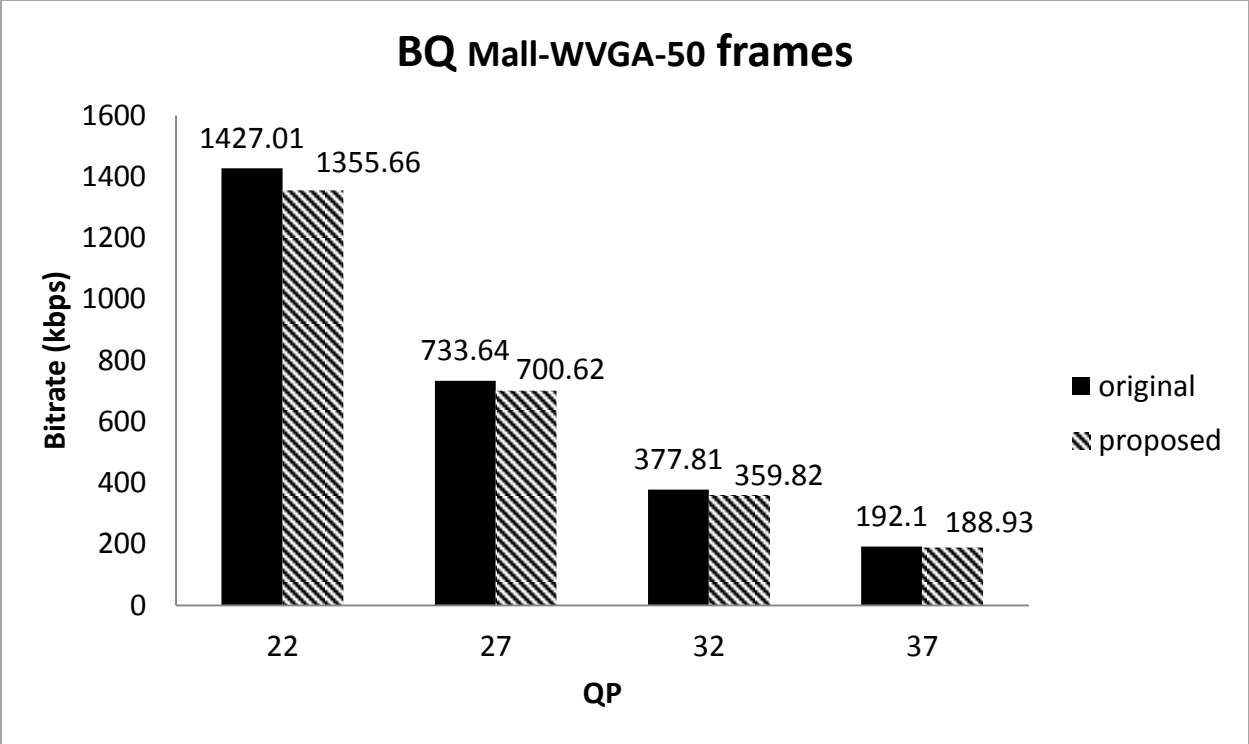


Figure 4.8 Bitrate vs quantization parameter for 'BQ Mall' sequence

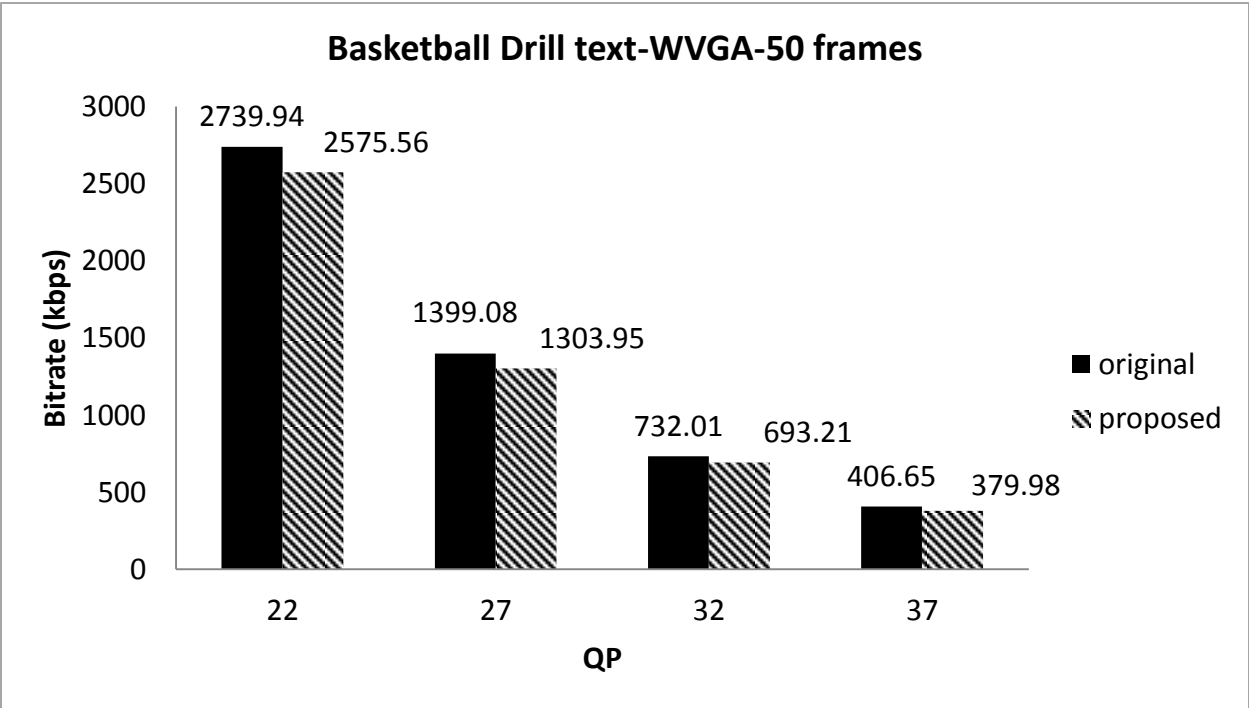


Figure 4.9 Bitrate vs quantization parameter for 'Basketball Drill Text' sequence

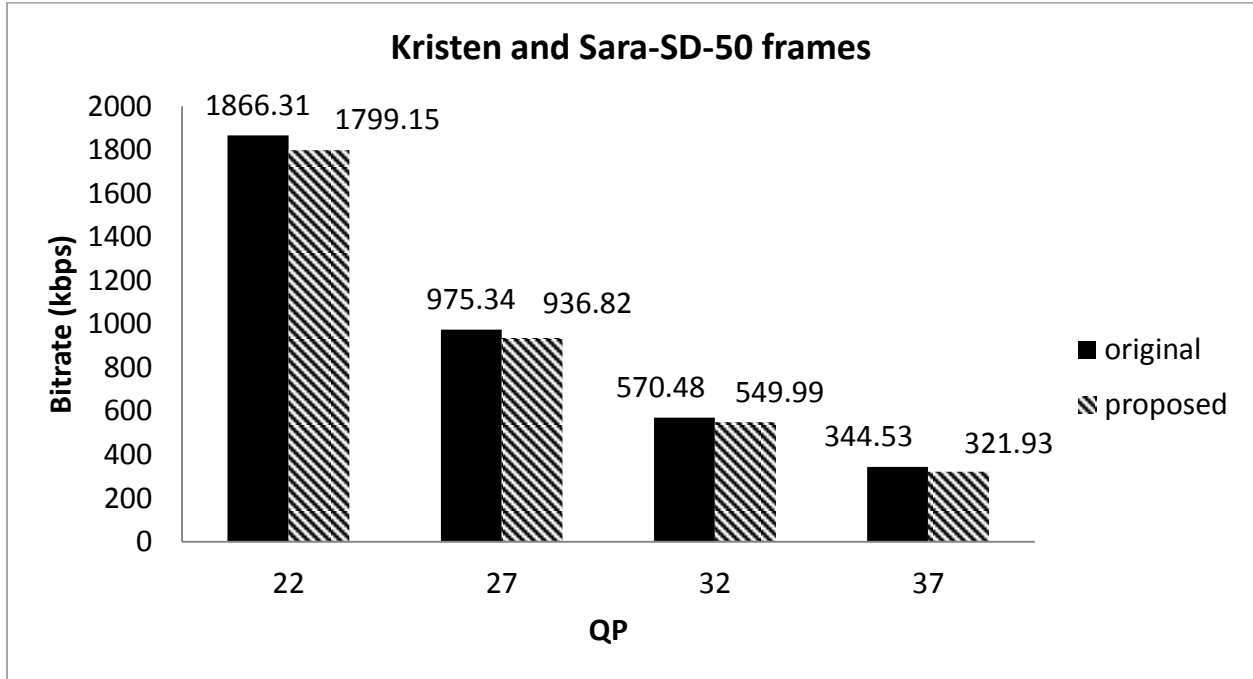


Figure 4.10 Bitrate vs quantization parameter for 'Kristen and Sara' sequence

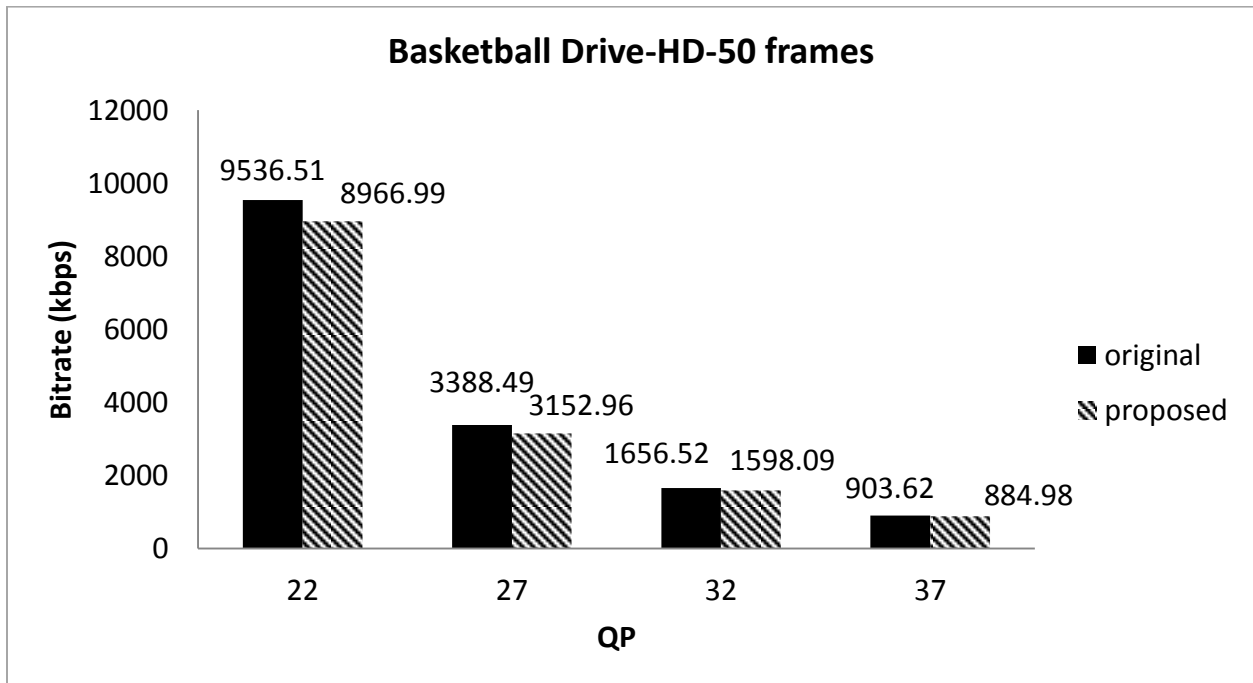


Figure 4.11 Bitrate vs quantization parameter for 'Basketball Drive' sequence

#### 4.4 Reduction in PSNR

The proposed algorithm provides a reduction in bitrate and encoding time but leads to reduction in PSNR of 2-6% which slightly impacts the quality of the signal. The results indicating the reduction in PSNR quality for different quantization parameters are shown in Figure 4.12 through Figure 4.16.

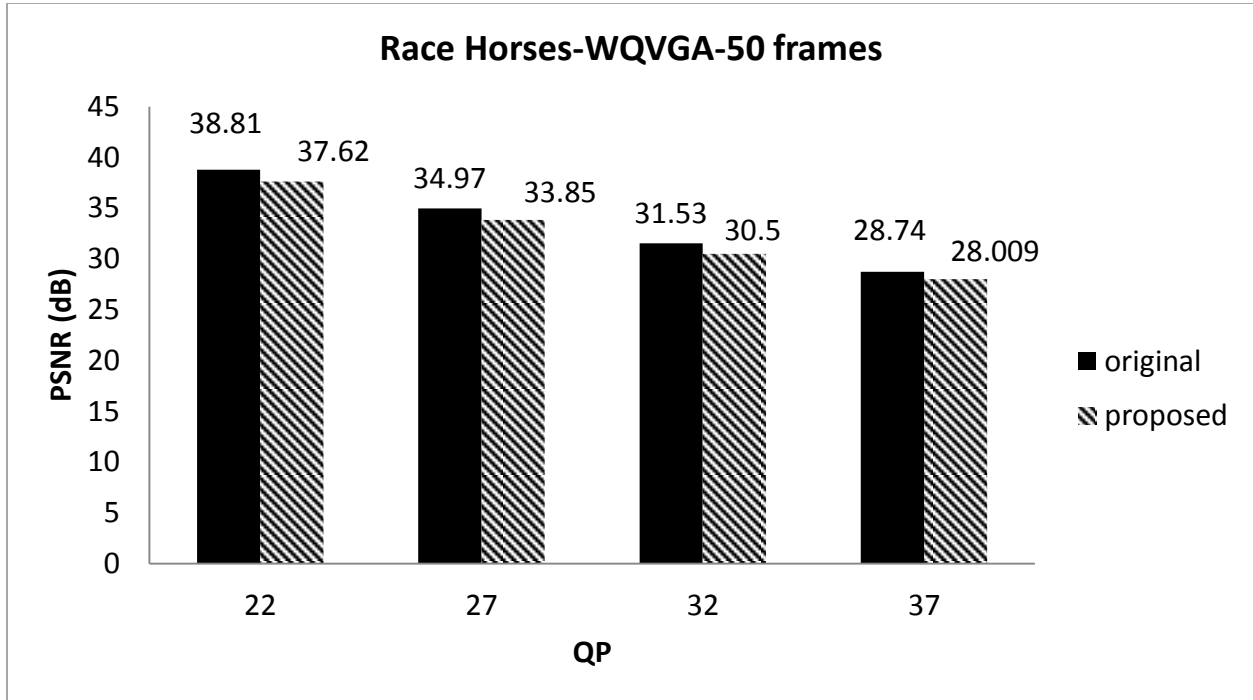


Figure 4.12 PSNR vs quantization parameter for 'Race Horse' sequence

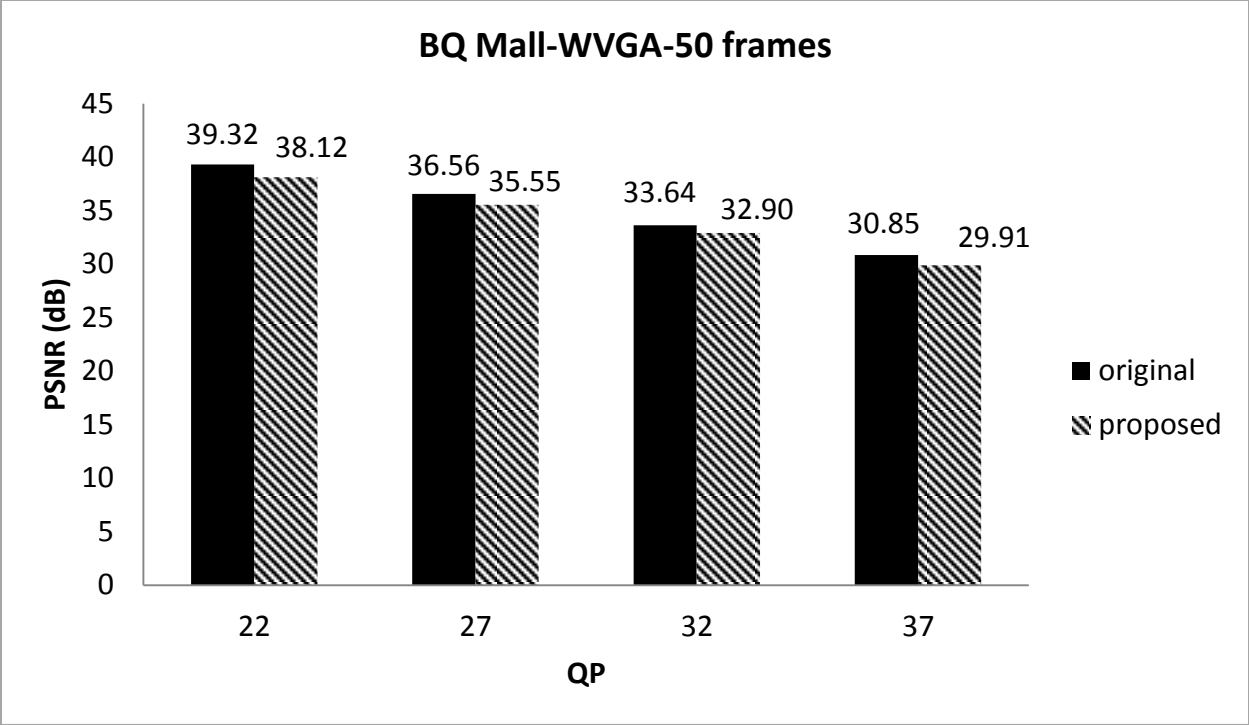


Figure 4.13 PSNR vs quantization parameter for 'BQ Mall' sequence

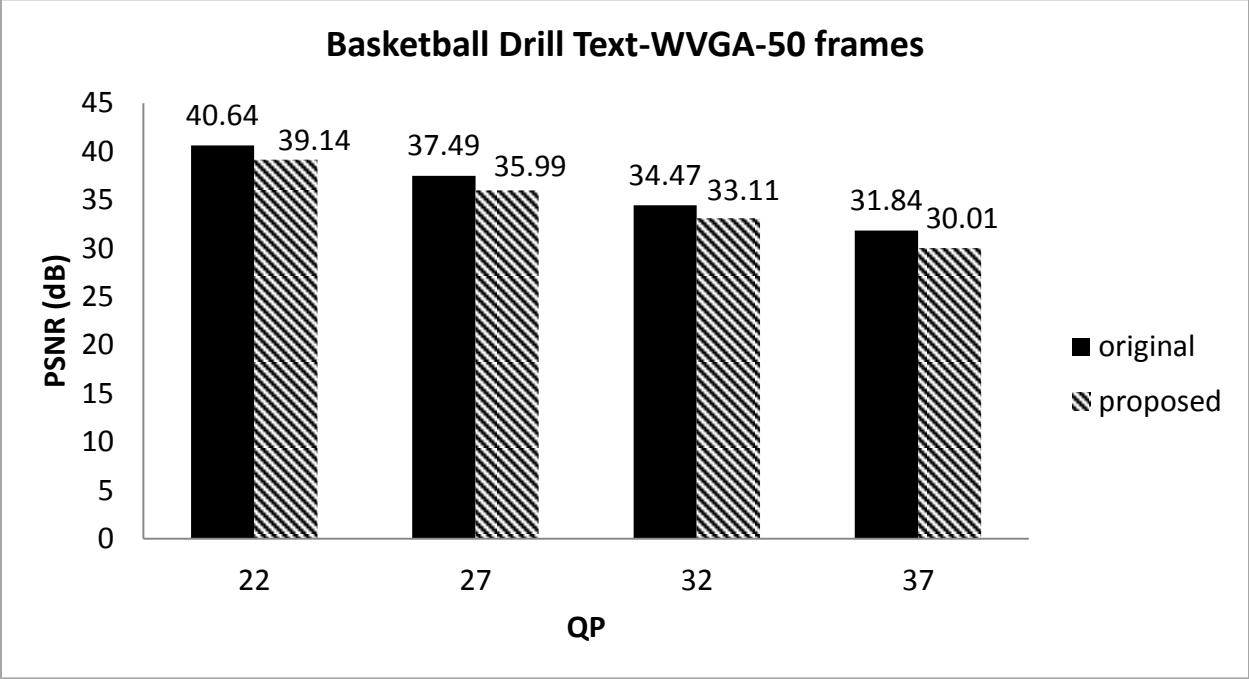


Figure 4.14 PSNR vs quantization parameter for 'Basketball Drill Text' sequence



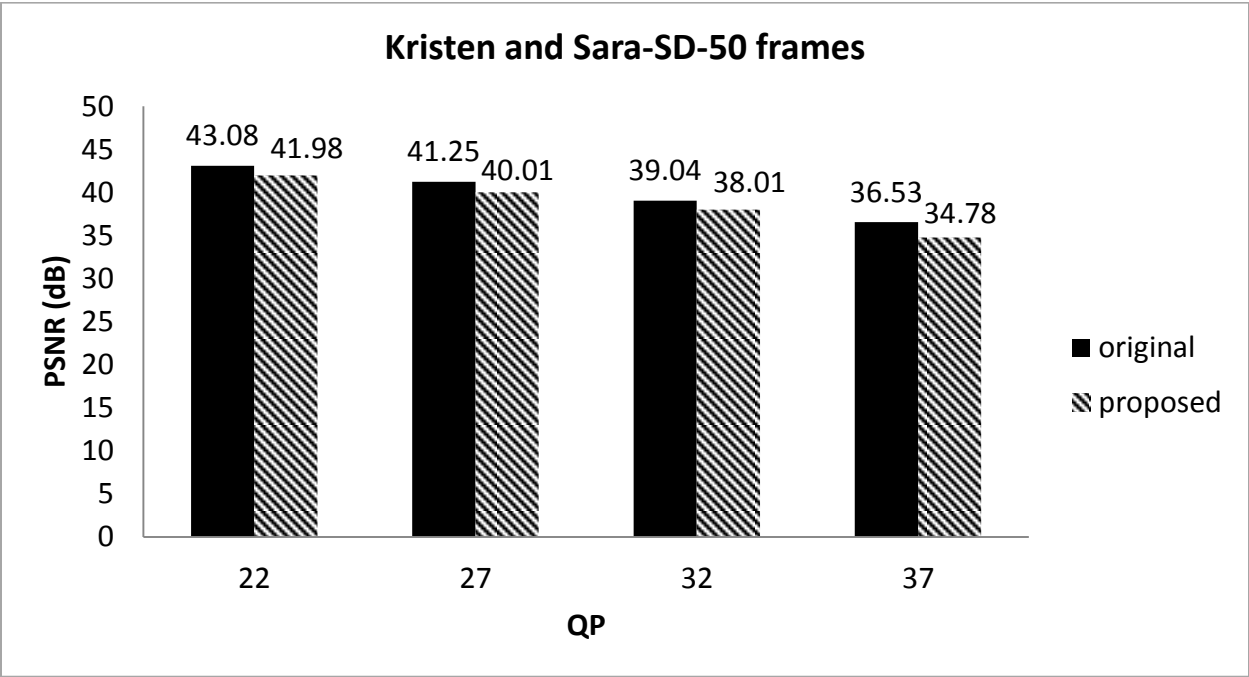


Figure 4.15 PSNR vs quantization parameter for 'Kristen and Sara' sequence

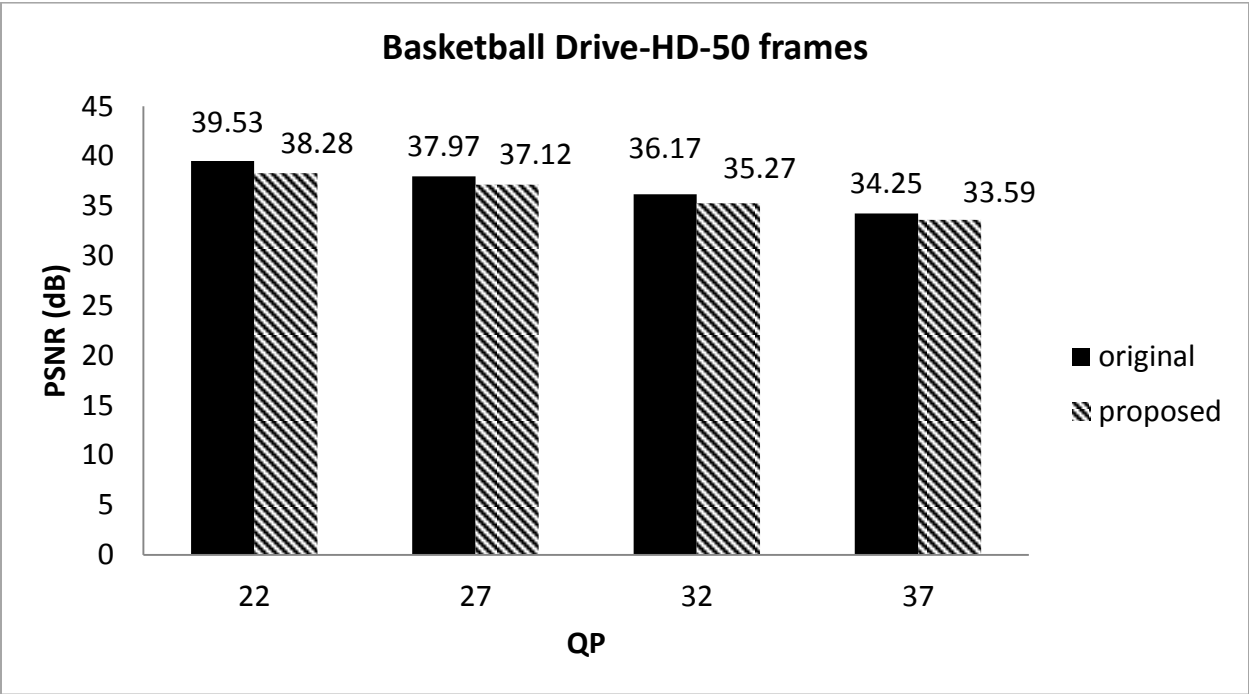


Figure 4.16 PSNR vs quantization parameter for 'Basketball Drive' sequence

#### 4.5 BD-PSNR and BD-Bitrate

To objectively evaluate the coding efficiency of a different video codecs Bjontegaard Delta PSNR (BD-PSNR) was introduced [39]. This metric is based on the rate-distortion (R-D) curve fitting using which the BD-PSNR is able to provide a good evaluation of the R-D performance of a video codec against another video codec. This metric provides good information on the quality of the video bitstream generated [40][41]. The metric suggests that to categorize a video codec as an improvement over another video codec it must obtain positive values of BD-PSNR in terms of decibels (dB) and negative values of BD-bitrate in terms of percentage (%). The BD-PSNR and values of the proposed vs original algorithm indicates positive values ranging from +0.29 to +0.56 and BD-bitrate values of -65% to -31%. The results indicating the BD-PSNR are in the Figure 4.17 to Figure 4.21 and the results indicating the BD-bitrates are in the Figure 4.22 through Figure 4.26.

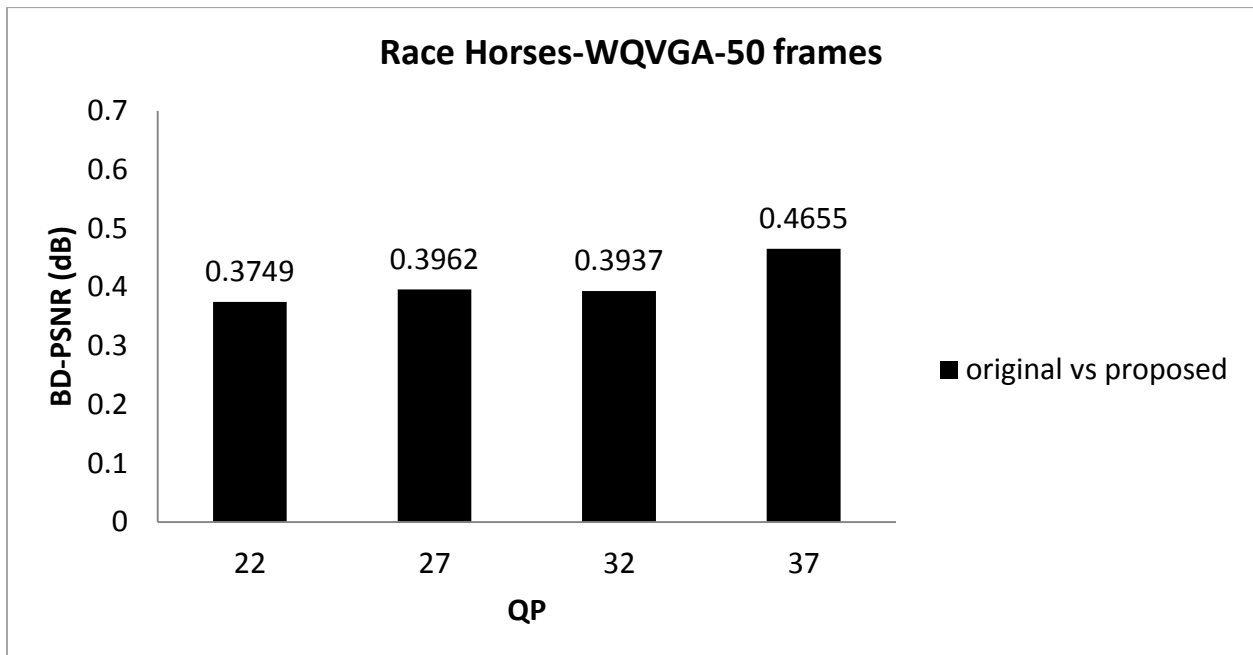


Figure 4.17 BD-PSNR vs quantization parameter for 'Race Horse' sequence

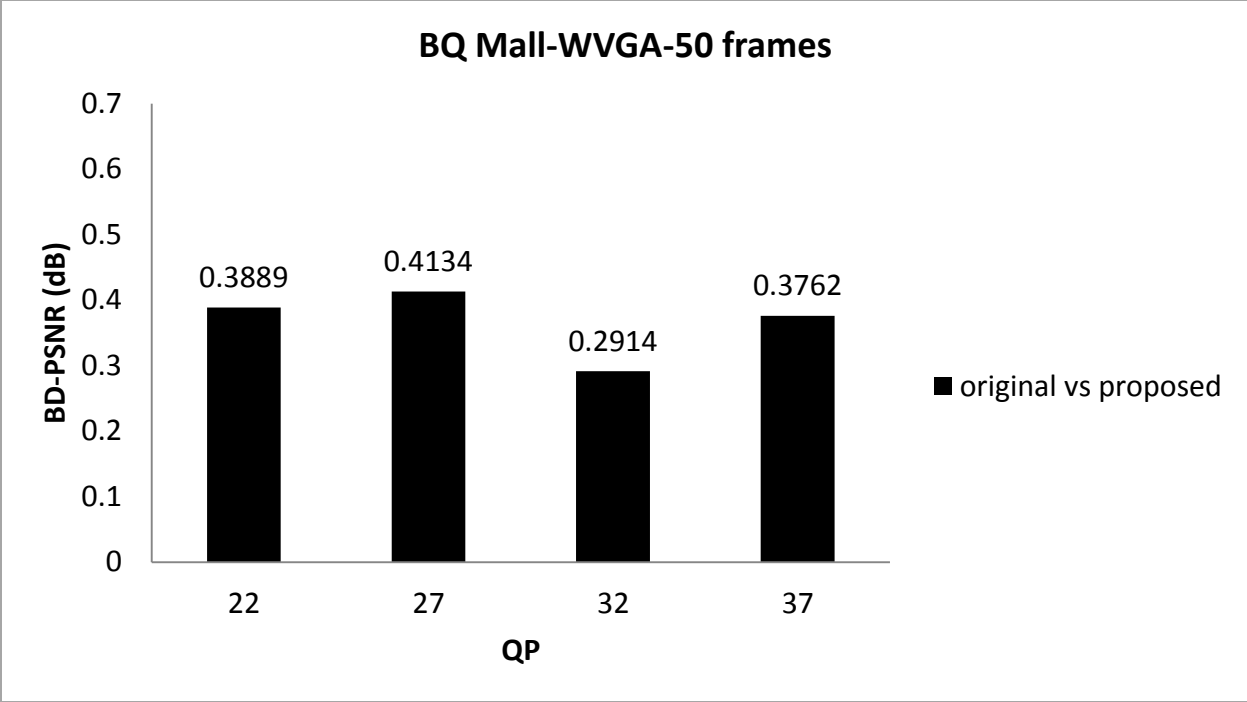


Figure 4.18 BD-PSNR vs quantization parameter for 'BQ Mall' sequence

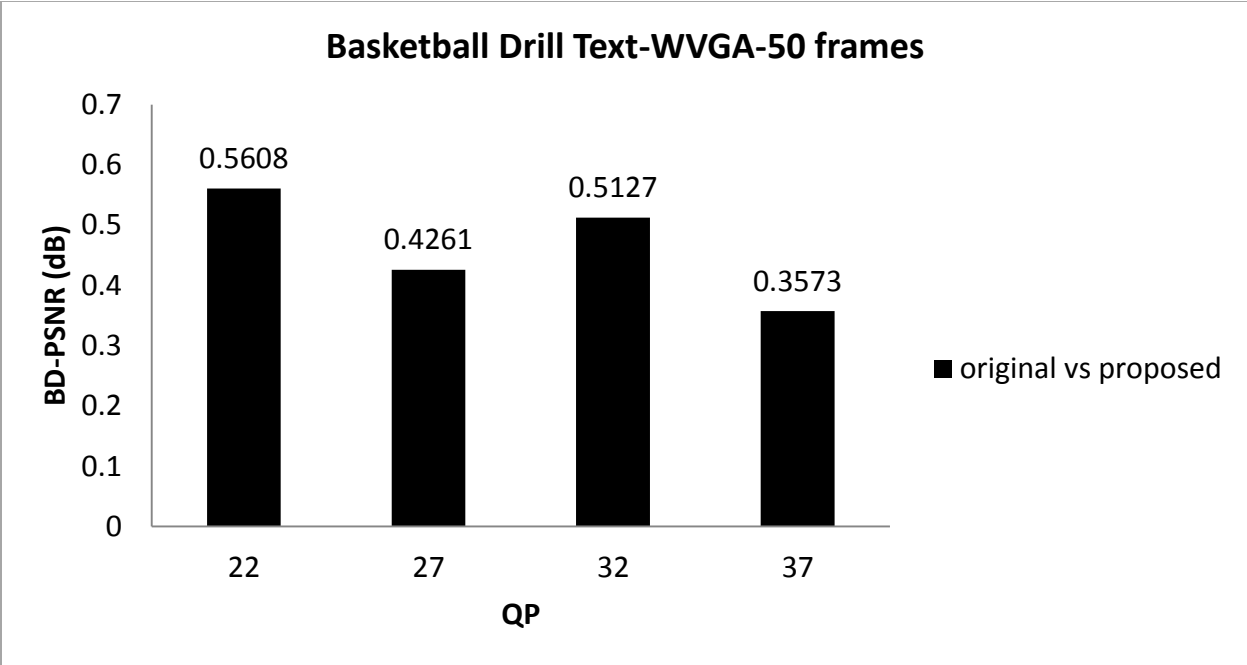


Figure 4.19 BD-PSNR vs quantization parameter for 'Basketball Drill Text' sequence

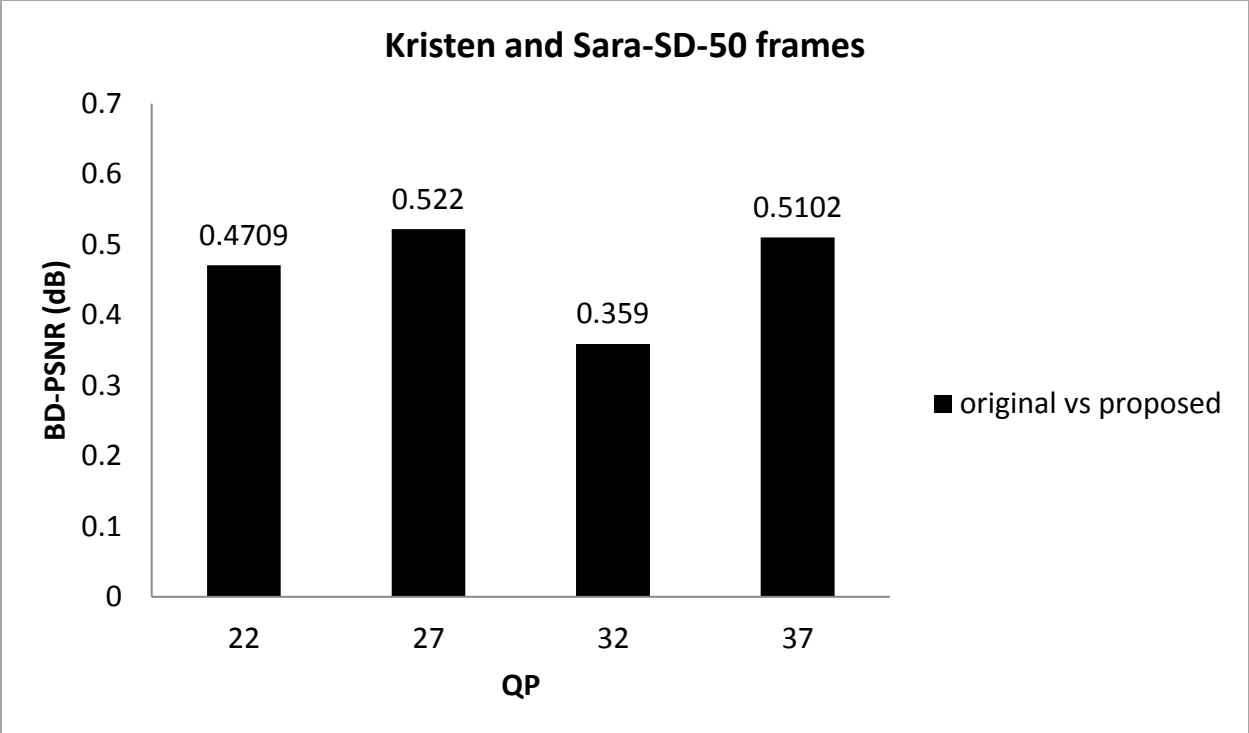


Figure 4.20 BD-PSNR vs quantization parameter for 'Kristen and Sara' sequence

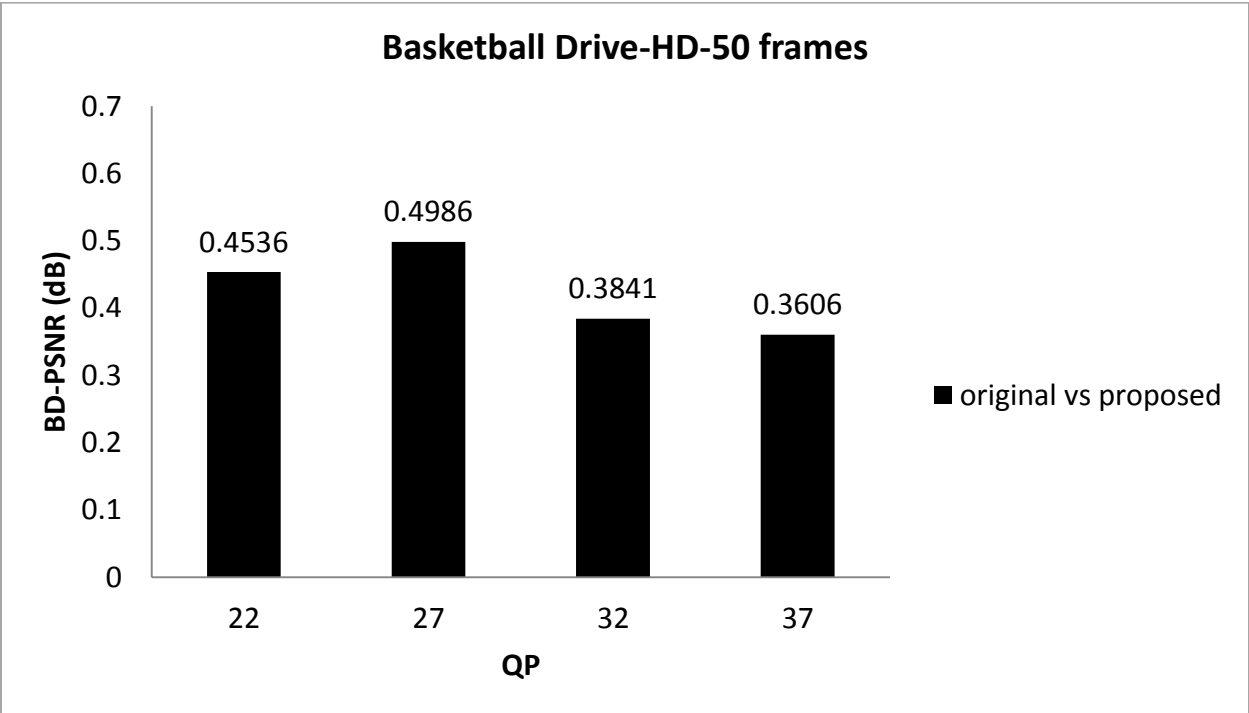


Figure 4.21 BD-PSNR vs quantization parameter for 'Basketball Drive' sequence

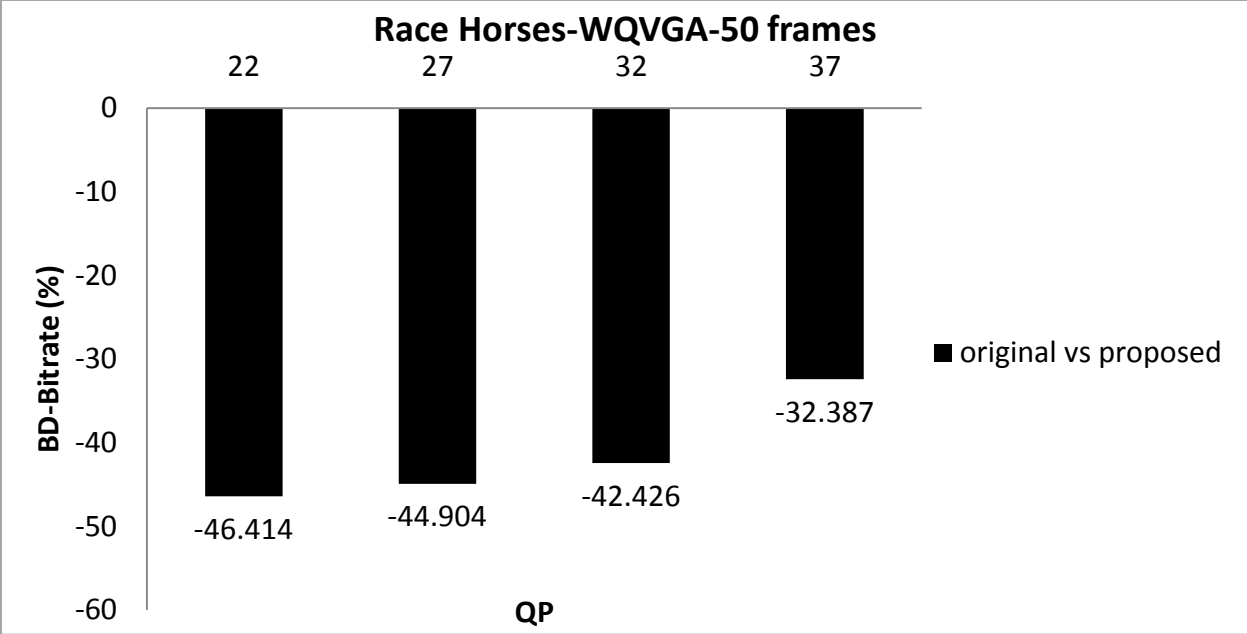


Figure 4.22 BD-bitrate vs quantization parameter for 'Race Horse' sequence

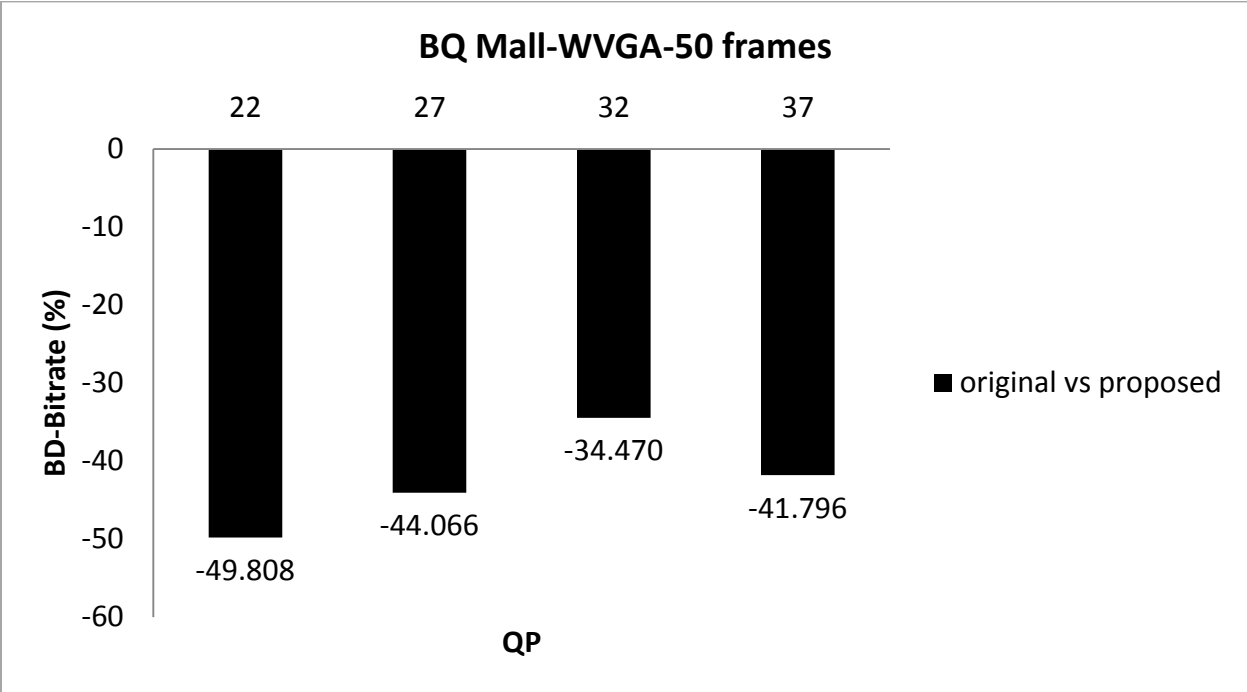


Figure 4.23 BD-bitrate vs quantization parameter for 'BQ Mall' sequence

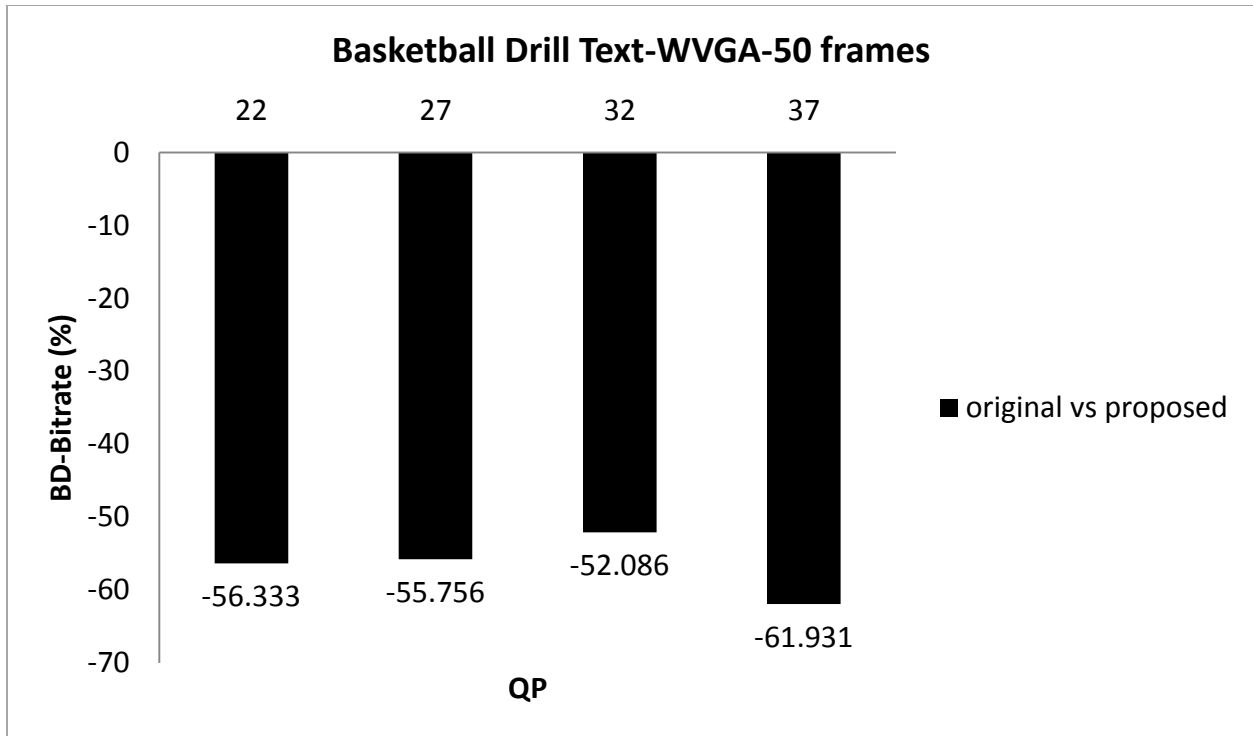


Figure 4.24 BD-bitrate vs quantization parameter for 'Basketball Drill Text' sequence

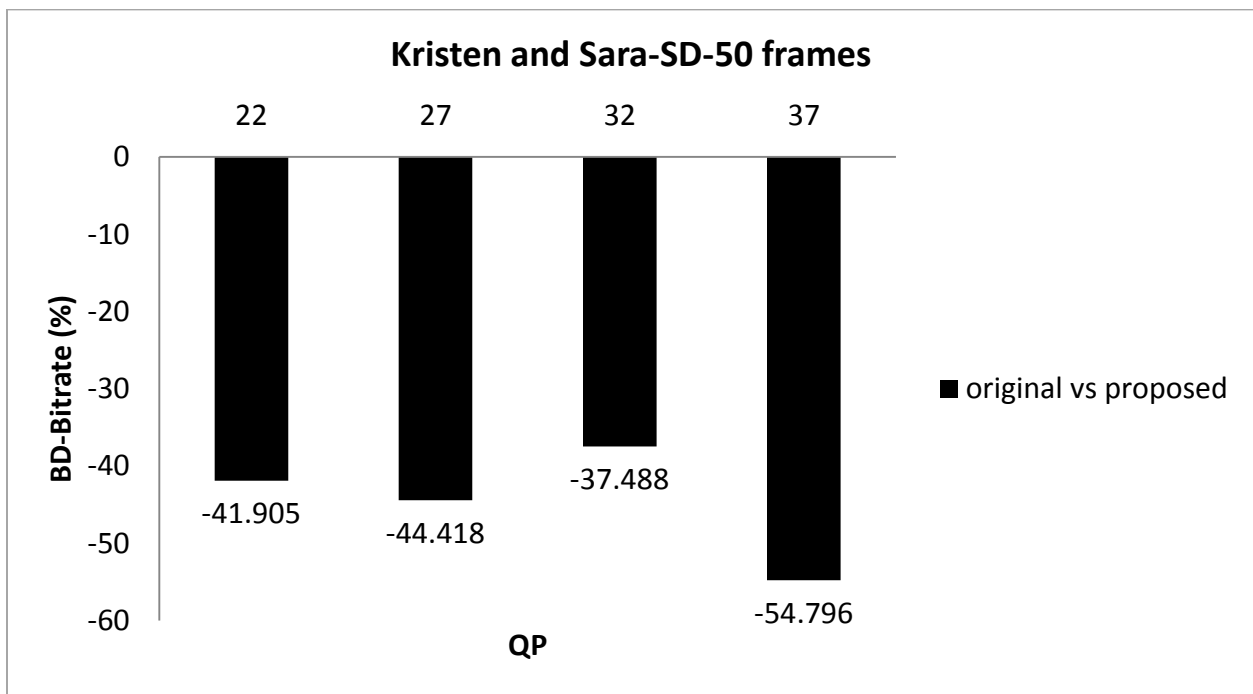


Figure 4.25 BD-bitrate vs quantization parameter for 'Kristen and Sara' sequence

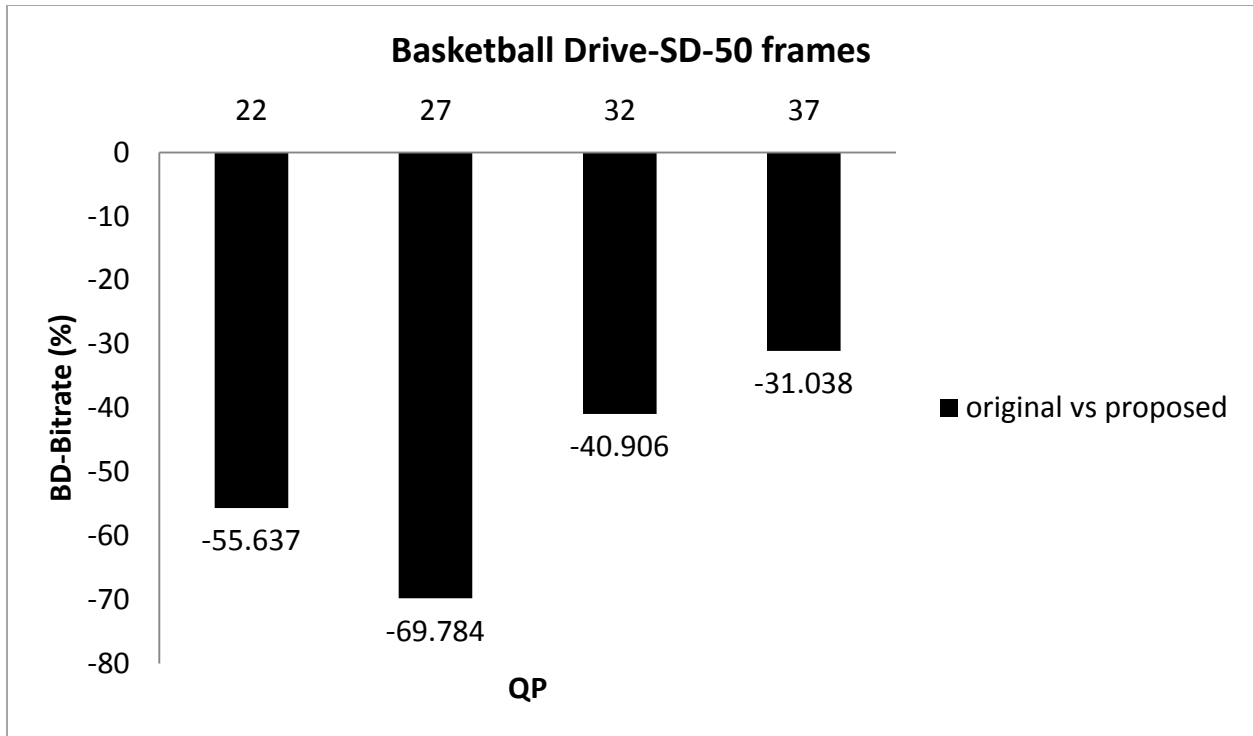


Figure 4.26 BD-bitrate vs quantization parameter for 'Basketball Drive' sequence

#### 4.6 Rate Distortion Plot (RD Plot)

The results described so far indicates a drop in bitrate and encoding time with a negligible loss in PSNR. The comparison of the impact original and the proposed algorithm on the bitrate and PSNR is shown in Figure 4.27 through Figure 4.31.

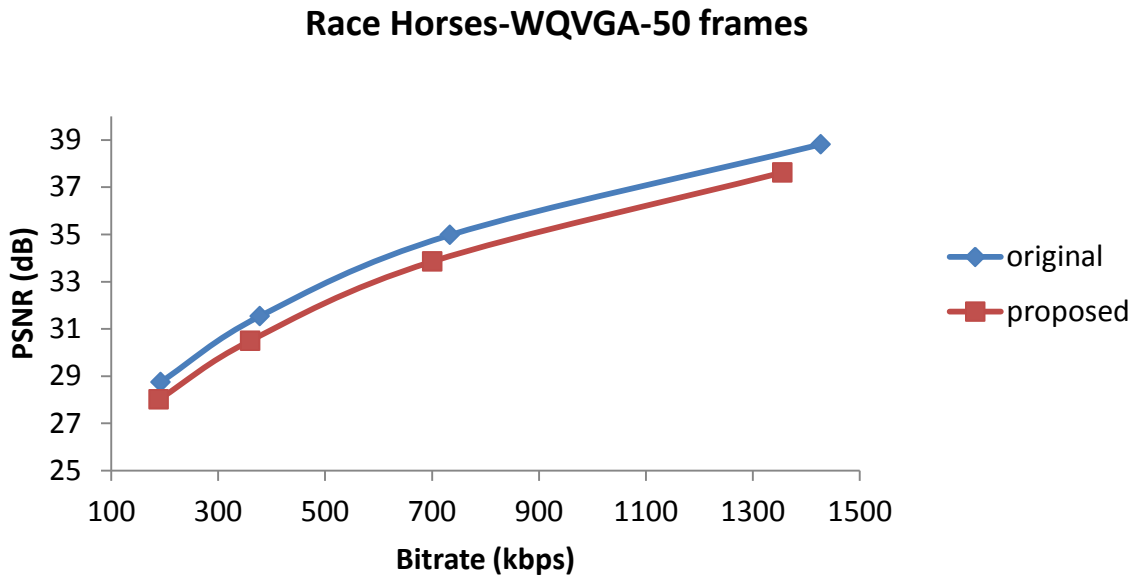


Figure 4.27 PSNR vs bitrate for 'Race Horse' sequence

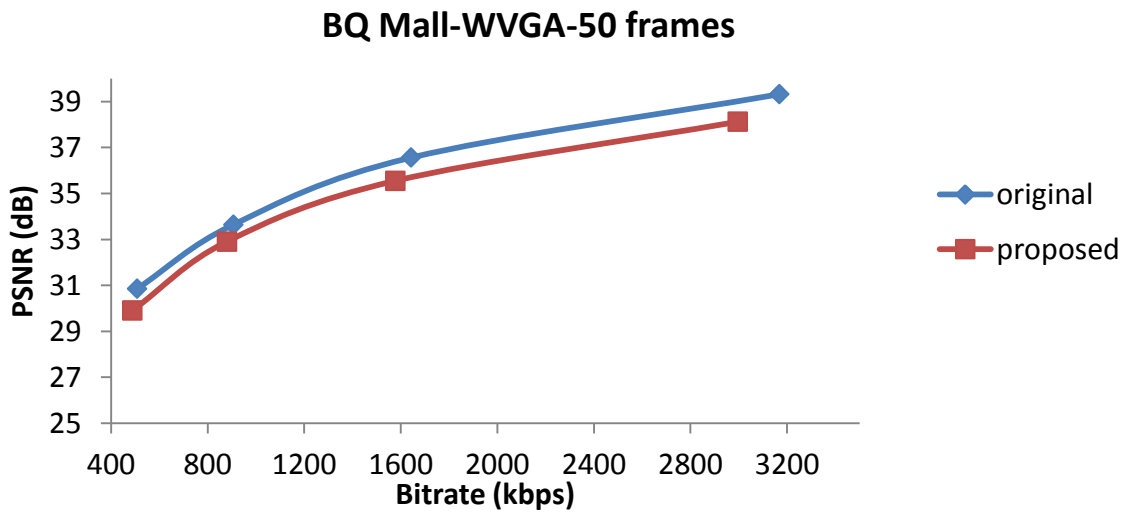


Figure 4.28 PSNR vs bitrate for 'BQ Mall' sequence



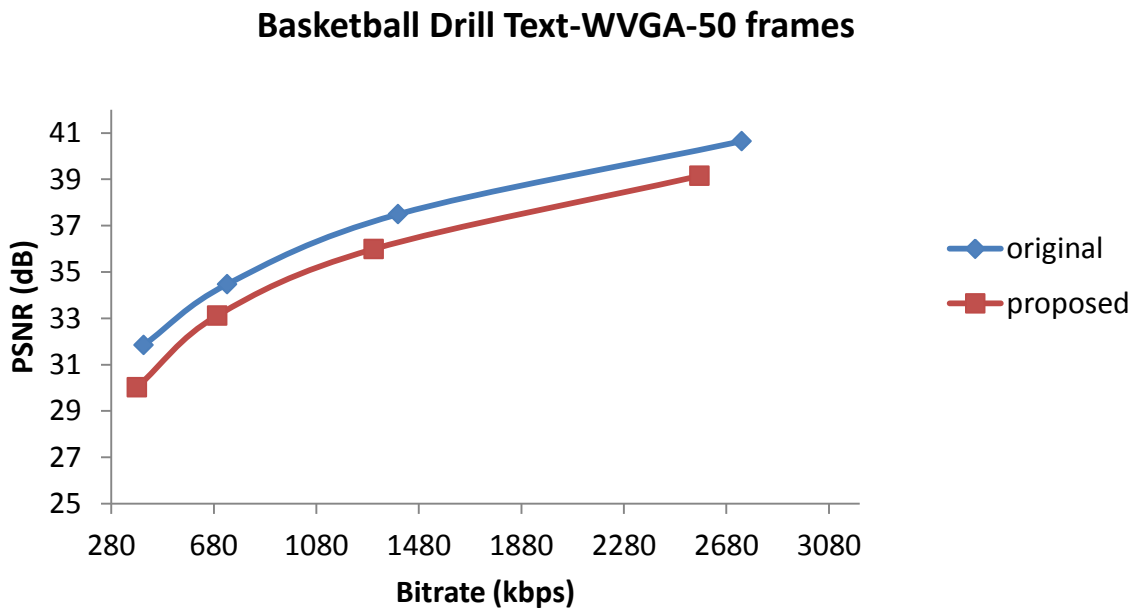


Figure 4.29 PSNR vs bitrate for "Basketball Drill Text" sequence

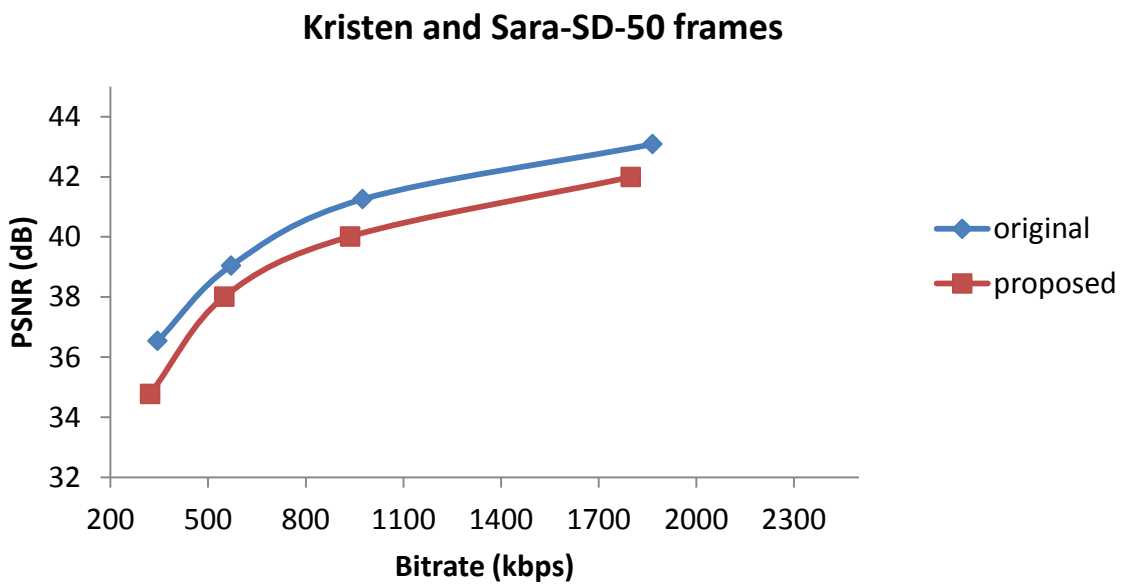


Figure 4.30 PSNR vs bitrate for 'Kristen and Sara' sequence

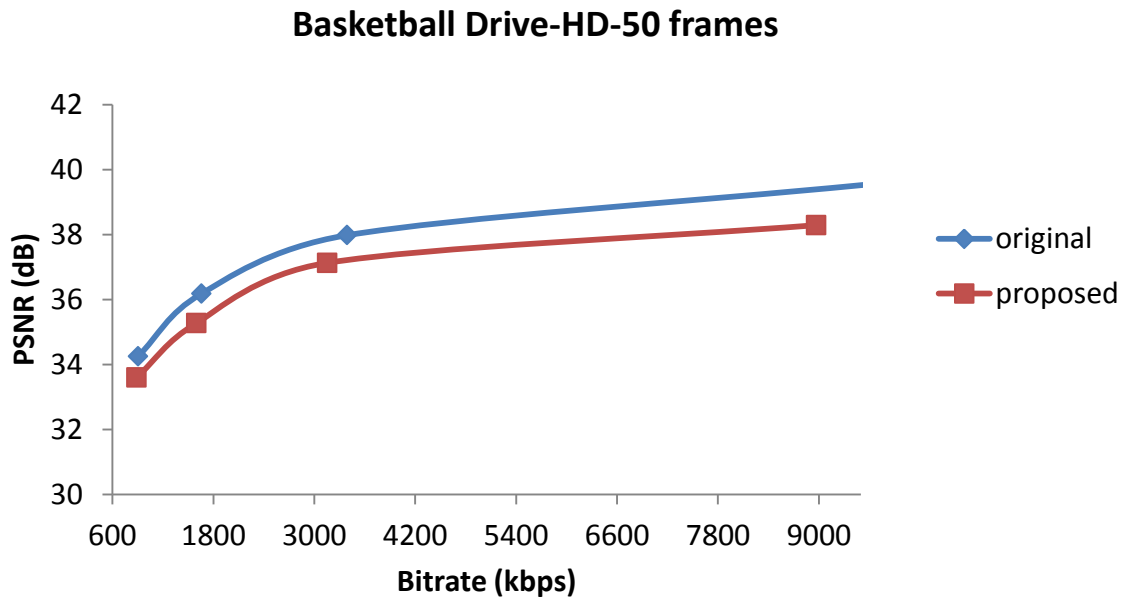


Figure 4.31 PSNR vs bitrate for 'Basketball Drive' sequence

#### 4.7 Summary

The chapter provided a quantitative comparison of the advantages and disadvantages of using the proposed algorithm in the HEVC standard against the unaltered HEVC reference software HM 13.0. This study was conducted using various metrics such as encoding time (sec), bitrate (kbps), PSNR (dB), BD-PSNR (dB), BD-bitrate (%) and PSNR vs bitrate all against different quantization parameters as suggested by the JCT-VC which aides in projecting the impact of the proposed algorithm on the bitrate reduction and encoding time reduction with a slight loss of quality. The chapter 5 will discuss on the conclusions drawn based on the study and areas of further improvements for future work.

## Chapter 5

### Conclusions and Future Work

#### 5.1 Conclusions

The latest video coding standard, High Efficiency Video Coding (HEVC) introduced in January 2013 by the Joint Collaborative Team on Video Coding (JCT-VC) has managed to achieve several advantages over the existing standards in terms of bit-rate reduction, ease of transport system integration, data loss resilience, increased video resolution and support for parallel processing architectures [11]. An extension of the HEVC standard is developed which supports encoding of increased bit depth videos, enhanced color component sampling, scalability and also 3-D/stereo/multi-view video coding [8]. However, the HEVC standard given its ability to accomplish all the mentioned improvements and enhancements is considered to be very complex in its encoding architecture and has areas which need complexity reduction. The thesis is a work on reducing the complexity of the HEVC encoder in the area of motion information management. The thesis proposes an algorithm for motion merging which reduces the time required for encoding and using the motion information by making use of redundancies in the data. The proposed algorithm shows a decrease in encoding time by 13-24%, reduction in bitrate by 2-7% with a slight loss of PSNR of 2-6% as opposed to the existing algorithm used in the HEVC reference software HM 13.0 [38]. The recently introduced and widely adopted metric BD-PSNR and BD-bitrate [40] which is used for comparing algorithms used in video codecs shows a positive values of BD-PSNR ranging from 0.29 to 0.56 and BD-bitrate of -31 % to -65% which indicates that the proposed algorithm has an improvement over the existing algorithm used in the unaltered HEVC reference software HM 13.0 [38].

## 5.2 Future Work

There are number of areas in inter/intra prediction and motion merging. The proposed algorithm makes use of sequential approach for its implementation, this can be made much faster by the use of parallel processing architectures which can lead to a significantly fast encoder with better use of computing resources. The proposed algorithm can be implemented along with the faster algorithms suggested for intra [42] and inter prediction [44] on a parallel processing architecture which can lead to better signal quality. The proposed work can also be implemented with other works on scalable extension of the HEVC [25] [45] which can lead to a faster and efficient video codec with applications on different platforms ranging from mobile devices to devices which are capable of 4K and more resolution.

Appendix A  
Test Sequences [46]

A.1 Race Horses



## A.2 BQ Mall



### A.3 Basketball Drill Text





#### A.4. Kristen and Sara



## A.5 Basketball Drive



Appendix B  
Test Conditions

The reference software used for this work is HM 13.0 [38]. The study was carried out on a Microsoft Windows 7 64-bit Operating system running on a 16 GB RAM at 3.70 GHz on an Intel Xeon CPU E5-1620 v2 processor.

## Appendix C

BD-PSNR and BD-Bitrate [40][41]

The Bjontegaard metric approved by ITU-T includes the BD-PSNR and BD-bitrate which are used in computing the average gain the PSNR and the average savings in bitrate between two rate-distortion graphs [39]. This method was developed by Bjontegaard and provides an accurate comparison between algorithms used in video codecs [40]. The MATLAB code is available online [41].

```
function avg_diff = bjontegaard2(R1,PSNR1,R2,PSNR2,mode)

%bjontegaard2  Bjontegaard metric calculation
% Bjontegaard's metric allows to compute the average gain in PSNR or the
% average per cent saving in bitrate between two rate-distortion
% curves [1].
% Differently from the avsnr software package or VCEG Excel [2] plugin this
% tool enables Bjontegaard's metric computation also with more than 4 RD
% points.
% Fixed integration interval in version 2.
%
% R1,PSNR1 - RD points for curve 1
% R2,PSNR2 - RD points for curve 2
% mode -
%   'dsnr' - average PSNR difference
%   'rate' - percentage of bitrate saving between data set 1 and
%           data set 2
%
% avg_diff - the calculated Bjontegaard metric ('dsnr' or 'rate')
%
% (c) 2010 Giuseppe Valenzise
%
%% Bugfix 20130515
% Original script contained error in calculation of integration interval.
% It was fixed according to description and figure 3 in original
% publication [1]. Script was verified using data presented in [3].
% Fixed lines labeled as "(fixed 20130515)"
%
% (c) 2013 Serge Matyunin
%%
%
% References:
%
% [1] G. Bjontegaard, Calculation of average PSNR differences between
%     RD-curves (VCEG-M33)
% [2] S. Pateux, J. Jung, An excel add-in for computing Bjontegaard metric and
```

```

%   its evolution
% [3] VCEG-M34. http://wftp3.itu.int/av-arch/video-site/0104\_Aus/VCEG-M34.xls
%
% convert rates in logarithmic units
IR1 = log(R1);
IR2 = log(R2);

switch lower(mode)
case 'dsnr'
    % PSNR method
    p1 = polyfit(IR1,PSNR1,3);
    p2 = polyfit(IR2,PSNR2,3);

    % integration interval (fixed 20130515)
    min_int = max([ min(IR1); min(IR2) ]);
    max_int = min([ max(IR1); max(IR2) ]);

    % find integral
    p_int1 = polyint(p1);
    p_int2 = polyint(p2);

    int1 = polyval(p_int1, max_int) - polyval(p_int1, min_int);
    int2 = polyval(p_int2, max_int) - polyval(p_int2, min_int);

    % find avg diff
    avg_diff = (int2-int1)/(max_int-min_int);

case 'rate'
    % rate method
    p1 = polyfit(PSNR1,IR1,3);
    p2 = polyfit(PSNR2,IR2,3);

    % integration interval (fixed 20130515)
    min_int = max([ min(PSNR1); min(PSNR2) ]);
    max_int = min([ max(PSNR1); max(PSNR2) ]);

    % find integral
    p_int1 = polyint(p1);
    p_int2 = polyint(p2);

    int1 = polyval(p_int1, max_int) - polyval(p_int1, min_int);
    int2 = polyval(p_int2, max_int) - polyval(p_int2, min_int);

    % find avg diff
    avg_exp_diff = (int2-int1)/(max_int-min_int);

```

```
    avg_diff = (exp(avg_exp_diff)-1)*100;  
end
```



## Appendix D

### Acronyms

AVC - Advanced Video Coding

AMVP – Advanced Motion Vector Prediction

BD - Bjontegaard Delta

CABAC – Context Adaptive Binary Arithmetic Coding

CB – Coding Block

CBF – Coding Block Flag

CFM – CBF Fast Mode

CTU – Coding Tree Unit

CTB – Coding Tree Block

CU – Coding Unit

DCT – Discrete Cosine Transform

DST – Discrete Sine Transform

HDTV - High Definition Tele Vision

HDR - High Dynamic Range

HDRI - High Dynamic Range Imaging

HEVC – High Efficiency Video Coding

HM – HEVC Test Model

HVS – Human Visual System

ISO – International Standards Organization

ITU – International Telecommunications Union

JCT-VC - Joint Collaborative Team on Video Coding

MB – Macroblock

MC – Motion Compensation

ME – Motion Estimation

MPEG – Moving Picture Experts Group

NAL – Network Abstraction Layer

PB – Prediction Block

PSNR – Peak Signal to Noise Ratio

PU – Prediction Unit

QP – Quantization Parameter

RDOQ – Rate Distortion Optimization Quantization

RGB – Red Green Blue

RMD – Rough Mode Decision

SATD – Sum of Absolute Transform Differences

SD – Standard Definition

SSIM – Structural Similarity

TB – Transform Block

TU – Transform Unit

URQ – Uniform Reconstruction Quantization

VCEG – Video Coding Experts Group

VPS – Video Parameter Set

WQVGA – Wide Quarter Video Graphics Array

WVGA – Wide Video Graphics Array

## References

- [1] K. Myszkowski, R. Mantiuk and G. Krawczyk, “High dynamic range video”, Synthesis Lectures on Computer Graphics and Animations, vol. 5, pp. 1-170, Jan. 2008.
- [2] K.R. Rao, D. N. Kim and J. J. Hwang, “Video coding standards: AVS China, H.264/MPEG-4 Part 10, HEVC, VP6, Dirac and VC-1”, Springer 2014.
- [3] A.C. Bovik, “Handbook of image and video processing”, Elsevier Academic Press, 2005.
- [4] I. Richardson, “Video codec design”, Wiley, 2002.
- [5] I. Richardson, “the H.264 advanced video compression standard”, Wiley, 2010.
- [6] K.R. Rao and J.J. Hwang, “Techniques and standards for image, video and audio coding”, Prentice Hall PTR, 1996.
- [7] O.E. Ping et al, “Perceptual quality and objective quality measurements of compressed videos”, Journal of Visual Communication and Image Representation, vol. 17, no. 4, pp. 717-737, August 2006.
- [8] N. Ling, “High efficiency video coding and its 3D extension: A research perspective,” 2012 7<sup>th</sup> IEEE conference on Industrial Electronics and Applications (ICIEA), pp. 2150-2155, Jul 2012.
- [9] Advanced Video Coding for Generic Audio-Visual Services, ITU-T Rec. H.264 and ISO/IEC 14496-10 (AVC), ITU-T and ISO/IEC JTC 1, May 2003 (and subsequent editions).
- [10] T. Wiegand et al, “Overview of the H.264/AVC video coding standard”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no.7, pp. 560-576, March 2004.

- [11] G. Sullivan et al, "Overview of the High Efficiency Video Coding (HEVC) standard", IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, pp. 1649-1668, Dec 2012.
- [12] M. Kitamura et al, "Beyond 4K: 8K 60p live video streaming to multiple sites", Future Generations Computer Systems, vol. 27, no. 7, pp. 952-959, July 2011.
- [13] K. Stamos, M. Dimitris and P. Georgios, "A GPU based real-time video compression method for video conferencing", IEEE International Conference on Digital Signal Processing, pp. 1-6, July 2013.
- [14] G. Sullivan et al, "Standardized Extension of HEVC", IEEE Journal of Selected topics in Signal Processing, Vol. 7, No. 6, pp. 1001-1016, December 2013.
- [15] C. Fogg, "Suggested figures for the HEVC specification", ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC) document JCTVC- J0292r1, July 2012.
- [16] Y. H. Tan, "On residual quad-tree coding in HEVC", IEEE International Workshop on Multimedia Signal Processing, pp. 1-4, October 2011.
- [17] Z. Wang, "Comments on 'a fast computational algorithm for the discrete sine transform'", IEEE Trans on Communications, vol. 34, no. 2, pp. 204-205, Feb. 1986.
- [18] M. Zhou, S. Vivienne and M. Budagavi, "Parallel tools in HEVC for high-throughput processing", SPIE Proceeding on Applications of Digital Image Processing, August 2012.f
- [19] B. Cauane, "Adaptive content-based Tile partitioning algorithm for the HEVC standard", IEEE Picture Coding Symposium, pp. 185-188, December 2013.
- [20] G. Bjontegaard et al, "HEVC deblocking filter", IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, pp. 1755-1764, October 2012.

- [21] C. Fu et al, "Sample adaptive offset filter in the HEVC standard", IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, pp. 1746-1754, October 2012.
- [22] H. Le and J. Bae, "High-throughput parallel architecture for H.265/HEVC deblocking filter", Journal of Information Science and Engineering, vol. 30, no. 2, pp. 281-294, March 2014.
- [23] W. Kim and D. Kwon, "Improved sample adaptive offset for HEVC", IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 1700-1703, May 2013.
- [24] R. Snjezana, O. Nemcic and V. Mario, "Scalable video coding extension of the H.264/AVC standard", International Symposium Electronics Proceedings, vol. 1. Pp. 9-12, 2008.
- [25] P. Helle et al, "A scalable video coding extension of HEVC", IEEE Data Compression Conference, pp. 201-210, March. 2013.
- [26] H. Tobia et al, "An HEVC extension for spatial and quality scalable video coding", SPIE Proceedings on Visual Information Processing and Communication, vol. 8666, 2013.
- [27] T. M. Bae, T. C. Thang and M. Yong, "Improvement of inter-layer motion prediction in scalable video coding", IEICE Transactions on Information and Systems, vol. E90, no. 10, pp. 1712-1715, October 2007.
- [28] C. Park et al, "Selective inter-layer residual prediction for SVC-based video streaming", IEEE Transactions on Consumer Electronics, Vol. 55, pp. 245-239, February 2009.
- [29] S. Rickard et al, "Overview of HEVC high-level syntax and reference picture management", IEEE Transactions on Circuits and Systems for Video Technology, vo. 22, no. 12, pp. 1878-1870, October 2012.

- [30] Z. Liang et al, "Simplified AMVP for high efficiency video coding", IEEE Visual Communications and Image Processing, pp. 1-4, November 2012.
- [31] M. Siwei et al, "Low complexity rate distortion optimization for HEVC", Data Compression Conference Proceedings, pp. 73-82, March 2013.
- [32] M. Cassa, M. Naccari and F. Pereira, "Fast rate distortion optimization for the emerging HEVC standard", IEEE Picture Coding Symposium, pp. 493-496, May 2012.
- [33] K. Ugur et al, "Motion compensated prediction and interpolation filter design in H.265/HEVC", IEEE Journal on Selected topics in Signal Processing, vol. 7, no. 6, pp. 946-956, December 2013.
- [34] K. Ugur et al, "Interpolation filter design in HEVC and its coding efficiency – Complexity analysis", IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 1704-1708, October 2013.
- [35] G. Sullivan et al, "High efficiency video coding (HEVC) test model 13 (HM13) encoder description", JCT-VC & Video, Geneva, Switzerland, November 2013.
- [36] P. Helle et al, "Block merging for quadtree-based video coding", IEEE Multimedia and Expo conference, pp. 1-6, July 2011.
- [37] A.C. Bovik and M.A. Saad, "Natural motion statistics for no-reference video quality assessment", IEEE International Workshop on Quality of Multimedia Experience, pp. 163-167, July 2009.
- [38] HEVC reference software HM 13.0 [online]. Available. <http://hevc.kw.bbc.co.uk/svn/jctvc-a124/branches/>
- [39] JCT-VC documents [online]. Available. <http://www.itu.int/en/ITU-T/studygroups/2013-2016/16/Pages/video/jctvc.aspx>

- [40] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves", Q6/SG16, Video Coding Experts Group (VCEG), April 2001.
- [41] BD metric code [online]. Available. <http://www.mathworks.com/matlabcentral/fileexchange/27798-bjontegaardmetric/content/bjontegaard.m>
- [42] MPL website [online]. Available. <http://www-ee.uta.edu/dip/>
- [43] D.P. Kumar, "Intra frame luma prediction using neural networks in HEVC", website: [http://www-ee.uta.edu/Dip/Courses/EE5359/Dilip\\_Thesis\\_Document.pdf](http://www-ee.uta.edu/Dip/Courses/EE5359/Dilip_Thesis_Document.pdf), Thesis, University of Texas at Arlington, UMI Dissertation Publishing, May 2013.
- [44] Kushal Shah, "Reducing the complexity of inter-prediction mode decision for HEVC", website: [http://www-ee.uta.edu/Dip/Courses/EE5359/KushalShah\\_Thesis.pdf](http://www-ee.uta.edu/Dip/Courses/EE5359/KushalShah_Thesis.pdf), Thesis, University of Texas at Arlington, UMI Dissertation Publishing, April 2014.
- [45] Karuna Gubbi, "Complexity reduction of interlayer inter prediction in Scalable High Efficiency Video Coding", website: [http://www-ee.uta.edu/Dip/Courses/EE5359/KarunaGubbi\\_Thesis.pdf](http://www-ee.uta.edu/Dip/Courses/EE5359/KarunaGubbi_Thesis.pdf), Thesis, University of Texas at Arlington, UMI Dissertation Publishing, July 2014.
- [46] HEVC test sequences [online]. Available. <ftp://ftp.tnt.uni-hannover.de/testsequences>



### Biographical Information

Abhishek Hassan Thungaraj was born in Bangalore, Karnataka, India in 1990. After completing his schooling at United Academy, Hassan in 2006, he went to obtain his bachelor's degree in engineering in Medical Electronics from Dayanand Sagar College of Engineering in Bangalore in 2012.

He joined the University of Texas at Arlington to pursue his master's degree in Electrical Engineering in 2012. This was around the time he joined the Multimedia Processing Lab.