# COMPLEXITY REDUCTION IN INTER LAYER INTER PREDICTION IN SCALABLE HIGH EFFICIENCY VIDEO CODING

by

KARUNA GUBBI SHIVASHANKAR SASTRI

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

Acknowledgements

First and foremost, I would like to thank Dr.K.R. Rao for being a guide and mentor and a constant source of encouragement throughout my thesis. I would like to thank Dr. M.T. Manry and Dr.H. Russell for serving on my committee.

I would like to thank Gary J. Sullivan, video and image technology architect at Microsoft Corporations and Shevach Riabtsev, Software Engineer, Algorithmists at Beamr for promptly replying to my querues.

I would like to thank my MPL lab-mates: Abhishek Hassan Thungaraj, Mrudula Warrier, Jayesh Dubhashi, Tuan Ho and Shiba Kuanar for providing valuable inputs throughout my research.

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

July 15, 2014

Abstract

COMPLEXITY REDUCTION IN INTER LAYER INTER PREDICTION IN

SCALABLE HIGH EFFICIENCY VIDEO CODING

Karuna Gubbi Shivashankar Sastri, M.S.

The University of Texas at Arlington, 2014

Supervising Professor: K.R. Rao:

After completion of High Efficiency Video Coding standardization the relevant international standardization committees have shifted their focus toward the development of several key extensions of its capabilities to address the needs of broader range of applications [25]. The extensions under current development fall into three categories: 1) the range extensions, which expand the range of bit depths and color sampling formats supported by the standard and include an increased emphasis on screen-content coding; 2) the scalability extensions, which enable the use of embedded bitstream subsets as reduced-bit-rate representations of the video content; and 3) the 3D video extensions, which enable stereoscopic and multiview representations.

In scalable extension of HEVC, the base layer is coded using normal HEVC codec and the enhancement layers are coded using interlayer prediction for both intra and inter predictions. The prediction process for the enhancement layer in scalable video coding is complex and the proposed algorithm in the thesis concentrates mainly on reducing the complexity of the inter-layer inter prediction to code the enhancement layer.

Table of Contents

List of illustrations

# Chapter 1

# Introduction

## 1.1 Basics of Video Compression

As a result of revolution in the ‘digital age’ we are surrounded with devices capable of capturing, processing, transmitting, receiving and displaying videos and multimedia files [1][2]. The devices vary from low processing handheld devices to powerful high-end devices. The video captured or created with any device either needs to be stored to watch in the future or transmitted to share it with other devices and hence processing of the digital video plays a very important role in our day to day life which relies on these technologies.

Video is a collection of images and Image is collection of pixels where each pixel comprises of brightness and color components [2]. The collection of pixels in its native state occupies huge space for its storage and high bandwidth for its transmission and hence compression of the video plays a very important role. The compression must be such that it reduces the size of the original video without degrading its quality. This compression can be achieved by exploiting the spatial and temporal redundancies present in the original/raw video file.

The sequence of images which make up a video are called as frames. In video streaming there are two types of frames depending on the how they are encoded [2]. The first type is called the I-frame or Intra frame is encoded/ decoded by exploiting the spatial redundancies present within the frame. The other type is called Inter frames which make use of temporal redundancies that usually exist with other frames which are termed as their reference frames. Inter frames are of two types P-frames and B-frames. A P-frame is encoded/ decoded by taking a past frame as its reference to exploit the temporal redundancy. A B-frame or a bi-directional frame is encoded/ decoded by taking a past

and a future frame as its reference. The sequence of frames used in encoding/decoding is called as ‘Group of Pictures’ (GOP). The selection of GOP for a particular user application depends on factors like the availability of bandwidth to transmit the data, storage space, time constraints to encode/decode etc. Figure 1-1 shows the use of different reference frames for compression of I, P and B frames [36].

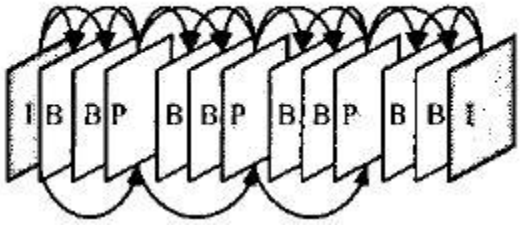



Figure 1-1: I, P and B-frame compression relative to the respective reference frames [36].

## 1.2 Video Compression Standards

The first video coding standard H.120 was developed by ITU-T (VCEG) (International Telecommunication Union-Telecommunications (Video Coding Experts Group)) in the year 1984. This was followed by H.261 standard in the year 1990 [6]. MPEG-1 (Moving Picture Experts Group-1) standard was developed by ISO (International Standardization Organization) and IEC (International Electro-technical Committee) in the year 1988 [6].

ITU-T (VCEG) and ISO/IEC (MPEG) jointly developed MPEG-2/H.262 [4] standard in the year 1993. ITU-T developed H.263 [3] in the year 1995 which is a huge step for the progressive video coding standards. Around the same time MPEG-4 introduced many new concepts such as 3D graphics, interactive graphics etc. H.264/MPEG-4 part 10/AVC (Advanced Video Codec) was developed to improve the compression ratio [6][7].

ITU-T (VCEG) and ISO/IEC (MPEG) collaborated to form JCT-VC (Joint Collaborative Team on Video Coding) in 2010 and developed the most recent video coding standard HEVC (High Efficiency Video Coding), the standard was finalized in the year 2013 [8]. Figure 1-2 shows the evolution of the video coding standards [9].

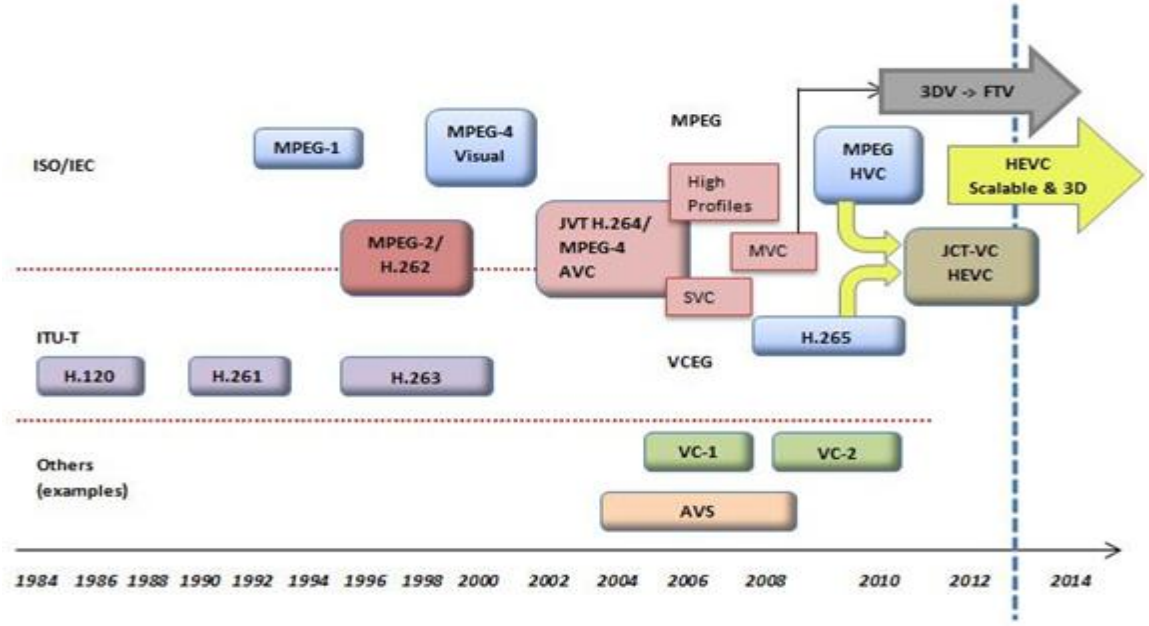



Figure 1-2: Evolution of the video coding standards [9].

## 1.3 Thesis Outline

Chapter 2 illustrates a brief introduction to HEVC standard. Chapter 3 will provide details about scalabale video coding in HEVC and the proposed algorithm to reduce the complexity for inter-layer inter prediction in SHVC. Chapter 4 describes the experimental conditions, its results and conclusions. Chapter 6 presents the possible future research.

# Chapter 2

# High Efficiency Video Coding (HEVC)

## 2.1 Introduction

High Efficiency Video Coding is the most recent video coding standard developed by JCT-VC (Joint Collaborative Team on Video Coding) [8]. Its predecessor H.264/MPEG-4 AVC developed in 2003 was used in many application domains including broadcast of high definition TV signals over cables, satellites, video content acquisition and editing systems, security applications etc., and it replaced all the other previous video compression standards [14]. In 2010 JCT-VC started to develop HEVC to address increasing diversity of services, emergence of higher resolution videos beyond HD formats (e.g. 4k x 2k or 8k x 4k resolution) [50]. HEVC is designed mainly to achieve higher compression, almost 50% bit-rate is reduced compared to H.264/AVC at the same visual quality and this is done by increasing the use of parallel processing architecture to address increased video resolution formats. Other goals of HEVC are to provide ease of transport system integration and data loss resilience.

The video coding layer (VCL) of HEVC uses the traditional block-based hybrid approach i.e. inter-/intra-picture prediction followed by 2-D transform coding. Figure 2-1 shows the encoder block diagram of HEVC [8]. The HEVC bitstream is produced by the encoding algorithm as follows: The picture to be encoded is partitioned into block-shaped regions. The first frame of the video sequence is encoded with the help of intra-picture prediction alone. The pictures between the random access points or the remaining pictures of sequence are coded using inter-picture prediction mode exploiting the temporal redundancies. Interpicture prediction comprises of choosing motion data, which consists of motion vector (MV) and selected reference picture to be used for predicting each block's samples. Mode decision data and motion vectors are transmitted to the

decoder as side data and using this information both encoder and decoder produce identical interpicture prediction signals. The difference between the original block and its predicted block is called residual signal and is generated for both intra and inter-prediction blocks. The residual signal is transformed by linear spatial transform, the resulting transform coefficients are scaled, quantized, entropy coded and finally transmitted along with the prediction information.

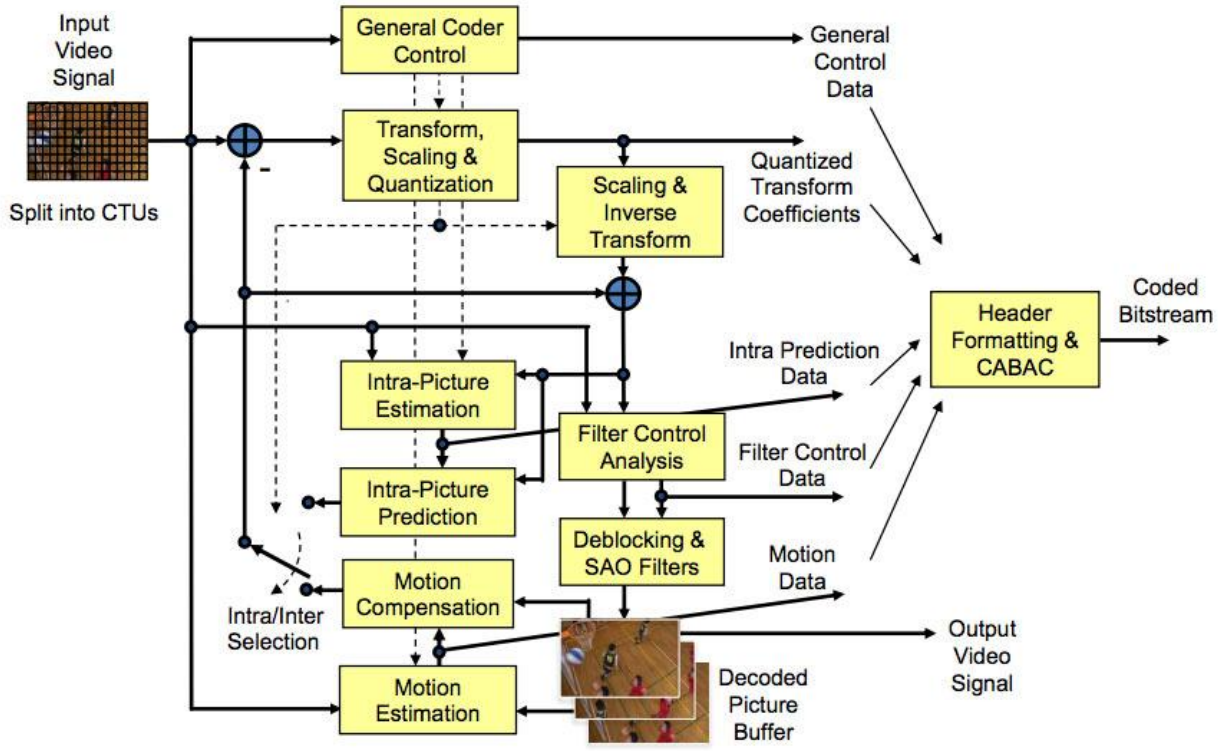



Figure 2-1: HEVC encoder block diagram [8].

The encoder consists of decoder in it such that both will generate exact predictions for the subsequent data. Hence the quantized transform coefficients are inverse scaled and inverse transformed to mimic the residual signal generated at the

decoder. This generated residual signal is added to the predicted signal and the result is smoothened by filtering. The final picture obtained is maintained in the decoded picture buffer to use it for further prediction of pictures. Figure 2-2 shows the decoder block diagram of HEVC [10].

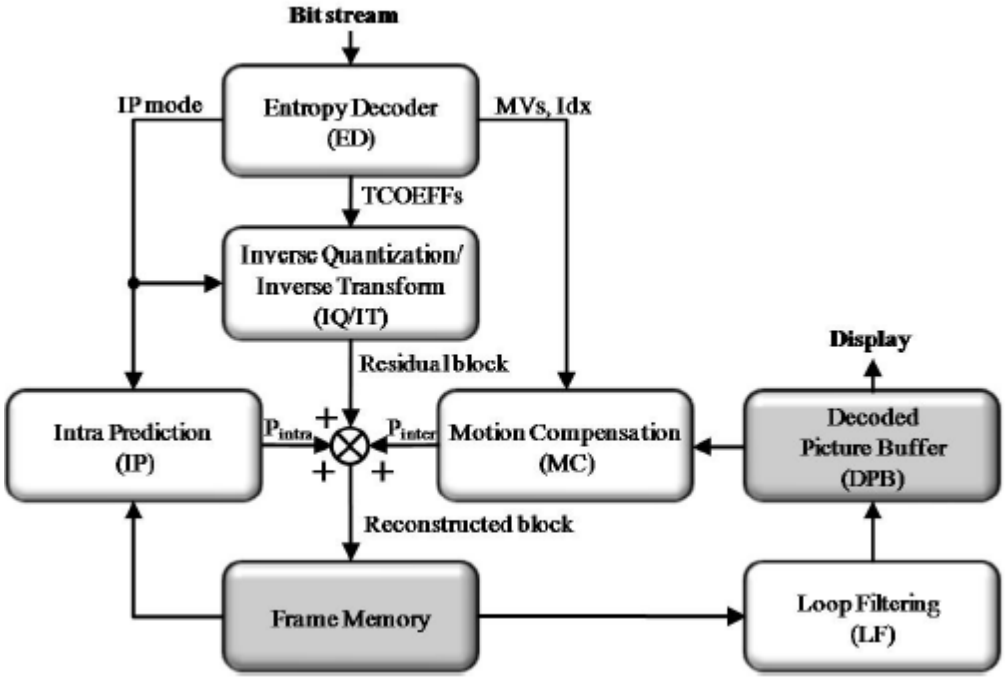



Figure 2-3: HEVC decoder block diagram [10].

## 2.2 Important Features of Video Coding Layer (VCL) [8]

### *2.2.1 Coding Tree Units and Coding Tree Blocks*

The coding layer's main essence in HEVC is Coding Tree Unit (CTU) which can be encoded in selectable size, the maximum size of CTU is 64x64 and is called the largest coding unit (LCU) [8][11]. The CTU is made of coding tree block (CTB)s and syntax elements, where CTB consists of luma CTB and corresponding chroma CTBs.

The size of luma CTB can be 64x64, 32x32, 16x16, higher the size of the CTB greater is the compression achieved..

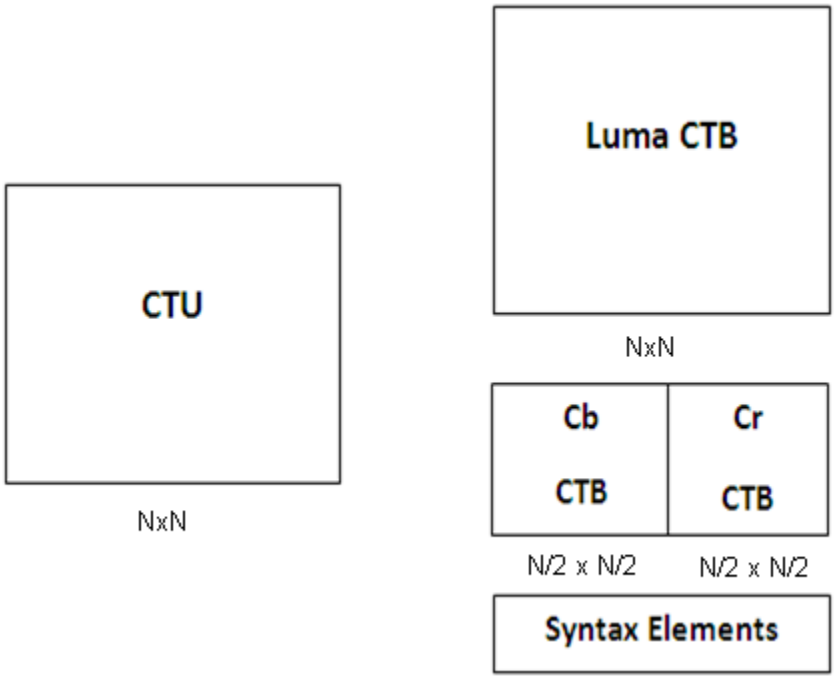



Figure 2-4: CTU composition in HEVC [12].

*2.2.2 Coding Unit and Coding Block*

HEVC supports quad-tree structure in the division of CTUs into coding blocks (CB) and it specifies the position and size of its CBs, hence the root of the quad-tree is the CTU [8]. Figure 2-5 shows the quad-tree division of CTU into corresponding CBs [13]. Coding unit (CU) consists of one luma CB and two chroma CBs and the syntax element associated with it. Luma CB can have the maximum size of luma CTB. CTB can have one or multiple coding units associated with it.

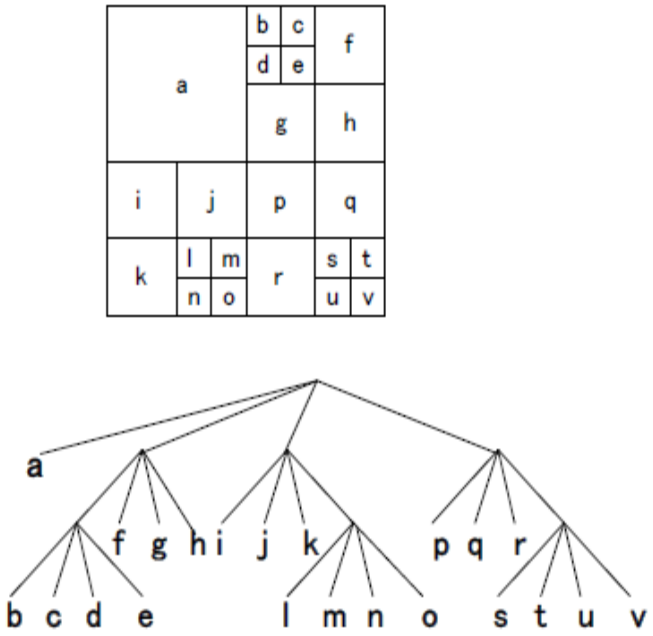



Figure 2-5: Quad-tree division of CTU into CBs [13].

*2.2.3 Prediction Units*

Coding unit is further partitioned into smaller units called prediction units (PUs) which form the base for prediction [11]. Each CU may contain one or multiple PUs and the size of PU varies from 4x4 to as large as the size of its root luma CU. PUs are of two types viz. symmetric and asymmetric. Symmetric PUs are either square or rectangular in shape and are used in both intra and inter-predictions. Asymmetric PUs are used only in inter-prediction. Figure 2-6 shows the CU partitioning into symmetric and asymmetric PUs. Figure 2-6 shows the different possible partitions of CB into prediction blocks (PBs) [12].

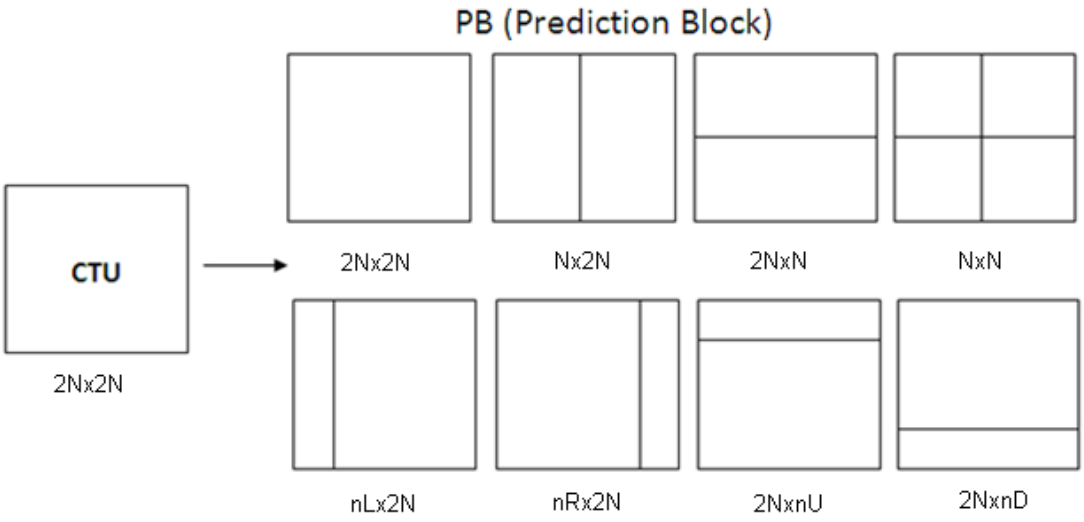



Figure 2-6: Possible partition of CB into PBs [12].

*2.2.4 Transform Units*

Transform unit (TU) forms the basic unit for transformation and quantization process in HEVC [11]. In HEVC integer discrete cosine transform (DCT) is used to de-correlate the data present in the residuals. TU size depends on its root PU size and may vary from 4x4 to 32x32 for square TUs and non-square TUs can have sizes such as 32x8, 8x32, 16x4 and 4x16. CU might consist of one or more TUs and quad-tree segmentation is used to partition each square CU into smaller TUs. Figure 2-7 shows the partition of CB into different possible TBs [12]. HEVC uses a version of 4x4 discrete sine transform (DST) for blocks coded with some directional intraprediction modes.

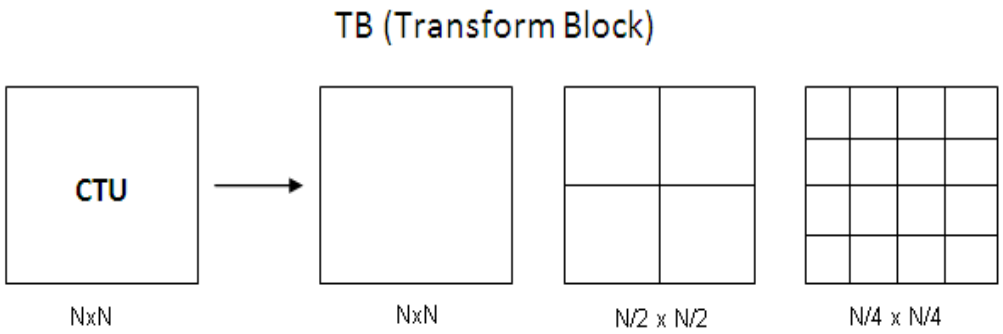



Figure 2-7: Possible partitions of CB into TBs [12].

*2.2.5 Motion Vector Signaling*

Advanced motion vector prediction (AMVP) is used, which includes derivation of several most probable candidates based on information from adjacent PBs and the reference picture [8]. Merge mode for motion vector (MV) coding can also be used. This allows the inheritance of MVs from spatial or temporal neighboring PBs. Skipped and direct motion inference is also specified here.

*2.2.6 Motion Compensation*

Quarter sample precision is used for MVs, and 7-tap (weights: -1, 4, -10, 58, 17, -5, 1) or 8-tap (weights: -1, 4, -11, 40, 40, -11, 4, 1) filters are used for fractional samples' interpolation [8]. To encode a particular picture, multiple reference pictures can be used. For each PB, either one or two MVs can be transmitted which results in unipredictive or bipredictive coding respectively. Figure 2-8 shows the motion compensation of the current picture using multiple reference pictures [14].

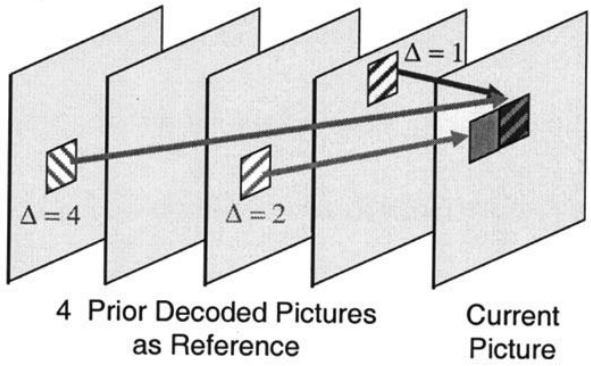



Figure 2-8: Motion compensation of the current picture using multiple reference pictures [14].

*2.2.7 Entropy Coding*

Entropy coding is applied to code all the syntax elements and quantized transform coefficients after transformation [11]. In HEVC context adaptive binary arithmetic coding (CABAC) is used for entropy coding. CABAC's arithmetic coding engine and more sophisticated context modeling result in better coding efficiency than CAVLC. CABAC increases coding complexity whilst it improves the coding efficiency. This is more pronounced at higher bit rates [small quantization parameters (QPs)], where transform coefficient data have a dominant role in the encoded bit streams. HEVC uses higher throughput alternative mode for coding transform coefficient data, to improve the worst case throughput. Figure 2-9 shows the block diagram of HEVC entropy coding [11].

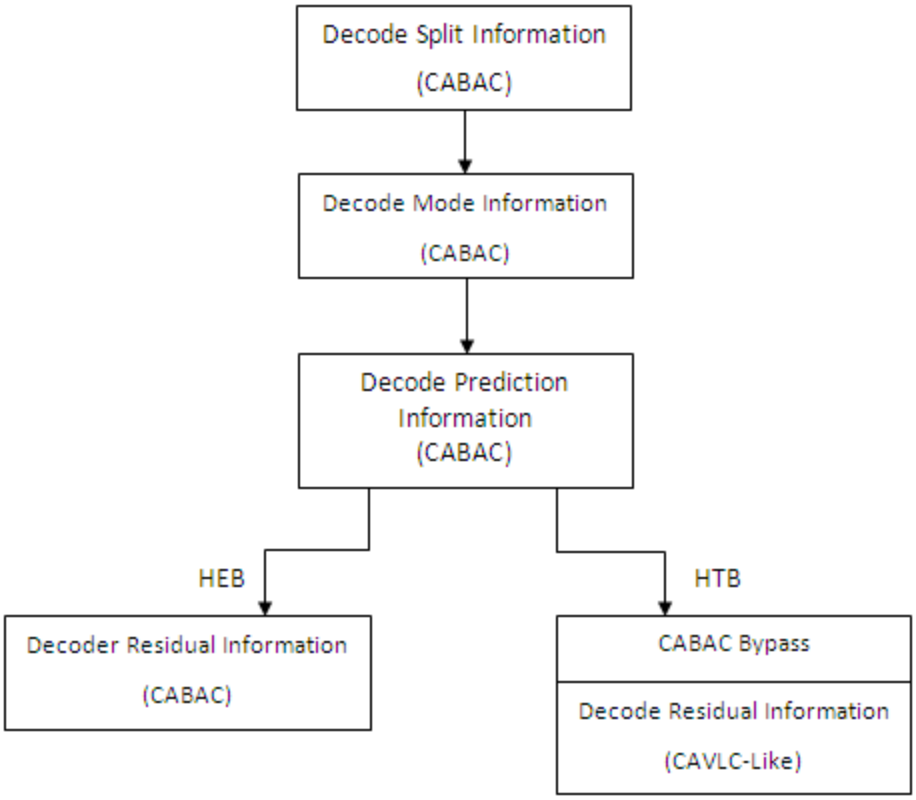



Figure 2-9: HEVC entropy coding block diagram [11].

In HEVC there are two modes of entropy coding: high efficiency binarization (HEB) and high throughput binarization (HTB). The HEB mode is completely CABAC based and HTB mode is partially based on CAVLC residual coding module. HTB is designed to serve as the high-throughput mode of HEVC and its use is signaled at slice level. In HTB mode, all syntax elements except the residual coefficients are coded using CABAC and CAVLC is used in coding the residual coefficients. Therefore HEVC entropy coding uses the best features of both CABAC and CAVLC coding i.e. high efficiency and low complexity respectively. .

### *2.2.8 In-Loop Deblocking Filter*

Blocking is one of the objectionable and most visible artifacts of block-based compression methods [11]. HEVC uses an in-loop deblocking filter to reduce the blocking. In HEVC there are many kinds of block boundaries such as CUs, PUs and TUs. The set of boundaries that may be filtered in HEVC is the union of all these boundaries except 4x4 blocks, they are not filtered to reduce the complexity. Decision is made for each boundary as whether to turn the deblocking filter on or off and also to check whether to apply strong or weak filtering. And this decision is based on the pixel gradient across the boundary and thresholds derived based on the QP in the blocks.

### *2.2.9 Sample Adaptive Offset*

Sample adaptive offset (SAO) is a new coding tool introduced in HEVC [11]. SAO involves sorting of pixels into different categories and adding a simple offset value to each pixel based on its category. The classification of the reconstructed pixels is based on either intensity or edge properties. It then adds an offset, either band offset (BO) or edge offset (EO), to the pixels in every category in a region to lower the distortion.

BO categorizes all pixels of a region into multiple bands with each band has pixels in the same intensity interval. There are 32 intervals in the intensity range which vary from zero to maximum intensity. The 32 intervals are divided into two groups, one consists of the central 16 bands and the other group consists of the rest of the 16 bands. Figure 2-10 shows the intensity bands and groups of bands in BO mode [11]. The encoder decides which group of bands to apply SAO, therefore the encoded bitstream will have 16 offset encoded in them.


Figure 2-10: Group of intensity bands in band offset (BO) mode [11].

There are four one-dimensional three pixel patterns; EO uses one of the four 1-D patterns to classify pixels based on their edge direction. Figure 2-11 shows the 1-D three pixel patterns [11]. If the value of the pixel is greater than the two neighbors, it is classified as peak; if it is less than the two neighbors, it is called valley; if it is equal to one neighbor it is called edge or none of the above three.

The encoder chooses BO or EO to apply to different regions of the picture and it can also signal that both EO and BO are not used in a particular region of the picture.

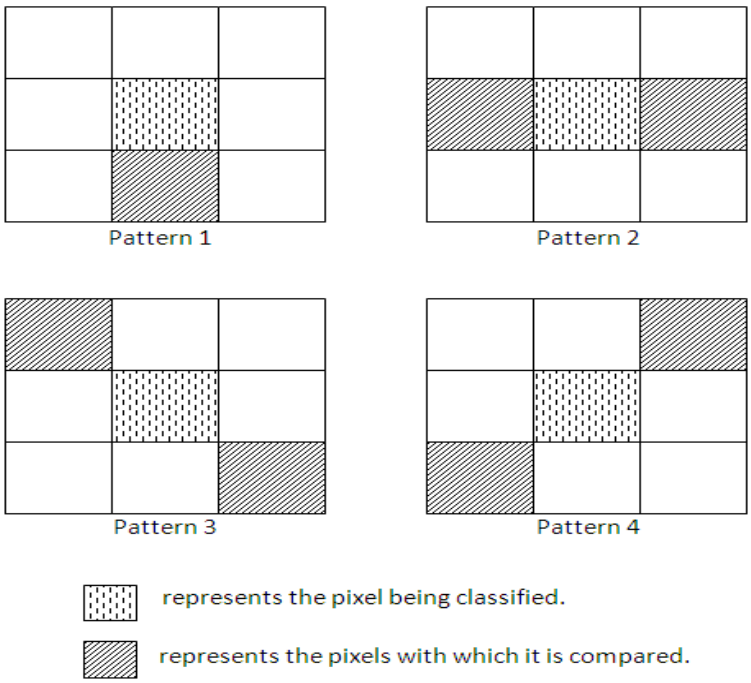



Figure 2-11: One dimensional three pixel patterns [11].

## 2.3. Intrapicture Prediction

HEVC uses block based intraprediction to exploit the spatial correlation within the picture [9][6]. It has 35 luma intraprediction modes, intraprediction can be done at different block sizes ranging from 4x4 to 64x64, the size PU has. Figure 2-12 shows the luma intraprediction modes of HEVC [8]. 35 modes include planar intraprediction mode, which is employed for predicting smooth picture regions, where the prediction is produced from the average of two linear interpolations (horizontal and vertical).

The number of chroma intraprediction modes in HEVC is six whereas H.264/AVC supports four chroma intraprediction modes. The six modes are shown in Table 2-1 [8]. In principle, DM and LM exploit the correlation of the luma and chroma components. If the texture directionality of luma and chroma components look alike, DM is selected, in this case it uses exactly the same mode as that of the luma component. On the other hand if the sample intensities of luma and chroma components are highly correlated, LM is selected. In this case, the chroma components are predicted using luma components reconstructed by the linear model relationship [15]. DM and LM complement each other in terms of coding parameter due to the type of correlations exploited by them. Because of the presence of correlation between luma and chroma, DM and LM are the most frequently selected modes for intraprediction of the chroma components.

Table 2-1: Chroma intra prediction modes [8].

| No. | Modes | Description |
|---|---|---|
| 1 | DM | Direct mode |
| 2 | LM | Linear mode |
| 3 | mode 0 | Vertical |
| 4 | mode 1 | Horizontal |
| 5 | mode 2 | DC |
| 6 | mode 3 | Planar |

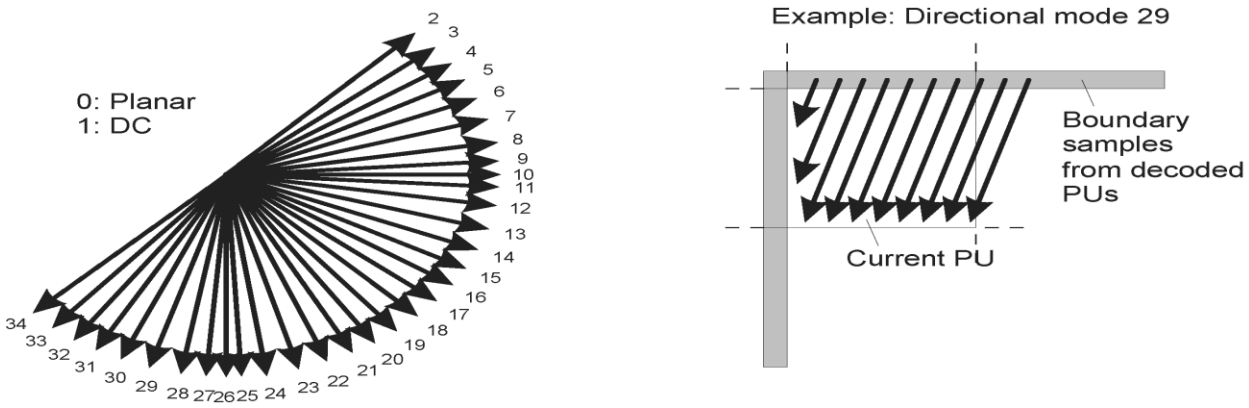



Figure 2-12: Modes and directional orientations for intrapicture prediction [8].

## 2.4 Summary

This chapter has introduced HEVC standard with its encoder and decoder block diagrams. Important video coding layer features and intra prediction are briefly introduced. In the next chapter, scalable video coding is introduced, scalable HEVC is briefly explained. Intra, inter prediction and inter-layer prediction is described and the proposed algorithm to reduce the inter-layer inter prediction is discussed.

# Chapter 3

## Scalable Video Coding

The evolution of digital video technology and the constant advancements in the communication infrastructures are propelling a great number of interactive multimedia applications, such as real time video conference, web video streaming and mobile TV among others [18][17]. The interactive video usage has created an exigent market of consumers, which demands the best video quality wherever they are and whatever their network support is. On this purpose the receiver's characteristics like required bit rate, resolution and frame rate should match those of the transmitted video, thus targeting to provide the best quality subject to receiver's and network's limitations. The same link is often used to transmit to the small cell phones or to the high performance equipments like HDTV workstations. In addition, the stream should also adapt to the lossy wireless networks. Therefore these in-deterministic and heterogeneous networks pose a huge problem for traditional video encoders which do not allow for on-the-fly video streaming adaptation.

To overcome this drawback, scalability of video coding has been introduced [18]. The principle of scalable video encoder is to divide the traditional single stream video in a multi stream flow, composed by distinct and complementary components, often referred to as layers. Figure 3-1 shows the concept of transmitter encoding the input video sequence into three complimentary layers [18]. The receiver can select and decode different number of layers each corresponding to distinct video characteristics in accordance with the processing constraints of both the network and device itself.

### 3.1 Types of Scalability

A video bitstream is called scalable when parts of the stream can be removed in a way that the resulting sub-stream forms another valid bitstream for a particular target

decoder and the sub-stream represents the source content with a reconstruction quality that is less than that of the complete original bit stream but is high when considering the lower quantity of the remaining data [16]. Bit streams that do not provide this property are referred to as single layer bit streams. The common modes of scalability are spatial, temporal and quality scalability. Figure 3-2 shows the basic types of scalable video coding [17].

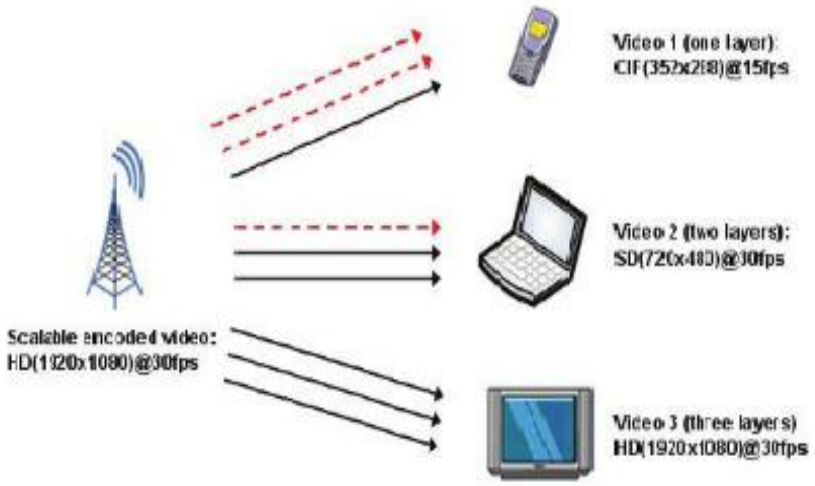



Figure 3-1: The streaming of the encoded layers and the particular layers decoded at the different receivers [18].

Spatial scalability describes the case in which subsets of bitstream represent the source content with a reduced picture size i.e. spatial resolution. Temporal scalability describes the case in which subsets of the bitstream represent the source content with reduced frame rate. In quality scalability, the substream gives the same spatio-temporal resolution as the complete bitstream, but with lower fidelity where fidelity is often informally referred to as signal-to-noise ratio (SNR). Quality scalability is commonly called as fidelity or SNR scalability. Region-of-interest (ROI) and object-based scalability where

the sub-streams typically represent spatially contiguous regions of the original picture area are more rarely required scalability types. The different types of scalability can also be combined such that multitude representation with different spatio-temporal resolutions and bit rates can be supported with a single scalable bit stream.

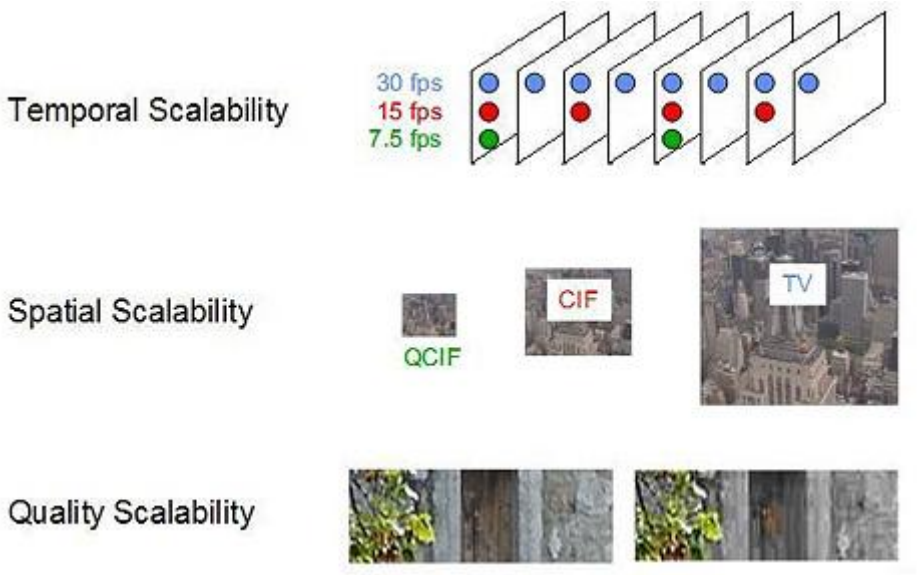



Figure 3-2: Basic types of scalable video coding [17].

## 3.2 Applications of Scalable Video Coding

Scalable video coding (SVC) supports number of applications [16][19]- [21]. For instance consider video transmission service with heterogeneous clients where multiple bit streams of the same source content differing in coded picture size, frame rate and bit rate should be provided simultaneously. With SVC scheme the source content has to be encoded only once – for the highest required resolution and bit rate, resulting in a scalable bit stream from which representations with lower resolution and/or quality can be obtained by discarding selected data. For instance, a client with restricted resources such as display resolution, battery power or processing power, needs to decode only a part of the delivered bit stream. Similarly, in multicast scenario, terminals with different

capabilities can be served by a single scalable bit stream. In an alternative scenario, an existing video format (like QVGA) can be extended in a backward compatible way by an enhancement video format (like VGA).

Scalable bit stream usually contains parts with differing importance in terms of decoded video quality [16]. This property in conjunction with unequal error protection is especially useful in any transmission scenario with unpredictable throughput variations and/or relatively high packet loss rates. By using a stronger protection of the more important information, error resilience with grateful degradation can be achieved up to a certain degree of transmission errors. Media Aware Network Elements (MANEs) [23], which receive feedback messages about the terminal capabilities and/or channel conditions, can remove unnecessary parts from a scalable bit stream before forwarding it. Therefore the loss of important transmission units due to congestion can be avoided and the overall error robustness of the video transmission service can be substantially improved.

Scalability plays a desirable role in the surveillance applications [23][16]. In surveillance, video sources not only need to be viewed on multiple devices ranging from high-definition monitors to videophones or personal digital assistants (PDAs), but also need to be stored and archived. With scalable video coding high resolution/high quality parts of a bit stream can ordinarily be deleted after some expiration time, so that only low quality copies of the video are archived for long term. This approach can also be used in personal video recorders and video networking.

Figure 3-3 depicts the encoder block diagram for scalable video coding with two layers, one being the base layer and other being the enhancement layer [33]. At first glance the scalable encoder consists of two encoders, one for each layer.

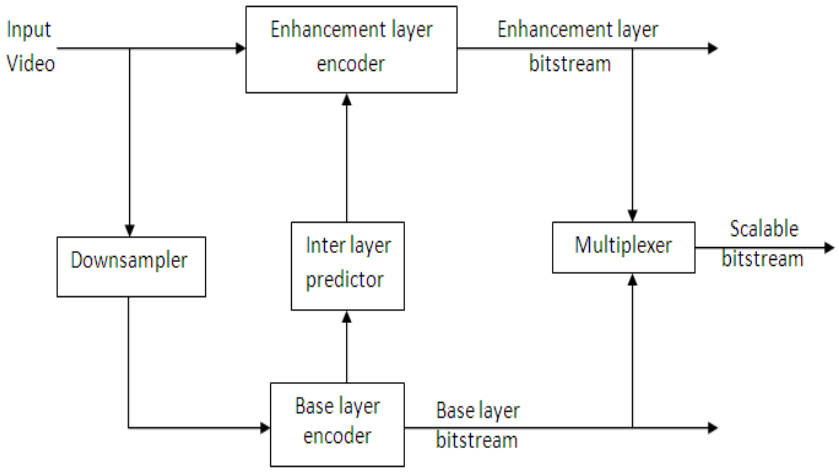



Figure 3-3: Scalable video coding encoder block diagram [33].

Figure 3-4 shows the decoder block diagram for scalable video coding [33], where after entropy decoding (ED) each layer is inverse quantized and then all are added together to represent the final DCT coefficients. These coefficients are inverse transformed and are added to the motion compensated previous picture to reconstruct the final picture.

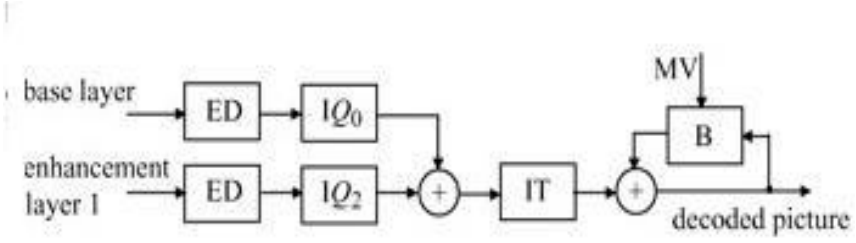



Figure 3-4: Scalable video coding decoder block diagram, where ED is entropy decoder, IQ is inverse quantizer, IT is inverse transform and B is picture buffer [33].

### 3.3 Scalable Extension of HEVC

The scalability extension to HEVC enables spatial and coarse grain SNR scalability and is referred to as “SHVC” [25][26]. The SHVC design uses a multi-loop

coding framework, such that in order to decode an enhancement layer, its reference layers have to first be fully decoded to make them available as prediction references. The high level design of the HEVC scalability extension, example multi-loop coder/decoder and restrictions against block-level changes were motivated by ease of implementation, especially the possibility to re-use existing HEVC implementations, even though the overall number of computations and memory accesses of the decoder would be higher than in a single-loop design. Multi-loop coding also provides coding efficiency advantages over single-loop coding designs.

The coding tools in the HEVC scalability extension are limited to changes at the slice level and above [25]. The reference layer picture, resampled if necessary, is used as additional reference picture for enhancement layer prediction, which enables inter-layer texture and motion parameter prediction. The multi-loop design is somewhat similar to AVC's and HEVC's multiview extensions 79[51], which require full decoding of the base view in the case of decoding dependent views. However, in the multiview case, all views have the same resolution so that no resampling is needed. The same applies for the case of SNR scalability, where scalable layers represent pictures of same spatial resolution.

In terms of performance and complexity, layer dependent coding is usually compared against simulcast (independent coding of equivalent signals). Typical applications where scalable coding or simulcast would be applied, such as flexible rate or resolution switching, would usually only output one of the layers. However in the case of multi-loop decoding, it is still necessary to decode all reference layers, such that the overall decoding complexity increases compared to simulcast. This effect is more critical in SNR scalability, where the reference layers are not subsampled. On the other hand,

dependent coding of layers has advantages over simulcast in terms of compression performance, as reported in the end of this section.

When spatial scalability is implemented, the decoded reference layer picture is resampled using a normatively defined upsampling filter for the spatial scalability case. Spatial scalability ratios in the current design of HEVC are limited to 1.5x and 2x spatial resampling factors in each dimension, since these factors are sufficient to cover the primary anticipated use cases and the restriction simplifies implementation[25].

*3.3.1 Upsampling Filter*

In HEVC scalability extension, the upsampling filter is used to map reconstructed sample values from the reference layer to higher resolution sampling grid of the enhancement layer [25][27]. This supports the use of the reconstructed reference layer sample values for enhancement layer prediction. In the scalability extension, the upsampling process is defined as a normative part of the standard. The downsampling process used to create the source pictures of lower resolution as input to the encoding process of the reference layer is left outside the scope of the standard.

In the scalability extension, the upsampling filter is defined as an 8-tap polyphase finite-impulse-response (FIR) filter for luma resampling, and a 4-tap polyphase FIR filter for chroma resampling [25]. One motivation for the number of taps is consistency with the HEVC motion compensation design for fractional-position interpolation, which also uses 8-tap and 4-tap FIR filters for luma and chroma interpolation, respectively. However, the corresponding reference layer position is defined with 1/16 sample precision, so filters for additional phase shifts are needed. Motion compensation operates with only 1/4 sample precision for luma and 1/8 sample precision for 4:2:0 chroma. The basic design enables the use of arbitrary upsampling ratios, in which filters for all 16 phase positions would be

necessary, but the current specification is restricted to ratios of 1.5 and 2, for which fewer positions are needed.

Scaled reference layer offsets may be signaled to enable the reference layer and enhancement layer the freedom to not fully correspond to the same region of a picture. Scale factors for the horizontal and vertical directions are computed as the ratio between the relevant enhancement and reference layer regions widths and heights, respectively. For each enhancement layer sample, the corresponding reference layer sample location and 1/16 sample phase is determined considering the scale factors and the scaled reference layer offsets. The 8-tap (or 4-tap filter) coefficients which correspond to the calculated phase are applied to the input reference layer samples, which are the samples at the reference sample location and its neighboring samples in the reference layer. Figure 3-5 shows the filter coefficients for the luma upsampling filter [25]. The selection of the tap values is again analogous to the HEVC motion compensation interpolation process. The 0 and 8/16 phases are identical to the 0 and 1/2 phases of the HEVC process, and are needed for upsampling by the ratio 2. The 0, 5/16 and 11/16 phases are needed for the ratio 1.5, where the latter two are designed using the same approach as the 1/4 and 3/4 phases in the motion compensation interpolator and satisfy the same constraints on frequency response and the precision of the calculation.

| Phase | $T_0$ | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_6$ | $T_7$ |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 64 | 0 | 0 | 0 | 0 |
| 5/16 | −1 | 4 | −11 | 52 | 26 | −8 | 3 | −1 |
| 8/16 | −1 | 4 | −11 | 40 | 40 | −11 | 4 | −1 |
| 11/16 | −1 | 3 | −8 | 26 | 52 | −11 | 4 | −1 |

Figure 3-5: 8-tap filter coefficients for upsampling filter [25].

Similarly, coefficients for the chroma upsampling filter are shown in Figure 3-6 [25]. Here, chroma upsampling requires the definition of nine phases of the polyphase filter to support the upsampling ratios of 1.5 and 2. The reason for the larger number of phases necessary for chroma is the inherent phase shift between luma and chroma samples in 4:2:0 chroma subsampling, which is considered when mapping base and enhancement layer chroma positions. As in the luma filter, the phases corresponding to those used in motion compensation have the same tap values, while phases not used in the motion compensation satisfy the same constraints on frequency response and calculation precision.

| Phase | $T_0$ | $T_1$ | $T_2$ | $T_3$ |
|---|---|---|---|---|
| 0 | 0 | 64 | 0 | 0 |
| 4/16 | −4 | 54 | 16 | −2 |
| 5/16 | −6 | 52 | 20 | −2 |
| 6/16 | −6 | 46 | 28 | −4 |
| 8/16 | −4 | 36 | 36 | −4 |
| 9/16 | −4 | 30 | 42 | −4 |
| 11/16 | −2 | 20 | 52 | −6 |
| 14/16 | −2 | 10 | 58 | −2 |
| 15/16 | 0 | 4 | 62 | −2 |

Figure 3-6: Filter coefficients for chroma upsampling filter [25].

*3.3.2 Intra Prediction using Reconstructed Base Layer Samples*

The first scalable coding tool in which the enhancement layer prediction signal is formed by copying or up-sampling the reconstructed samples of the co-located area in the base layer is called IntraBL prediction mode. IntraBL prediction mode is illustrated in Figure 3-7 [28]. This coding mode is already known from the scalable profiles of H.262 | MPEG-2 Video, H.263, and MPEG-4 Visual [25]. It is also similar to the inter-layer intra mode in the scalable extension of H.264 | MPEG-4 AVC. However, in H.264 | MPEG-4 AVC, the usage of this mode is restricted to macroblocks for which the co-located base

layer area is coded using an intra prediction mode (for enabling single-loop decoding), whereas this mode can be selected at a CU level in the HEVC extension,. This mode is also used implicitly for the InterBL mode, which is described in sec. 3.6, when a CU completely or partially covers an intra block in the base layer. The residual signal is transmitted by transform coding using the syntax for inter-predicted CUs. At the decoder side, the final reconstructed signal is obtained by adding the transmitted residual signal to the inter-layer intra prediction signal. Up-sampling filters are required in order to use base layer signals for prediction in the enhancement layer, if the enhancement layer uses a spatial resolution larger than the base layer. While the scalable extension of H.264/MPEG-4 AVC uses 4-tap FIR filters for upsampling of the luma signal [16], 8-tap filters are applied in the proposed HEVC extension. For chroma, bi-linear filters are used. The filters are 2-D separable, i.e., 1-D filters operate horizontally and vertically. Similar to H.264 | MPEG-4 AVC, the filters are provided with approximately 1/16th sample phase offsets. For supporting arbitrary resolution ratios, for each enhancement layer sample position, the used filter is selected based on the required phase shift [18].

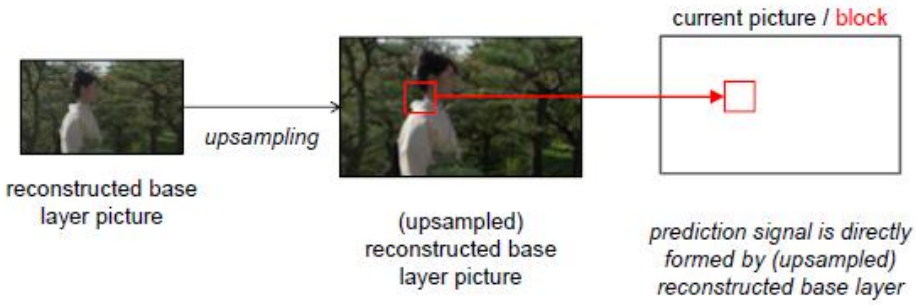



Figure 3-7: IntraBL prediction mode [28].

The upsampling filters used for the IntraBL mode are designed to provide a good coding efficiency over a wide variety of base and enhancement layer signals. However, even within each picture, video signals may show a high degree of non-stationarity.

Additionally, quantization errors and noise may show varying characteristics in different parts of a picture. Hence, to adapt the upsampling filter to local signal characteristics, another inter-layer intra coding mode, referred to as InterBLFilt mode is introduced. This mode is used in the same way as the InterBL mode. The only difference is that, for generating the enhancement layer prediction signal, a smoothing filter with coefficients [1 2 1] / 4 is applied horizontally and vertically after upsampling or copying the reconstructed base layer samples. If the IntraBL or IntraBLFilt mode is selected, the intra deblocking filter strength as specified in HEVC can be too high, since the base layer signal that is used as prediction has already been deblocked. This is taken into account by adapting the de-blocking filter strength derivation in the enhancement layer.

*3.3.3 Intra Prediction using a Difference Signal*

In the IntraBL and IntraBLFilt modes only the reconstructed base layer samples are used for generating the prediction signal for an enhancement layer [28]. As a consequence, the prediction signal has a systematic error. In spatial scalable coding, the generated prediction signal mainly contains low-frequency components; the high-frequency components which represent the difference between the base and enhancement layer resolutions are missing in the enhancement layer prediction signal. In quality scalable coding, the quantization step size used for the base layer is larger than the quantization step size for the enhancement layer. Hence, the inter-layer intra prediction signal contains the larger quantization noise of the base layer. To some extent, this effect also appears in spatial scalable coding. For reducing these systematic errors, two further inter-layer intra prediction modes have been introduced, for which the final enhancement layer prediction signal is obtained by superimposing two intermediate prediction signals. The first intermediate prediction signal is generated by using the reconstructed base layer signal of the co-located area in the base layer and the second

intermediate prediction signal is generated by using enhancement layer data. The first combined base and enhancement layer intra prediction mode, which is also referred to as DiffIntra mode, uses a difference signal between already reconstructed enhancement layer samples and upsampled base layer samples [30][26]. Figure 3-8 shows the generation of the prediction signal for an enhancement layer block 77[28].

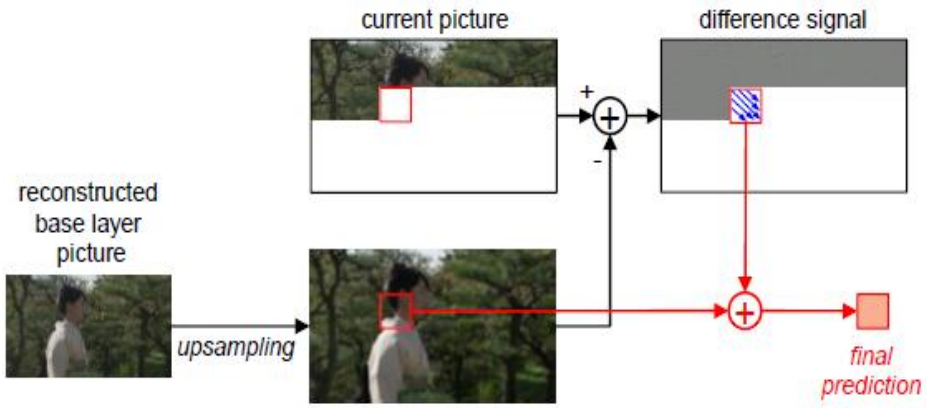



Figure 3-8: Spatial intra prediction using a difference signal [28].

The first component of the enhancement layer prediction signal is derived by copying or, for spatial scalable coding, upsampling the reconstructed base layer samples of the co-located area in the base layer. For upsampling the base layer signal, the same interpolation filters as for the IntraBL mode described in section 3.3.2 are used. The second component of the prediction signal is derived by spatial intra prediction using a difference signal for neighboring samples of already reconstructed blocks. This difference signal represents the sample-wise difference between the reconstructed enhancement layer signal and the upsampled or copied reconstructed base layer signal of the co-located area. For spatial intra prediction, the same spatial intra prediction modes as for single-layer HEVC are used. The selected intra prediction modes are transmitted using the conventional HEVC syntax. Basically, the only difference to the spatial intra prediction as specified in HEVC is that difference samples instead of reconstructed enhancement

layer samples are used. The final enhancement layer prediction signal is obtained by adding up the upsampled base layer signal and the intra-predicted difference signal.

Similar to the IntraBL and IntraBLFilt modes, the DiffIntra mode can be selected at a CU level. The residual signal is transmitted via transform coding using the syntax for intra-predicted CUs.

*3.3.4 Weighted Intra Prediction using Base and Enhancement Layer Samples*

Another intra prediction mode in which a reconstructed base layer signal is combined with an enhancement layer prediction signal is called WeightIntra [28]. Similarly to the DiffIntra mode, the reconstructed and, for spatial scalable coding, upsampled base layer signal of the co-located area in the base layer constitutes one component of the prediction signal. For upsampling, the same interpolation filters as for the IntraBL mode are used. The second component is obtained by conventional spatial intra prediction using neighboring enhancement layer samples of already reconstructed blocks. The selected spatial intra prediction modes are transmitted using the regular HEVC syntax. The final enhancement layer prediction signal is obtained by low-pass filtering the base layer component, high-pass filtering the enhancement layer component, and adding up the resulting signals

Figure 3-9 shows the implementation of the low- and high-pass filtering done in the transform domain [28]. Therefore, both intermediate prediction signals are first transformed using an approximation of a DCT of the corresponding block size. Then, the resulting transform coefficients are weighted according to the spatial frequencies, where the weights for the base layer signal are set such that the low-frequency components are retained and the high frequency components are suppressed and the weights for the enhancement layer signal are set and vice versa. The final enhancement layer prediction

signal is obtained by summing up the weighted base and enhancement layer coefficients and applying an approximation of an inverse DCT.

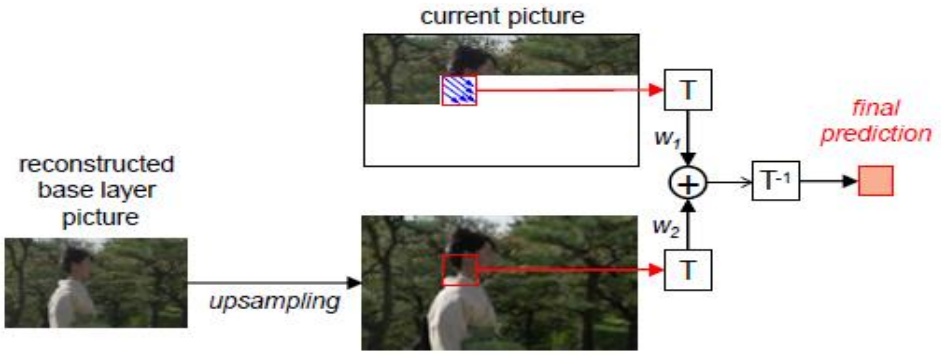



Figure 3-9: Weighted intra prediction. *w1, w2* represent weights, T and $T^{-1}$ represent the forward and inverse transforms [28].

The used inverse transforms are the same as the ones specified in the HEVC decoding process. The used forward transforms are the inverses of these transforms. The weighting matrices have been obtained by least-squares optimization using a set of test sequences. Different weighting matrices are used for different block sizes. Furthermore, one out of three sets of weighting matrices is selected based on the ratio of enhancement and base layer resolution. The sets of weighting matrices have been optimized for resolution ratios of 1.0, 1.5, and 2.0. The WeightIntra mode can be selected on a CU level and the residual signal is transmitted via transform coding using the syntax for intra-predicted CUs.

*3.3.5 Inter Prediction using Difference Pictures*

We have seen DiffIntra mode illustrated in section *3.3.2*, the scalable extension of HEVC also supports a difference prediction mode for motion-compensated prediction, where the final enhancement layer prediction signal is formed as a sum of base layer signal and a difference prediction signal. This mode of prediction is also referred to as DiffInter mode. Figure 3-10 illustrates the generation of the prediction signal for

enhancement layer [28]. In spatial scalable coding, the first component of the enhancement layer prediction signal is derived by upsampling the reconstructed base layer samples of the co-located area in the base layer. The second component of the prediction signal is obtained by motion-compensated prediction using difference pictures. The difference pictures represents the difference between the enhancement layer reconstruction for already reconstructed pictures and the corresponding reconstructed and upsampled base layer pictures. The final enhancement layer prediction signal is obtained by summing the motion-compensated difference signal to the base layer component.

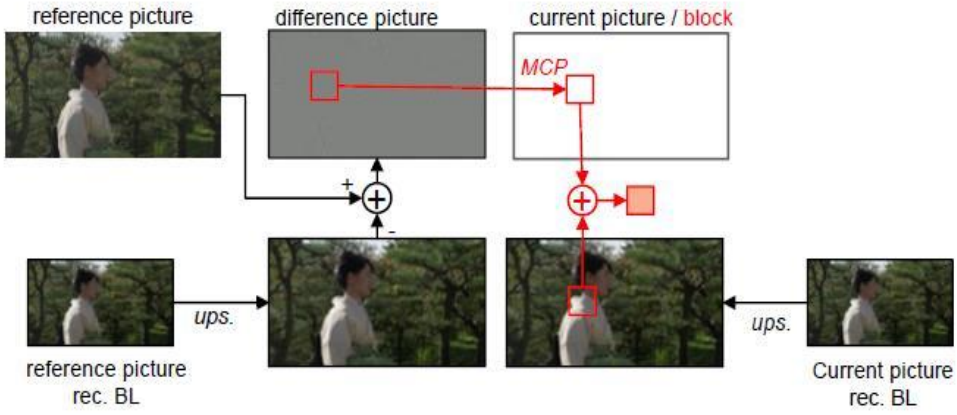



Figure 3-10: Motion-compensated prediction using difference pictures [28].

For the motion-compensated prediction using difference pictures, the same partitioning types as for conventional inter modes are supported and the regular HEVC syntax is used for motion parameters transmission. However to generate the motion-compensated prediction signal at fractional sample positions, a simple bi-linear interpolation with quarter sample position is used instead of 8-tap or 7-tap interpolation filters.

### *3.3.6 Inter-Layer Prediction of Prediction Parameters*

Scalable extension of HEVC supports a method for improving the coding of intra prediction modes by using the base layer intra prediction mode and also supports three methods for exploiting the motion information of the base layer for enhancement layer coding [28]. Conceptually two methods for motion parameter coding are similar to the inter layer motion prediction in the scalable extension of H.264/MPEG-4 AVC, but they differ in many details [16][28].

In the current scalable extension of the HEVC standard, the syntax element indicates whether the CU is coded in inter or intra prediction mode and the corresponding prediction parameters are transmitted [28]. Calculations for partitioning of CU in enhancement layer is carried independently similar to the base layer however the motion parameters are derived from base layer. In order to derive the motion information from the base layer, each CUs in the enhancement layer are decomposed into 8x8 sub-blocks and for each such sub-blocks a co-located block in the base layer is determined. If such co-located block in the base layer is coded in an intra prediction mode, the 8x8 sub-block in the enhancement layer is marked as intra-coded otherwise the 8x8 sub-block in the enhancement layer is simply associated with that of the base layer motion parameters. The number of motion hypotheses and reference indices are copied from the base layer block and the motion vectors are scaled according to the resolution ratio between base and enhancement layer. Next, the 8x8 sub-blocks in the enhancement layer which posses the same motion information are recursively joined in a quadtree fashion. If all 4 sub-blocks of 16x16 region have the same motion parameters or if all are marked as intra-coded, the sub-blocks are represented as single 16x16 sub-block. Next, four neighboring 16x16 blocks can be represented by a single 32x32 block and so on. If four neighboring blocks do not have same motion parameters but each horizontal or vertical

pair of the sub-blocks has the same prediction parameters, the corresponding PU partitioning is selected. The joining of the sub-blocks is carried out until there can be no possibility for any further joining and a valid CU/PU partitioning for the considered CU is obtained.

### 3.4 Proposed Algorithm to Extract the Motion Information from BL to EL

In order to reduce the computational complexities involved in the calculation of PU partitioning and inter layer motion prediction an efficient algorithm for reuse of information available in the BL is proposed. In the proposed algorithm the resolution ratio between the base and enhancement layer say 'R1' is calculated. The base layer is encoded with normal HEVC encoder, encoding involves motion prediction process along with other processes. To encode enhancement layer, the proposed algorithm bypasses the calculation of PU partitioning sizes and motion prediction step, but in turn extracts the motion information and the PU partitioning sizes required for the EL directly from the BL using its motion parameters and PU division information. This is carried out by scanning the BL in a left to right fashion starting at the first available block at the top left. On meeting a PU in the BL the information of its size and motion vector will be scaled according to the resolution ratio to produce a PU of scaled size. Also in order to avoid the computational complexity of copying all the scaled motion information to the PU in EL the scaled motion information will directly be used to obtain a prediction image from the reference image in the EL further avoiding the burden of time spent on coding these motion parameters. After reaching the last available PU in the bottom right corner of the BL, the predicted image in the EL will be used to obtain the residual image. The generated residual frame is then encoded and transmitted.

## 3.5 Summary

In this chapter, basics of scalable video coding with its encoder and decoder block diagrams are briefly introduced. Concepts in scalable extension of HEVC were explained together with different intra prediction modes, inter prediction and inter layer prediction for prediction parameters are explained. Based on the proposed algorithm results for the thesis are provided in the next chapter.

# Chapter 4

# Results and Conclusions

## 4.1 Test Conditions

The algorithm proposed in the thesis was tested using four video sequences listed in Table 4-1, with thirty frames for each sequence. Latest HM-10.0-dev-SHM was used to implement the algorithm in the 'random access main' profile with GOP size equal to eight [28] [29]. The algorithm was implemented in C++ programming language, with QPs of 22, 27, 32 and 37 for the base layer and QPs of 20, 25, 30 and 35 for the enhancement layer as recommended by JCT-VC [37]. A frame of each video sequence used is shown in Appendix A.

Table 4-1: Video test sequences [31].

| No. | Sequence name | Resolution | Type | No. of frames |
|---|---|---|---|---|
| 1 | City | 176x144 | QCIF | 30 |
| | | 352x288 | CIF | 30 |
| | | 704x576 | 4CIF | 30 |
| 2 | Crew | 176x144 | QCIF | 30 |
| | | 352x288 | CIF | 30 |
| | | 704x576 | 4CIF | 30 |
| 3 | Harbour | 352x288 | CIF | 30 |
| | | 704x576 | 4CIF | 30 |
| 4 | Ice | 176x144 | QCIF | 30 |
| | | 352x288 | CIF | 30 |

## 4.2 Encoding Time Gain

The encoding time gain achieved by implementing the proposed algorithm for inter-layer inter prediction is in the range of 24%-28% as shown in Figures 4-1 thru 4-12 when compared to unmodified HM 10.0-dev-SHM.

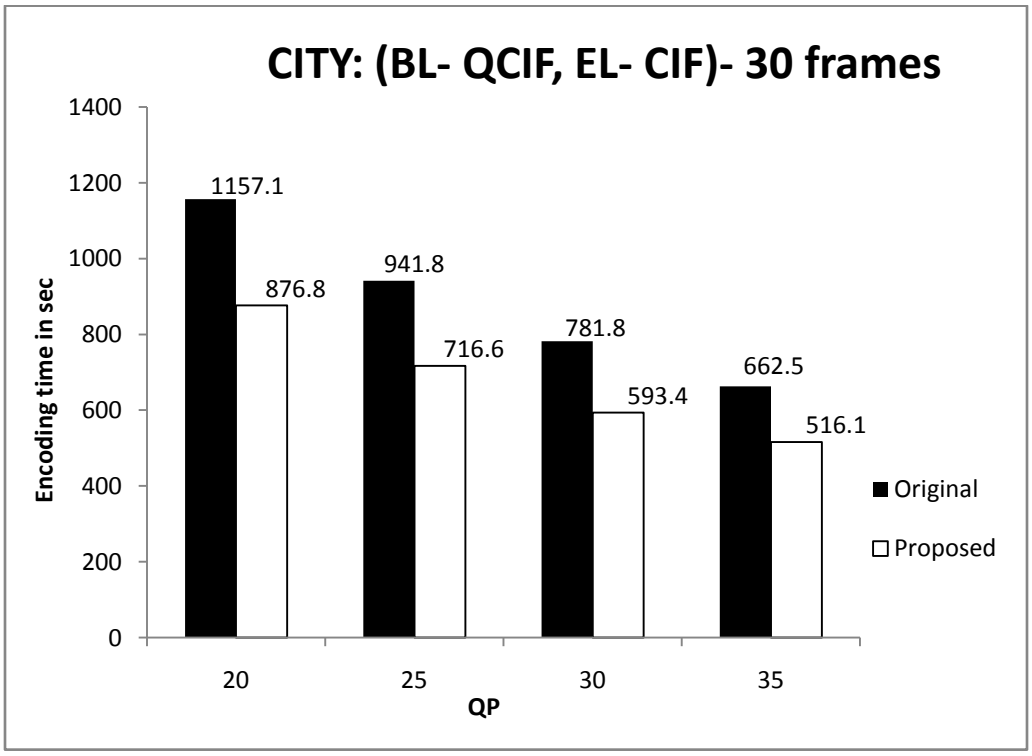



Figure 4-1: Encoding time vs. quantization parameter for City with BL- QCIF and EL- CIF

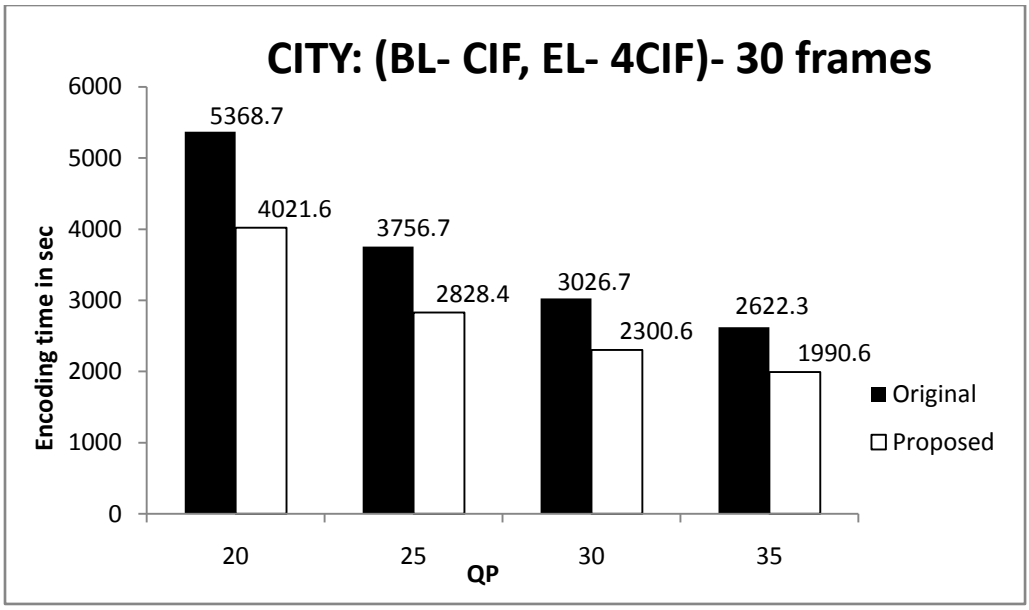



Figure 4-2: Encoding time vs. quantization parameter for City with BL- CIF and EL- 4CIF

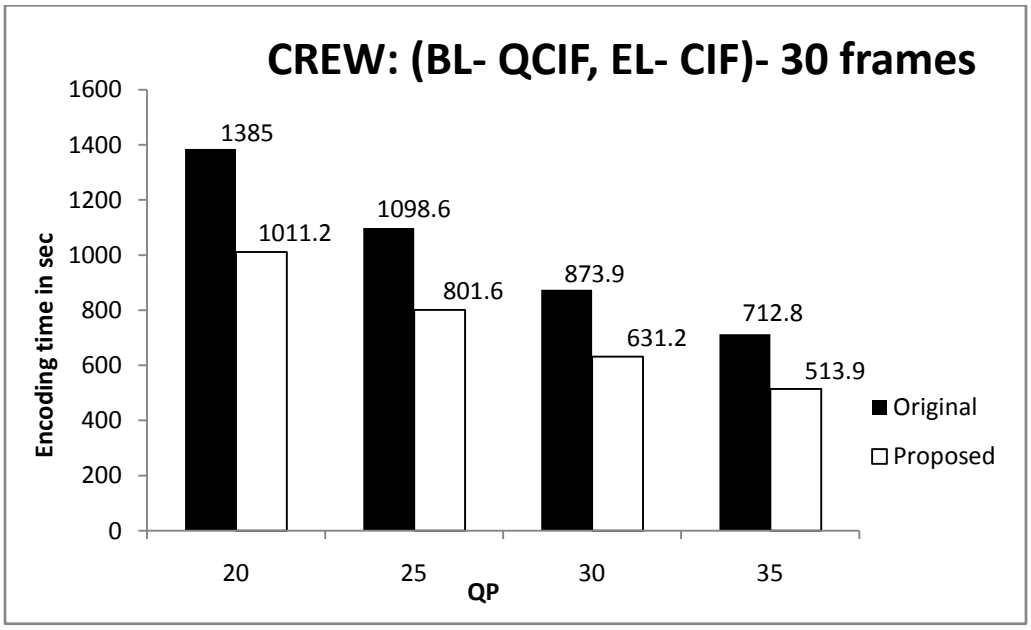



Figure 4-3: Encoding time vs. quantization parameter for Crew with BL- QCIF and EL- CIF

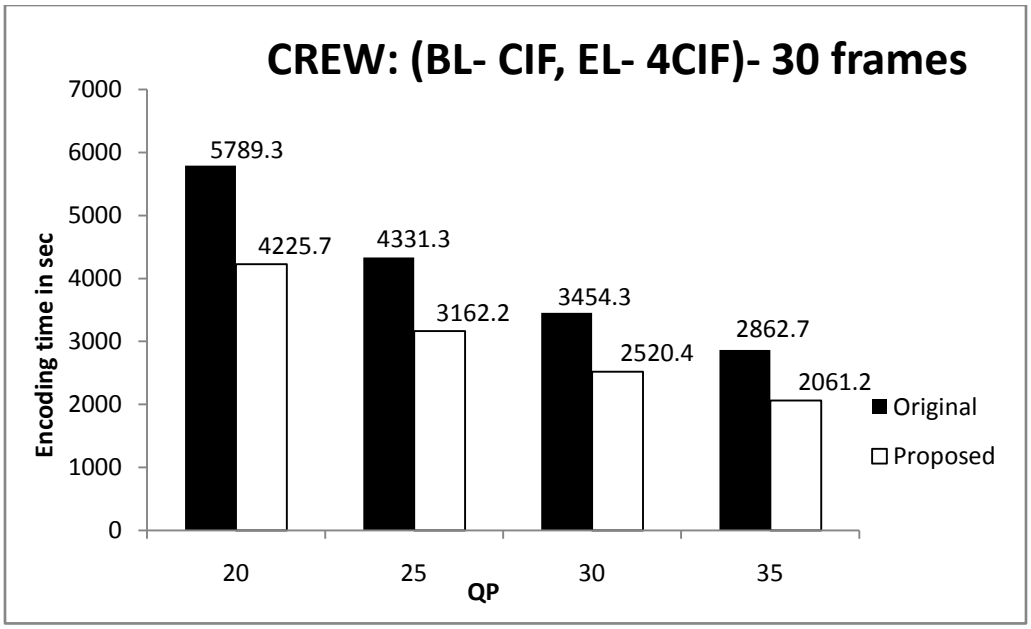



Figure 4-4: Encoding time vs. quantization parameter for Crew with BL- CIF and EL- 4CIF

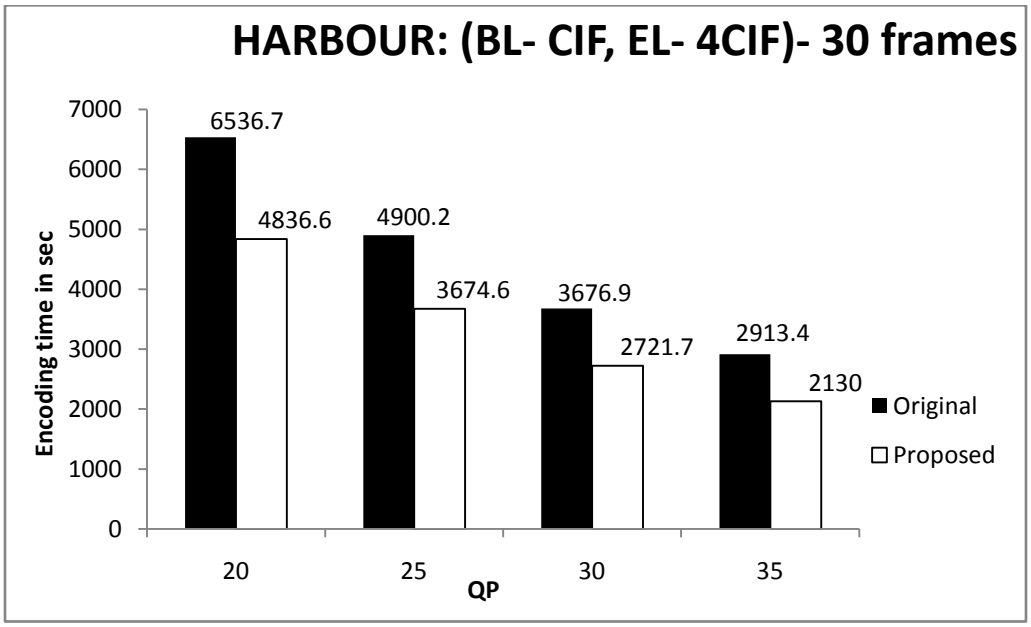



Figure 4-5: Encoding time vs. quantization parameter for Harbour with BL- CIF and EL- 4CIF

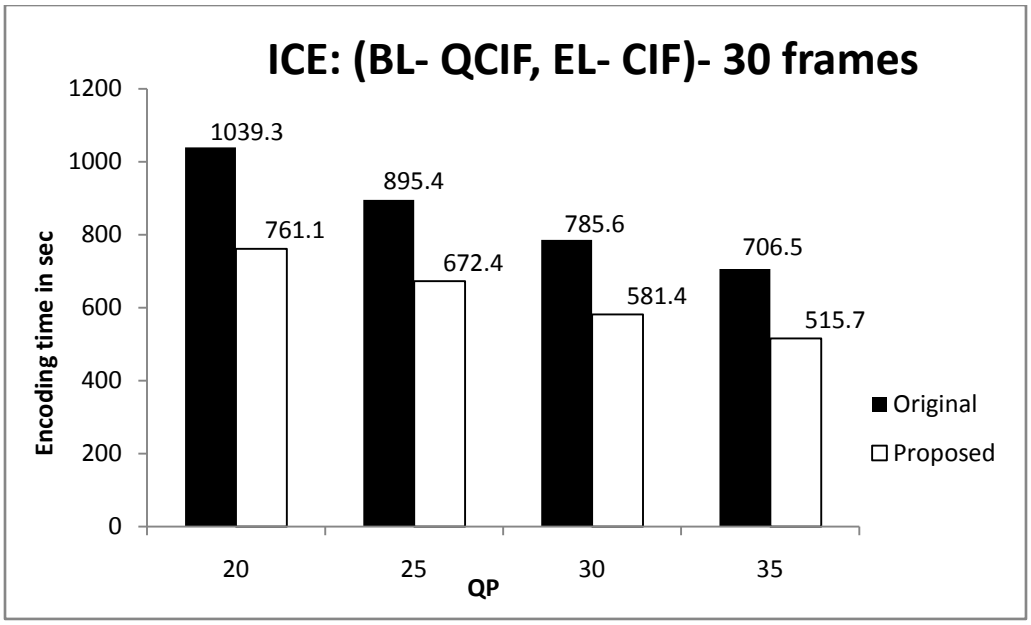



Figure 4-6: Encoding time vs. quantization parameter for Ice with BL- QCIF and EL- CIF

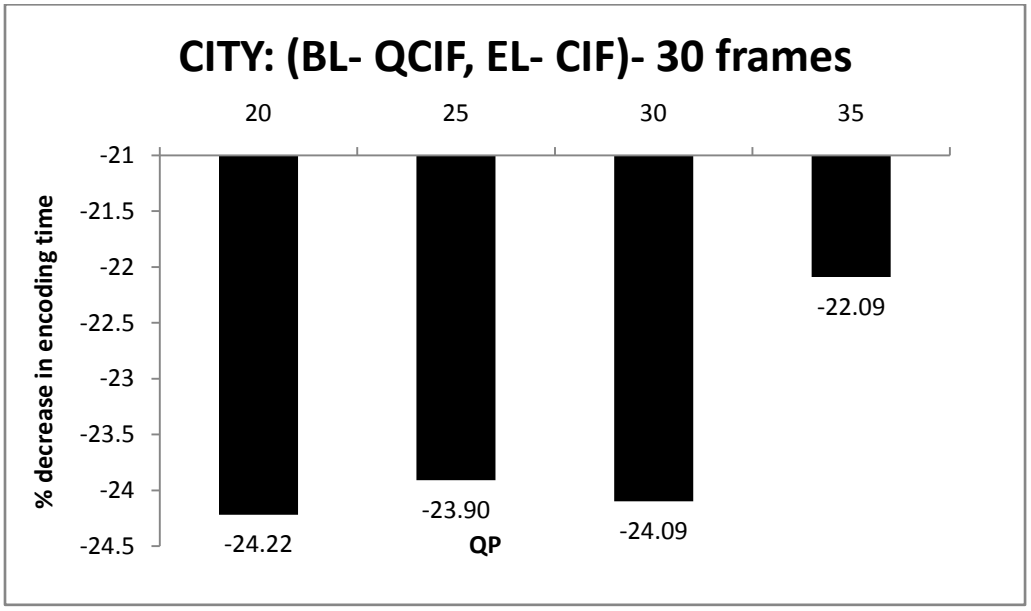



Figure 4-7: %decrease in encoding time vs. quantization parameter for City with BL-QCIF and EL- CIF

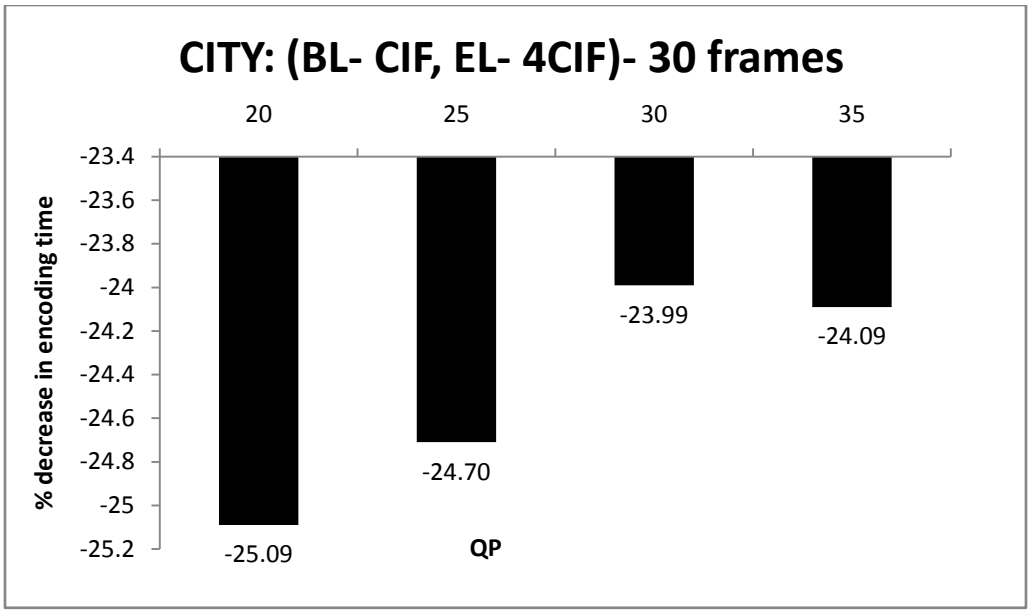



Figure 4-8: : %decrease in encoding time vs. quantization parameter for City with BL- CIF and EL- 4CIF

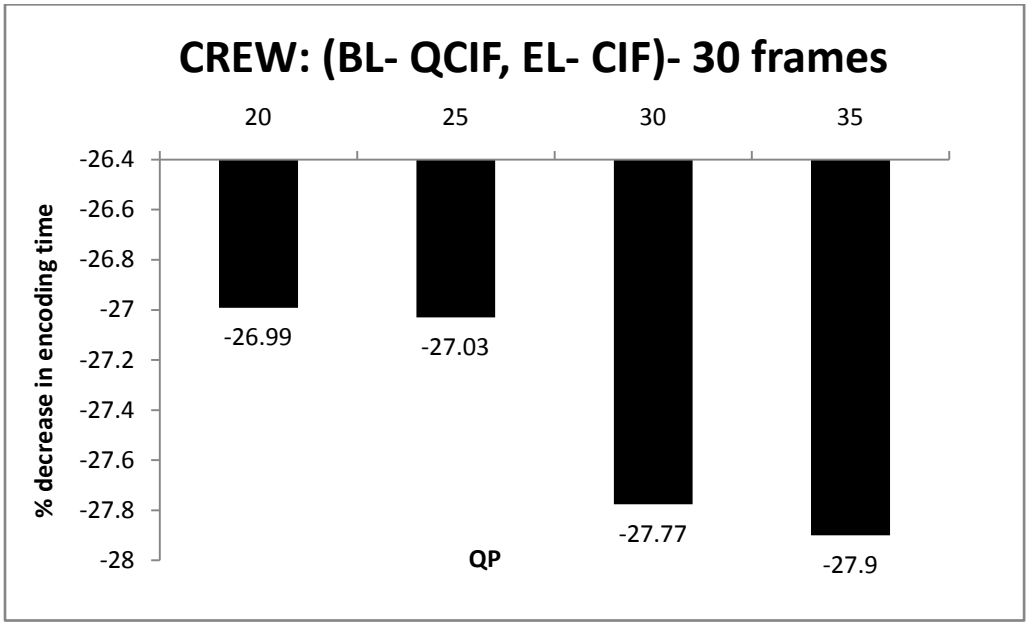



Figure 4-9: %decrease in encoding time vs. quantization parameter for Crew with BL- QCIF and EL- CIF

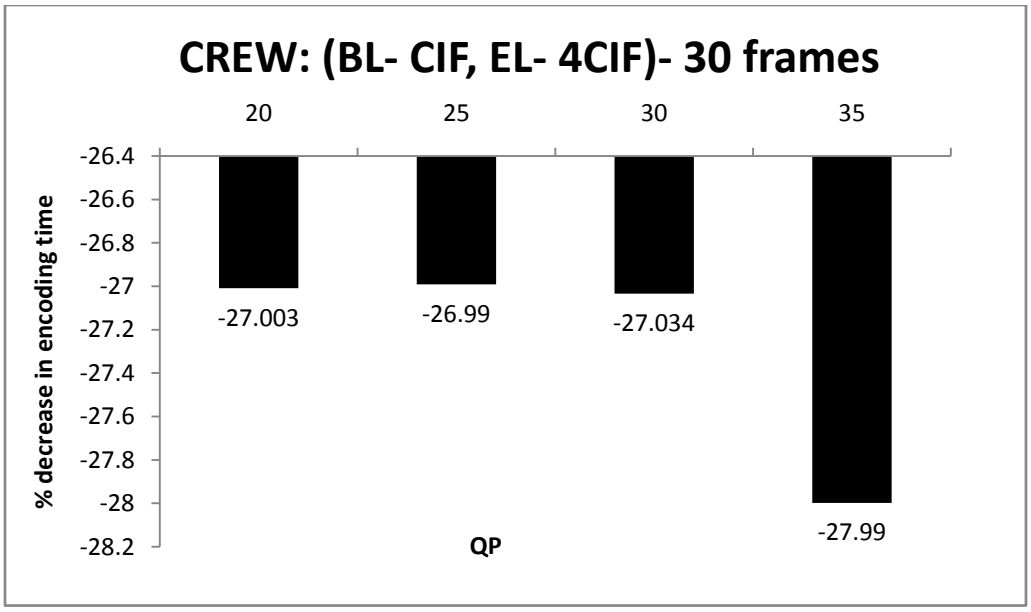



Figure 4-10: %decrease in encoding time vs. quantization parameter for Crew with BL- CIF and EL- 4CIF

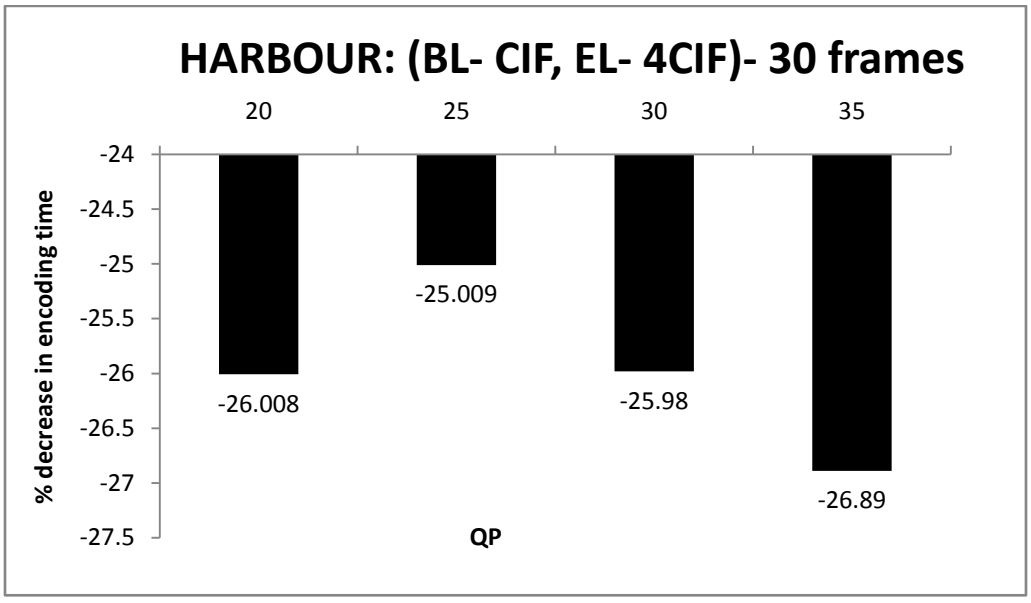



Figure 4-11: %decrease in encoding time vs. quantization parameter for Harbour with BL- CIF and EL- 4CIF

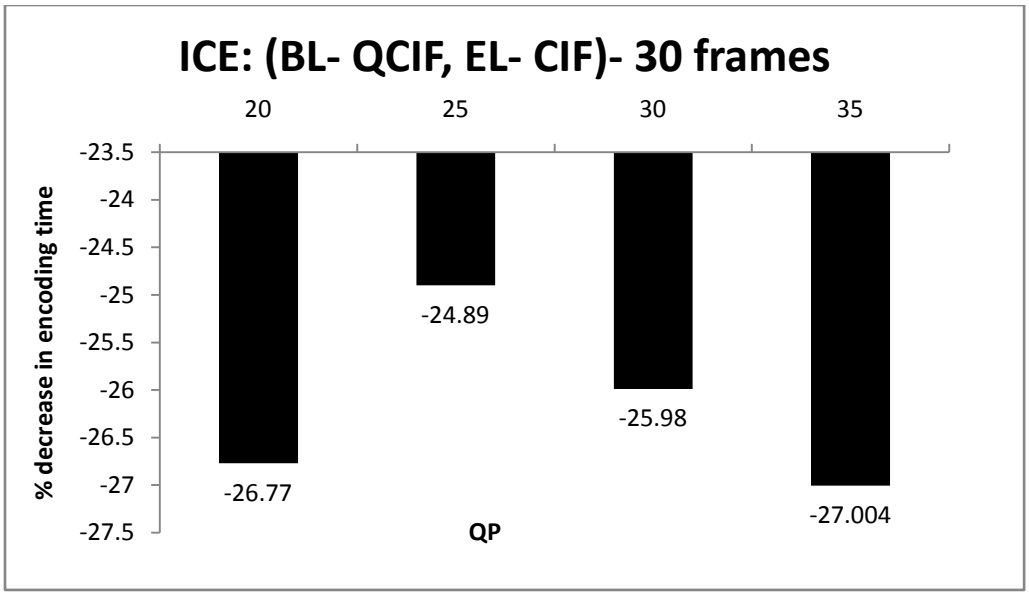



Figure 4-12: %decrease in encoding time vs. quantization parameter for Ice with BL- QCIF and EL- CIF

## 4.3 BD-PSNR

Bjøntegaard Delta PSNR (BD-PSNR) was proposed to objectively evaluate the coding efficiency of the video codecs [35] [40][41]. BD-PSNR provides a good evaluation of the rate-distortion (R-D) performance based on the R-D curve fitting. BD-PSNR is a curve fitting metric based on rate and distortion of the video sequence. However this does not take the encoder complexity into account. BD metrics tell more about the quality of the video sequence. Ideally, BD-PSNR should increase and BD-bitrate should decrease. BD-PSNR for the proposed algorithm when compared with the unaltered HM 10.0-dev-SHM are illustrated in the Figures 4-13 thru 4-18. It is observed that BD-PSNR has increased by 0.2dB to 0.3dB which implies that the qualities of the videos are not degraded when encoded using the proposed algorithm in the reference software.

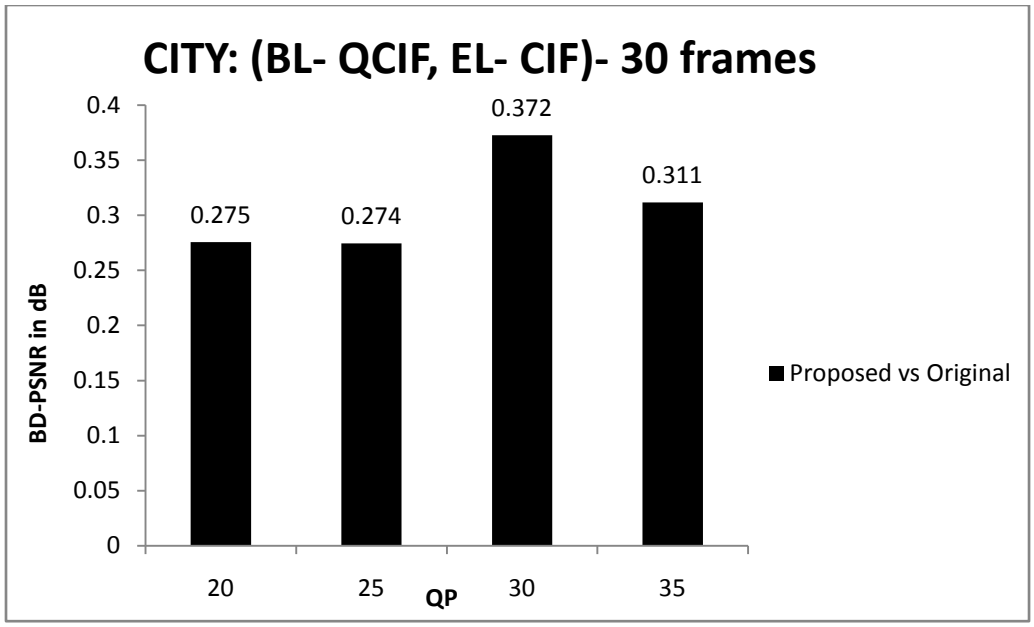



Figure 4-13: BD-PSNR vs. quantization parameter for City with BL-QCIF and EL-CIF

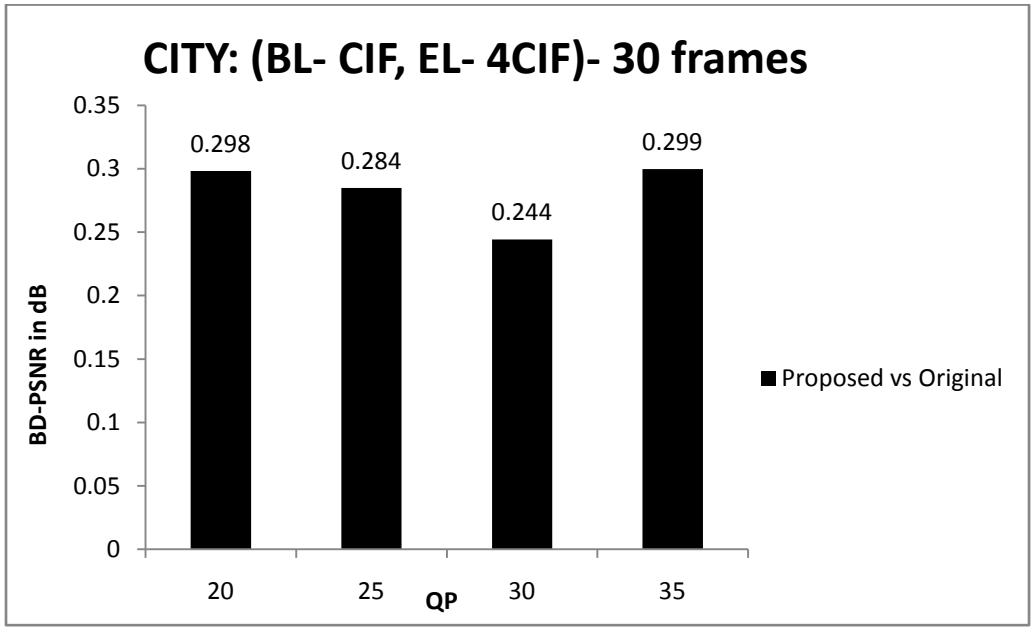



Figure 4-14: BD-PSNR vs. quantization parameter for City with BL-CIF and EL- 4CIF

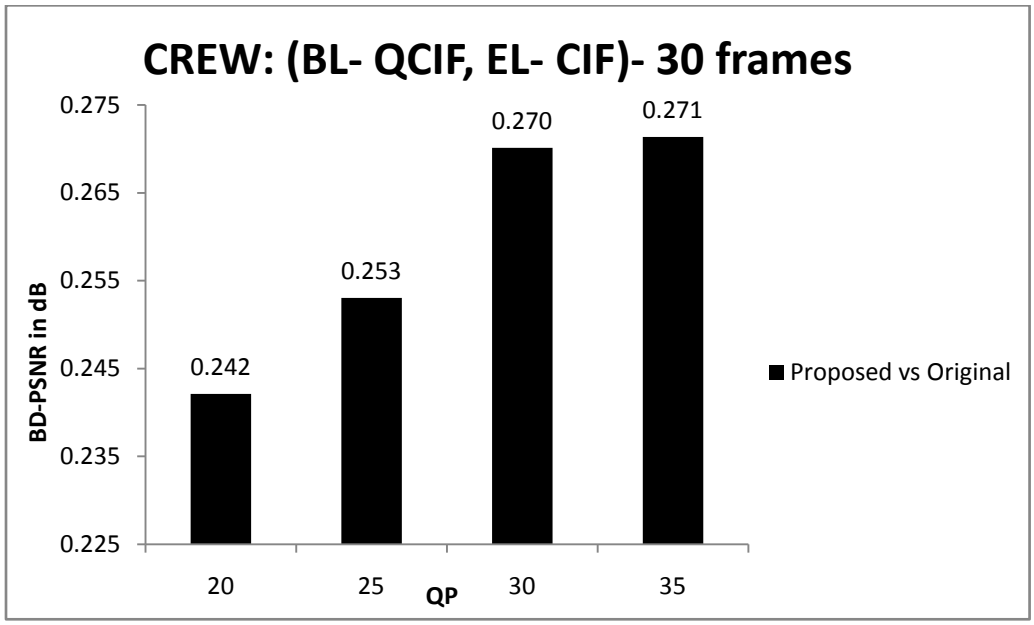



Figure 4-15: BD-PSNR vs. quantization parameter for Crew with BL- QCIF and EL- CIF

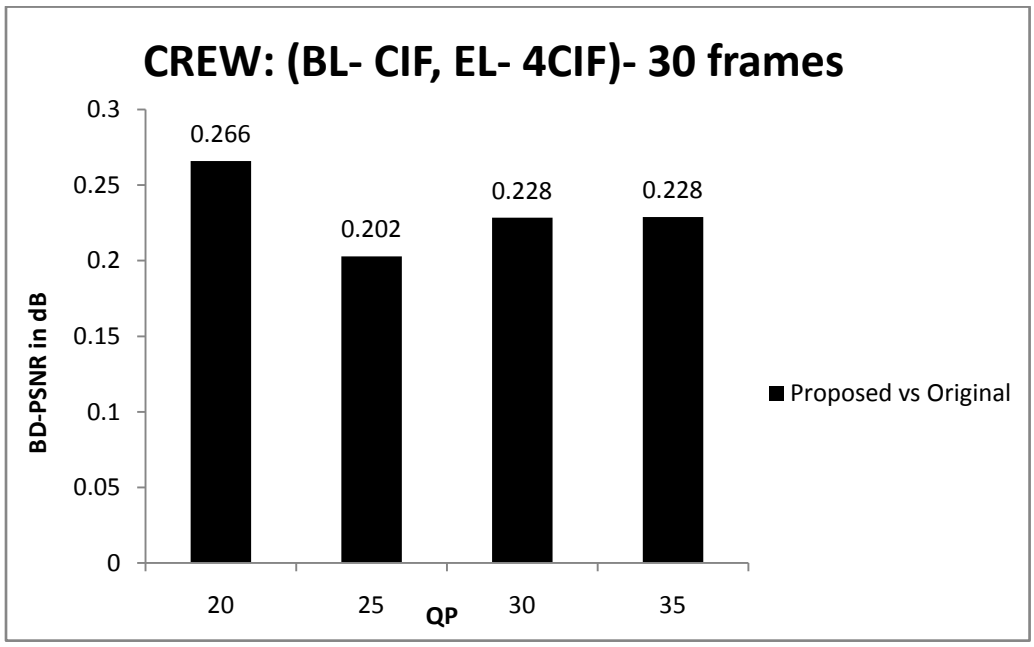



Figure 4-16: BD-PSNR vs. quantization parameter for Crew with BL- CIF and EL- 4CIF

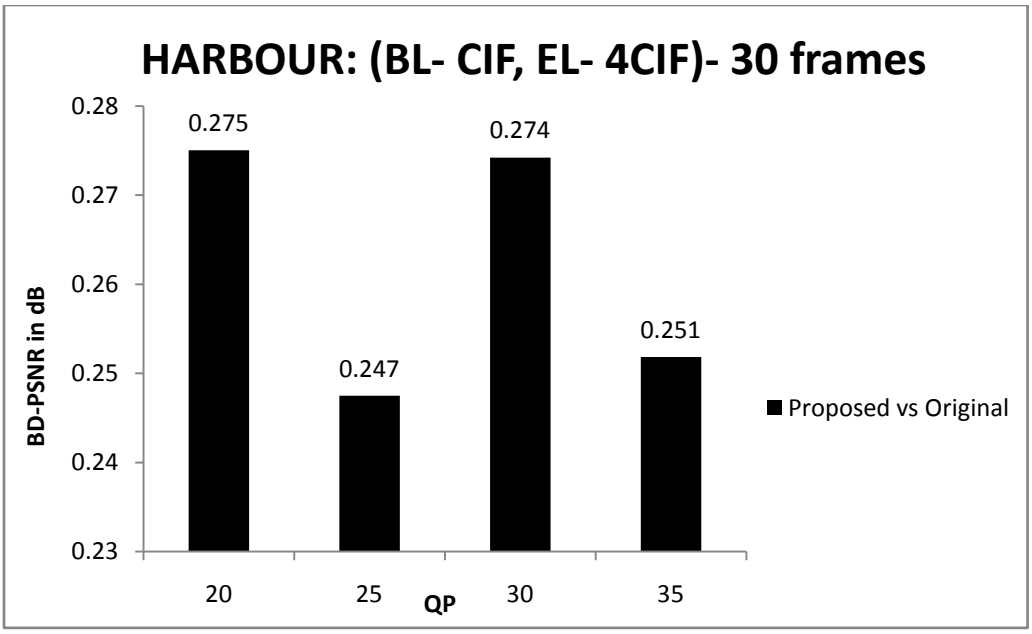



Figure 4-17: BD-PSNR vs. quantization parameter for Harbour with BL- CIF and EL- 4CIF

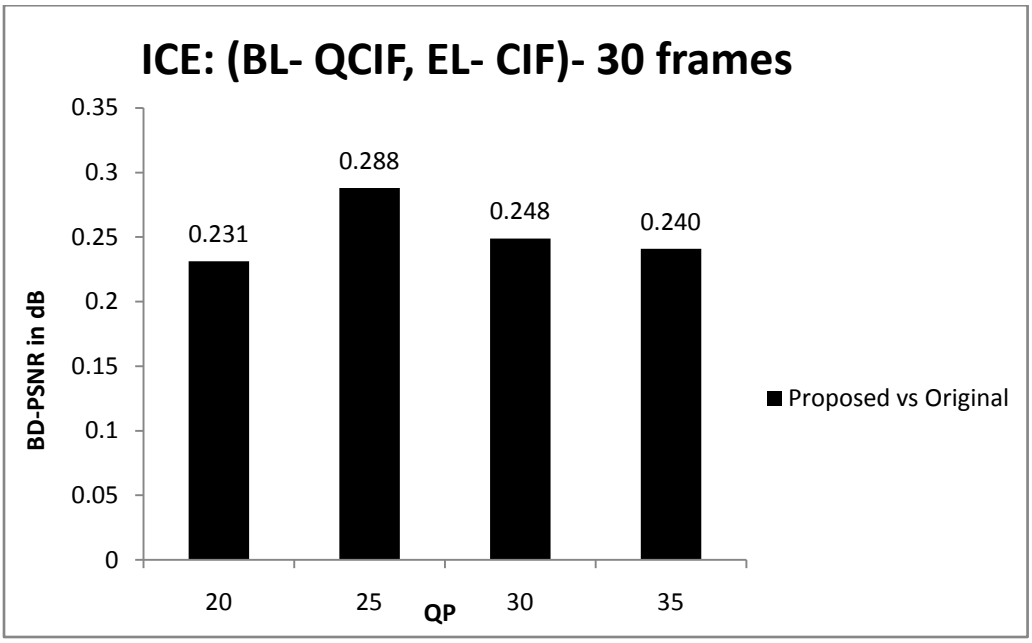



Figure 4-18: BD-PSNR vs. quantization parameter for Ice with BL- QCIF and EL- CIF

## 4.4 BD-Bitrate

BD-bitrate also determines the quality of the encoded video sequence similar to BD-PSNR. Ideally BD-bitrate should decrease for a good quality video [40][41]. Figures 4-19 thru 4-24 illustrate the BD-bitrate for the encoded bitstreams of proposed algorithm compared with the bitstreams encoded using the unaltered reference software. From the figures it can be seen that the BD-bitrate has decreased by 17% t0 29% which implies that the quality of the encoded bitstream using the proposed algorithm has not degraded compared to the bitstream encoded with the unaltered reference software.

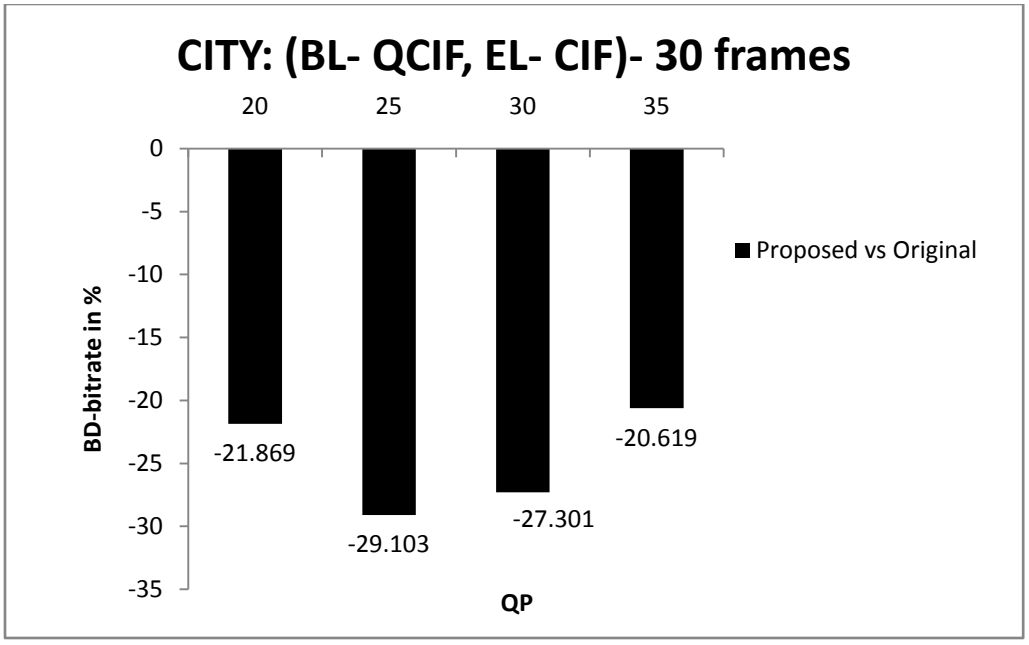



Figure 4-19: BD-bitrate vs. quantization parameter for City with BL- QCIF and EL- CIF

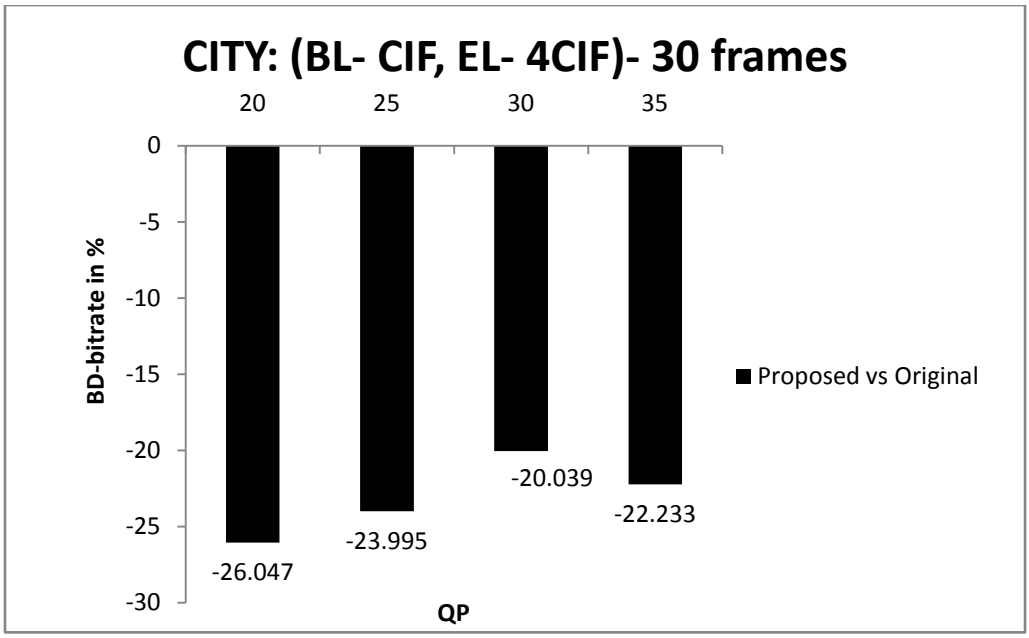



Figure 4-20: BD-bitrate vs. quantization parameter for City with BL- CIF and EL- 4CIF

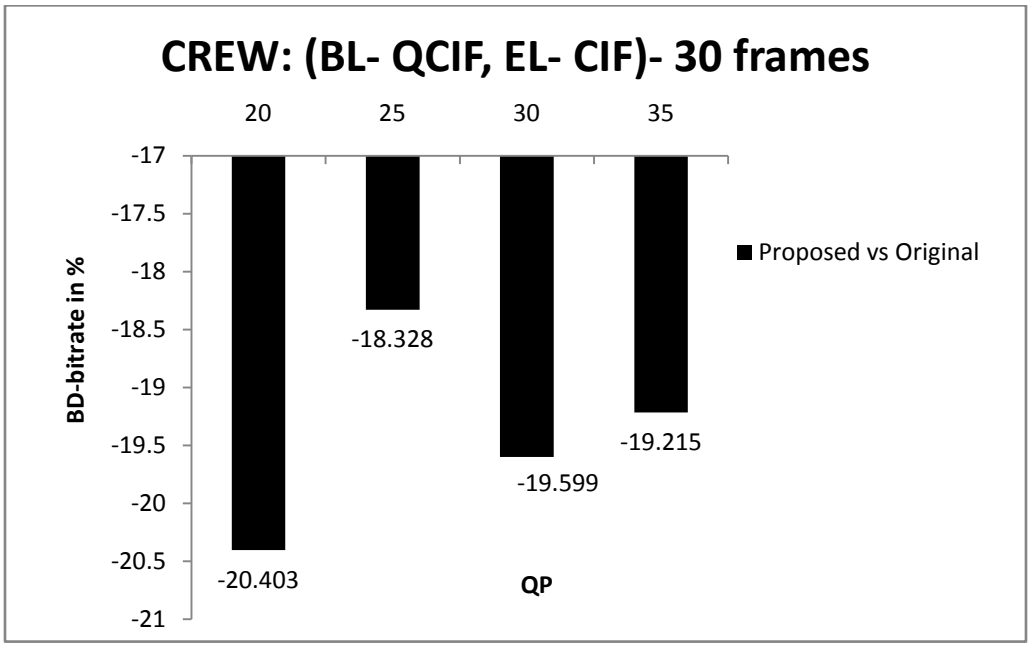



Figure 4-21: BD-bitrate vs. quantization parameter for Crew with BL- QCIF and EL- CIF

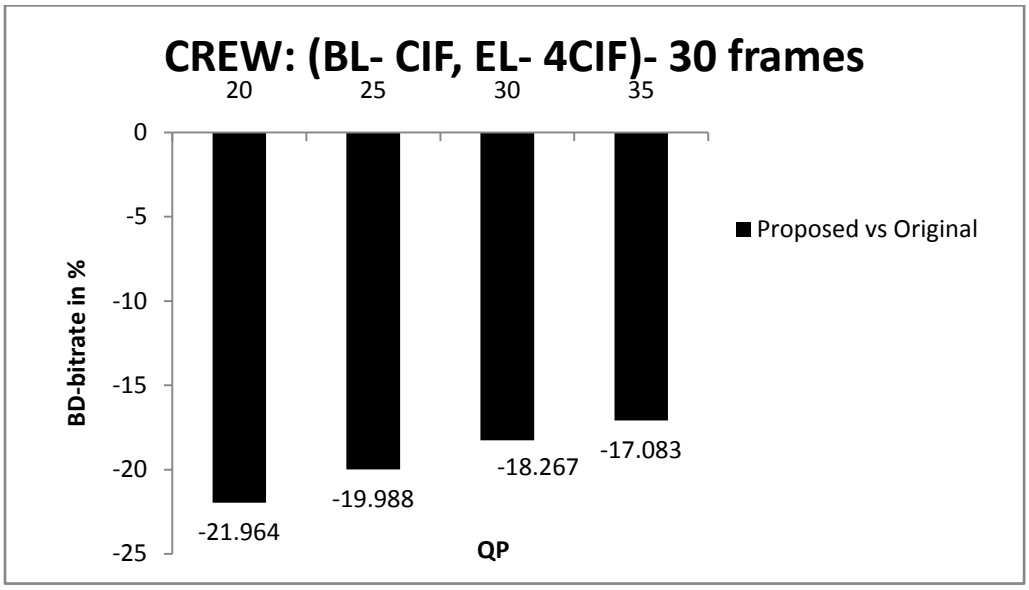



Figure 4-22: BD-bitrate vs. quantization parameter for Crew with BL- CIF and EL- 4CIF

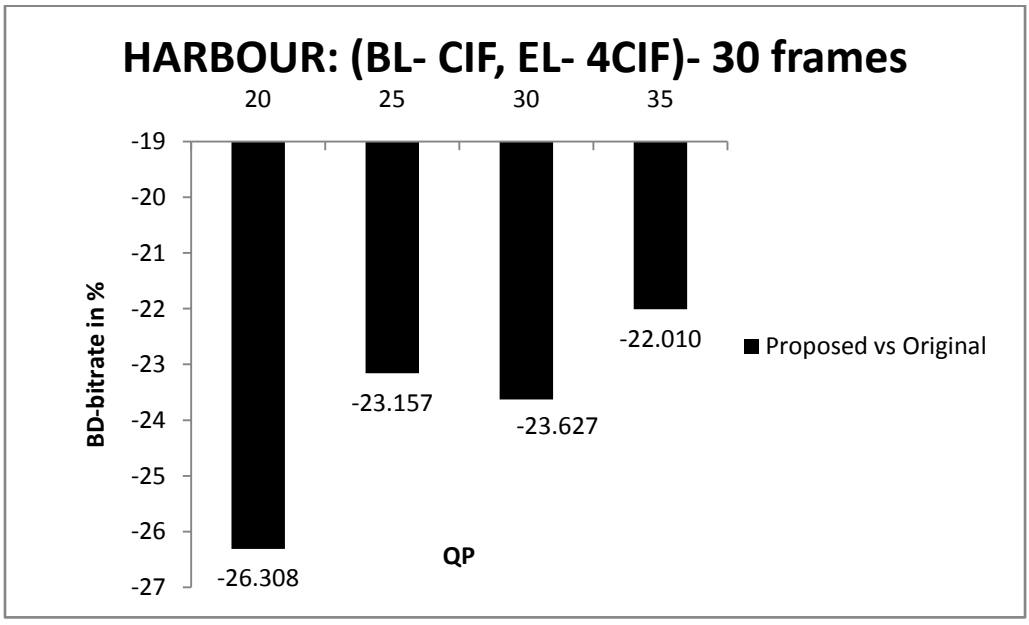



Figure 4-23: BD-bitrate vs. quantization parameter for Harbour for BL- CIF and EL- 4CIF

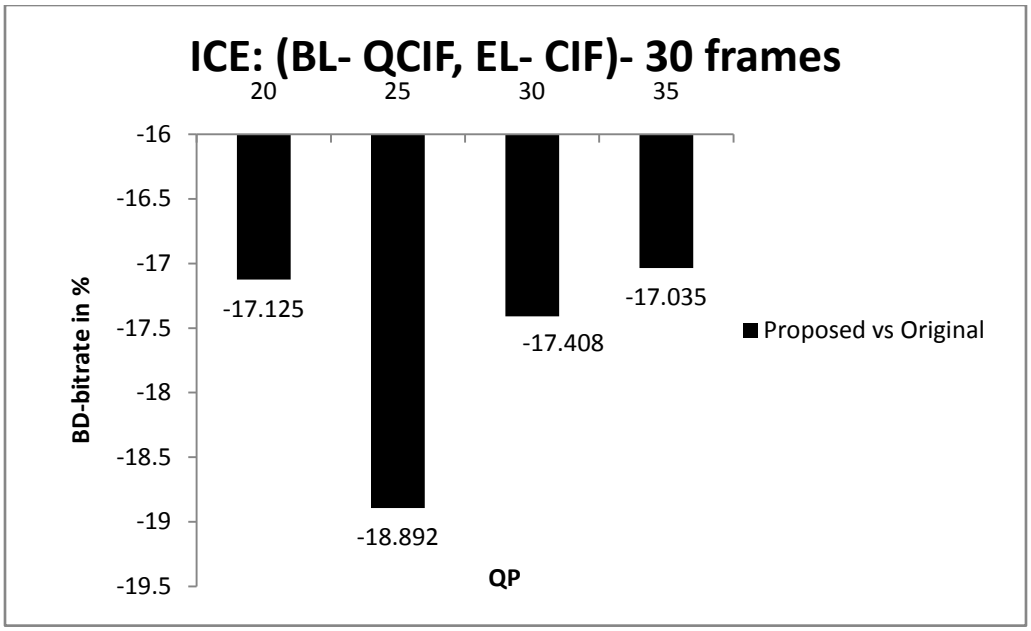



Figure 4-24: BD-bitrate vs. quantization parameter for Ice with BL- QCIF and EL- CIF

## 4.5 Bitrate vs. PSNR Plots

Bitrate vs PSNR plots for the proposed algorithm and the unaltered reference software are illustrated in Figure 4-25 thru 4-30. It can be observed from the graphs that there is 0.93% to 1.5% decrease in the PSNR for proposed algorithm compared to the unaltered reference software.

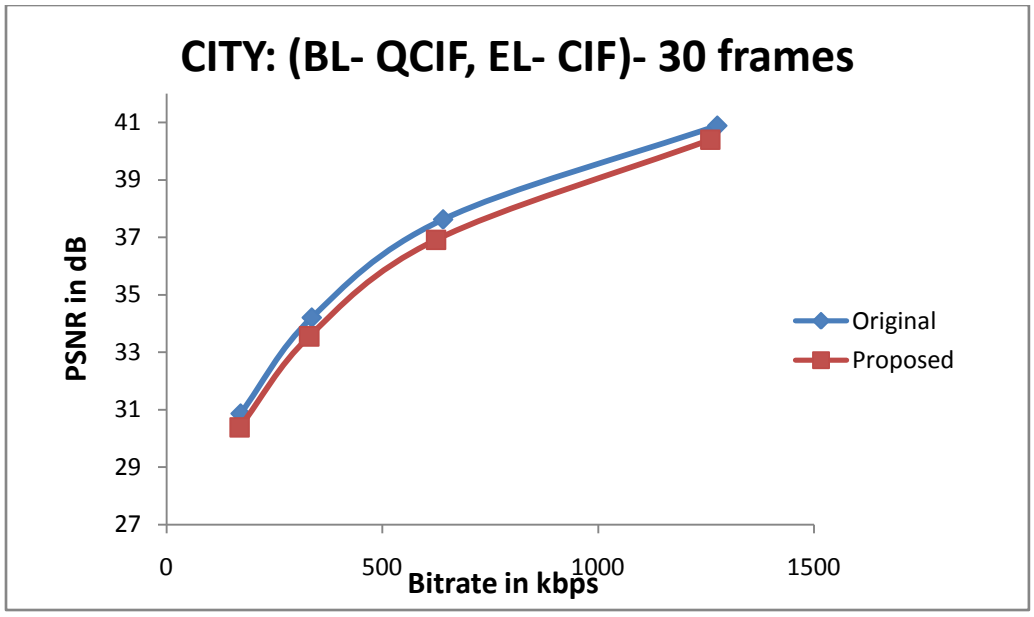



Figure 4-25: PSNR vs. bitrate for City with BL- QCIF and EL- CIF

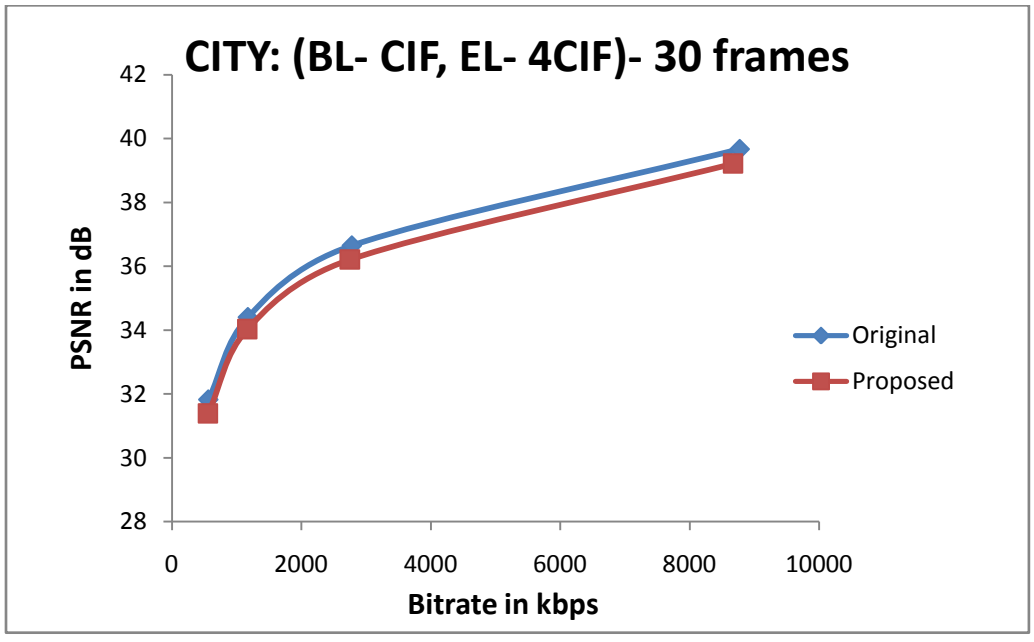



Figure 4-26: PSNR vs. bitrate for City with BL- CIF and EL- 4CIF

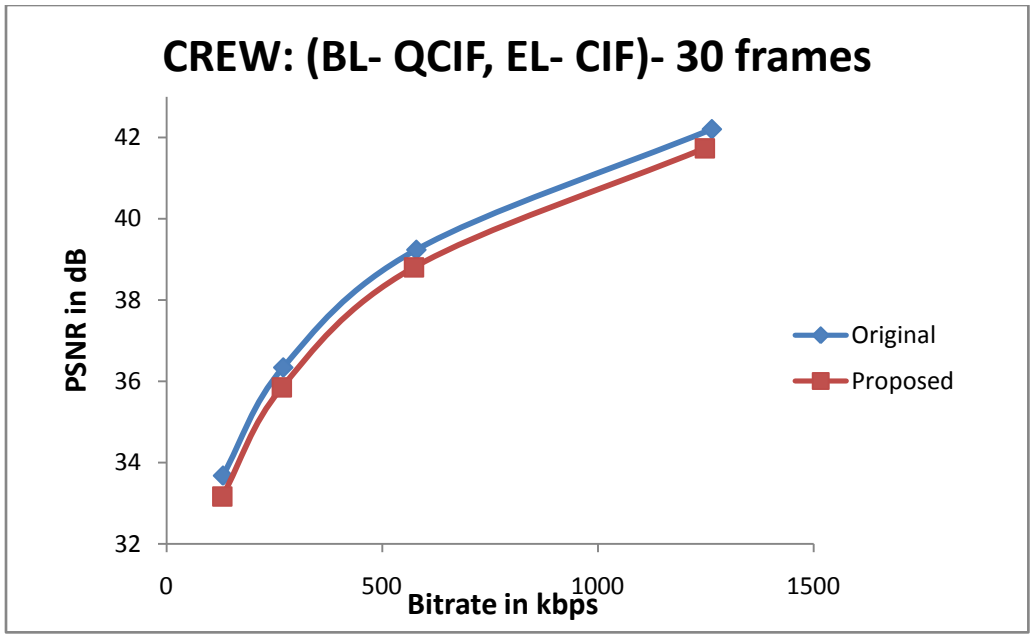



Figure 4-27: PSNR vs. bitrate for Crew with BL- QCIF and EL- CIF

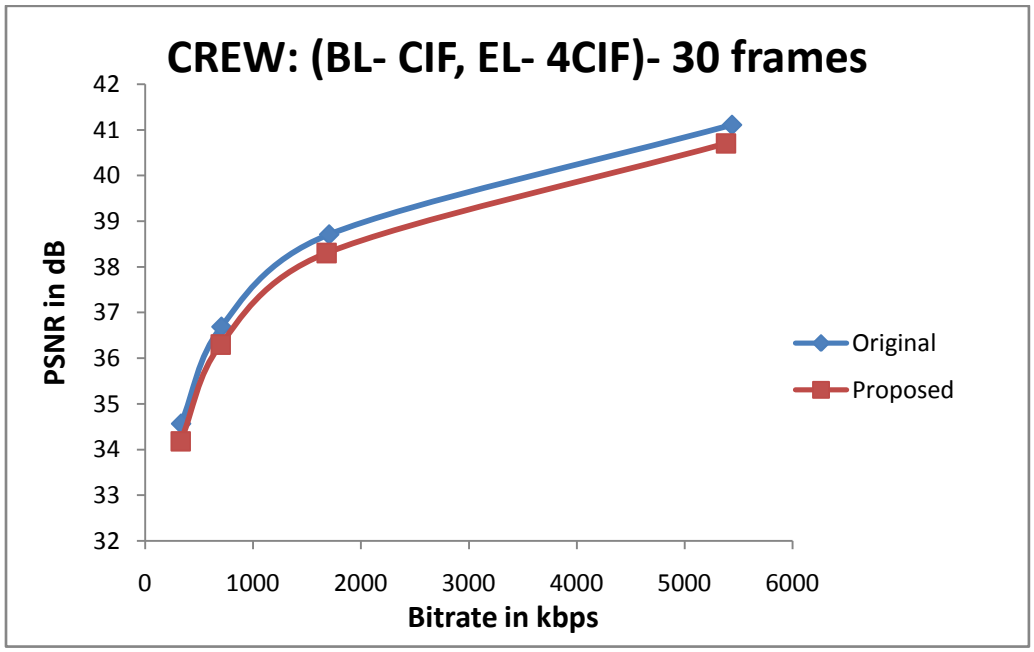



Figure 4-28: PSNR vs. bitrate for Crew with BL- CIF and EL- 4CIF

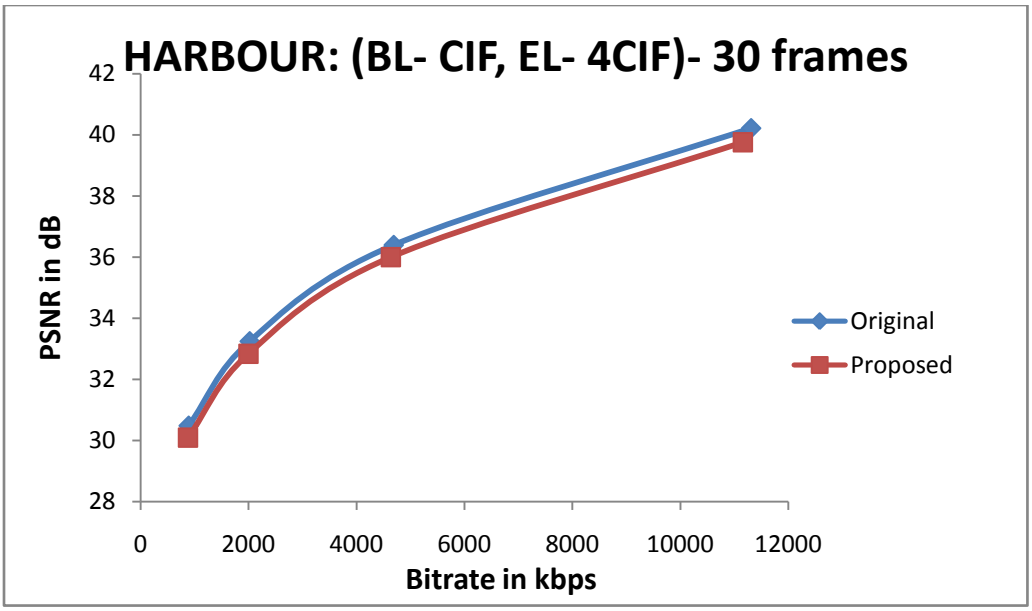



Figure 4-29: PSNR vs. bitrate for Harbour with BL- CIF and EL- 4CIF

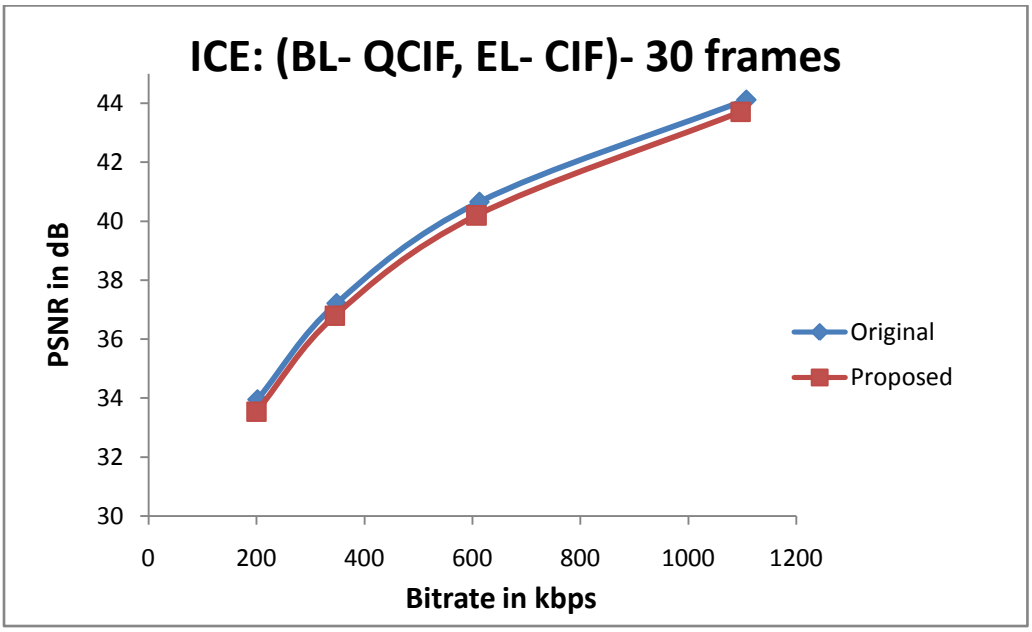



Figure 4-30: PSNR vs. bitrate for Ice with BL- QCIF and EL- CIF

## 4.6 Bitstream Size

The bitstream size obtained after encoding the video sequences using the proposed algorithm for inter-prediction of the enhancement layer and the unaltered reference software are plotted and compared. It can be observed from Figures 4-31 thru 4-36 that the bitstream size for the proposed algorithm has decreased by 0.88% to 1.3% compared to bitstream size obtained using the unaltered reference software.

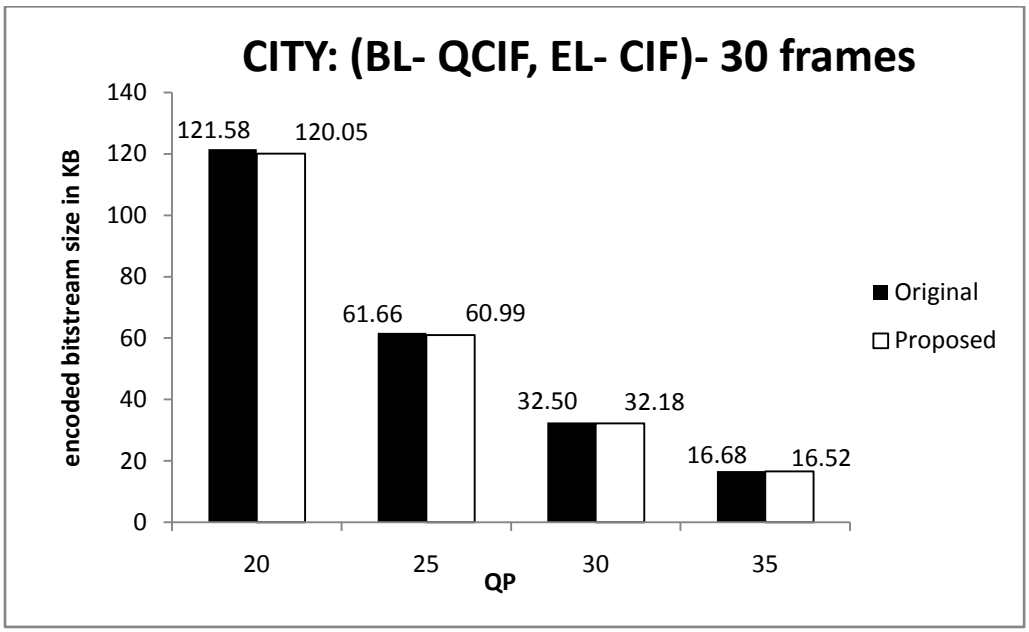



Figure 4-31: Encoded bitstream size vs. quantization parameter for City with BL- QCIF and EL- CIF

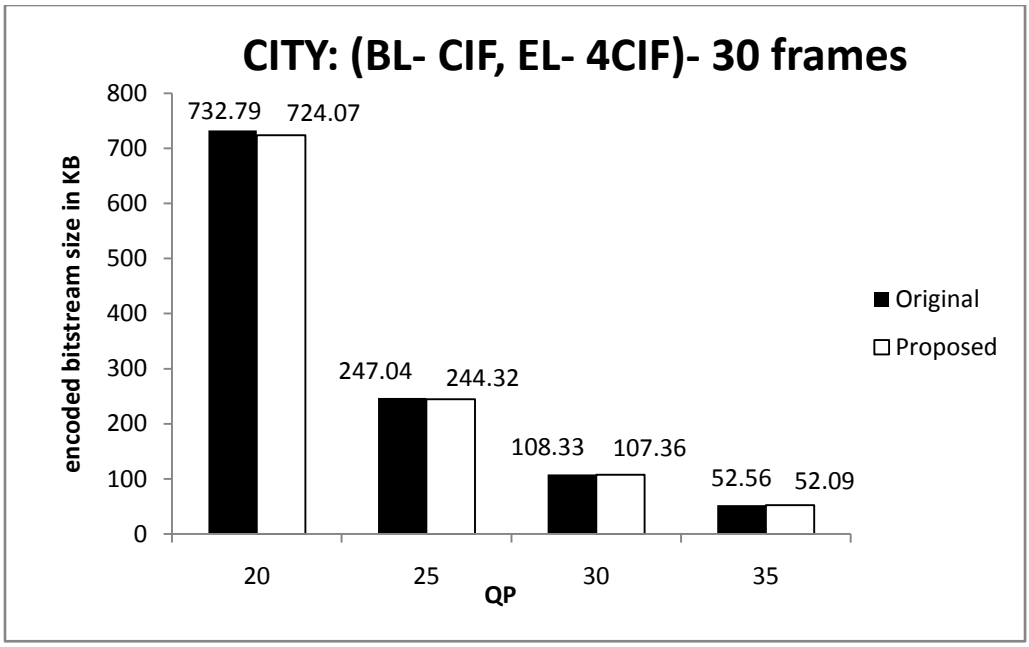



Figure 4-32: Encoded bitstream size vs. quantization parameter for City with BL- CIF and EL- 4CIF

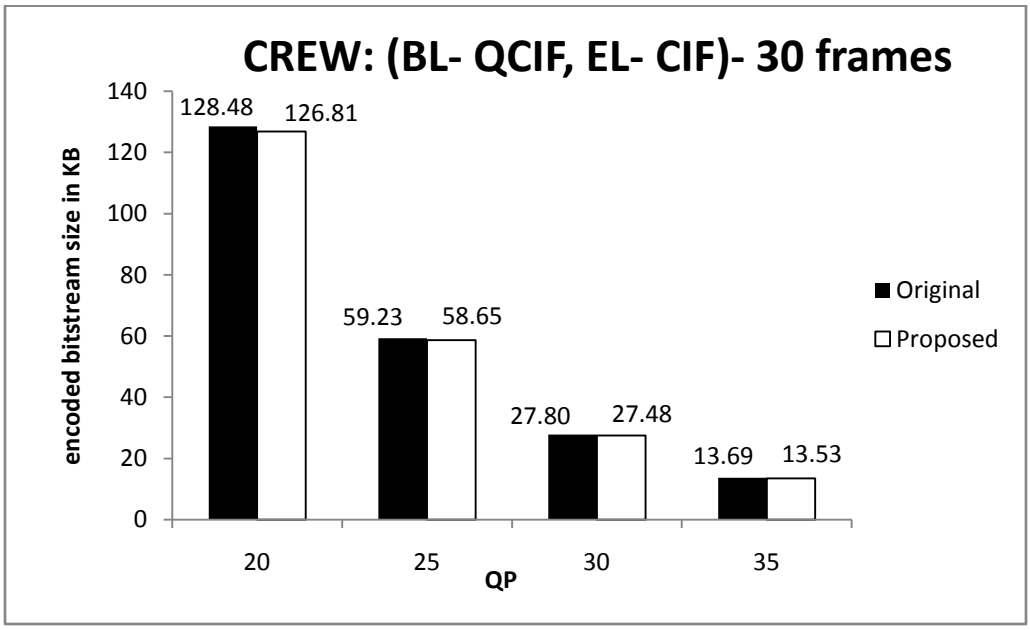



Figure 4-33: Encoded bitstream size vs. quantization parameter for BL- QCIF and EL- CIF

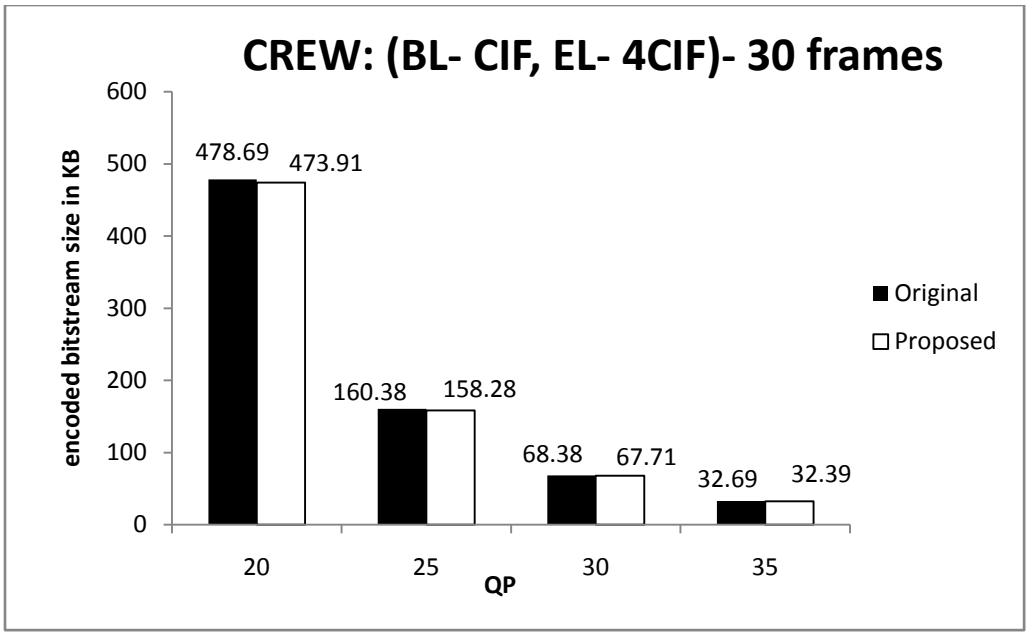



Figure 4-34: Encoded bitstream vs. quantization parameter for Crew with BL- CIF and EL- 4CIF

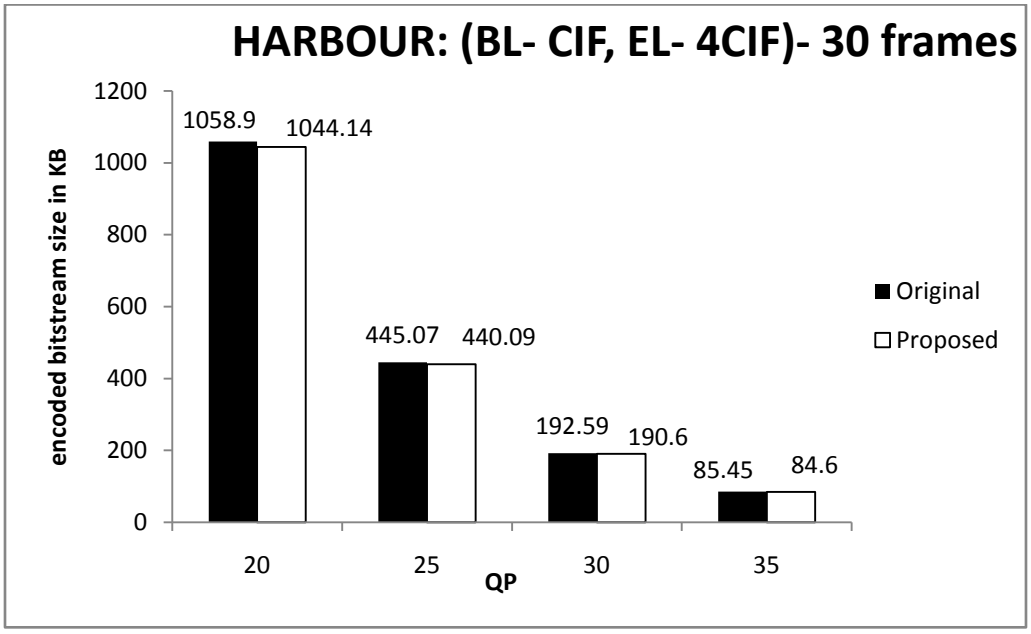



Figure 4-35: Encoded bitstream vs. quantization parameter for Harbour with BL- CIF and EL- 4CIF

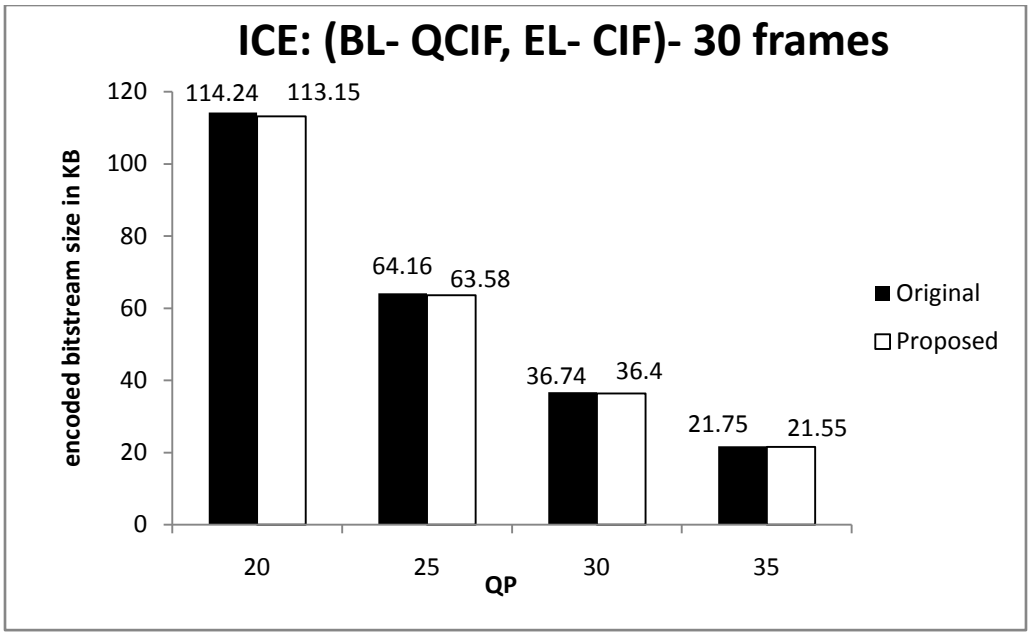



Figure 4-36: Encoded bitstream vs. quantization parameter for Ice with BL- QCIF and EL- CIF

## 4.7 Memory Usage

The memory used by the reference software encoder is 114212 bytes and the memory usage by the altered reference software with my algorithm is 114364 bytes. The extra memory used by my algorithm is 152 bytes.

## 4.8 Conclusions

It is observed from previous sections that there is slight drop in PSNR by 0.93% to 1.5% for the algorithm proposed for inter-prediction between the layers when compared with the unaltered reference software. Encoding time has decreased by 23% to 28% and the encoded bitstream size has reduced by 0.88% to 1.3% for the proposed algorithm. BD-PSNR has increased by only 0.2dB to 0.3dB and BD-bitrate has decreased by 17% to 29% when the proposed algorithm is compared with the unaltered reference

software. The extra memory used for my algorithm compared to the base algorithm is observed to be 152 bytes. The following chapter relates to the future work.

# Chapter 5

## Future Work

Time complexity in HEVC which is the base layer codec in scalable HEVC can be reduced by many other ways [40][41][42]. Early termination for inter-prediction mode decision can be implemented [40], fast residual quadtree encoding can be employed, fast intra-prediction algorithms [41][42]can be implemented along with fast inter-prediction algorithms to reduce the base layer encoding complexity. Early CU decision leading to early termination can also be employed to reduce the encoding complexity.

Along with the base layer encoding complexity reduction techniques, many other areas for inter-layer prediction can be explored [44][46]. Many mode decision algorithms to reduce the complexity in scalable video coding can be investigated [44][45][47]. Intra prediction, inter prediction complexity reduction algorithms in both base and enhancement layers along with fast inter-layer prediction algorithms can be applied for further reduction in the codec complexity to get better results.

# Appendix A

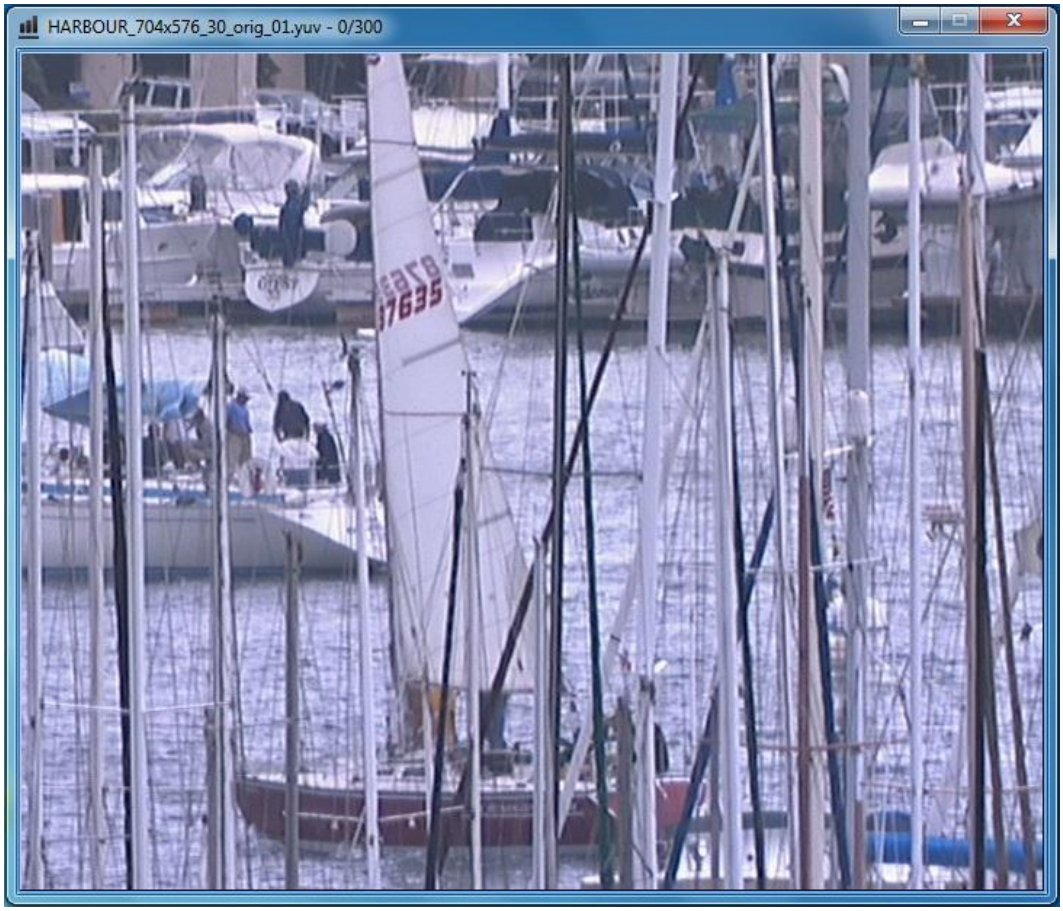
## Test Sequences [31]

This appendix displays a frame of each test sequence used.

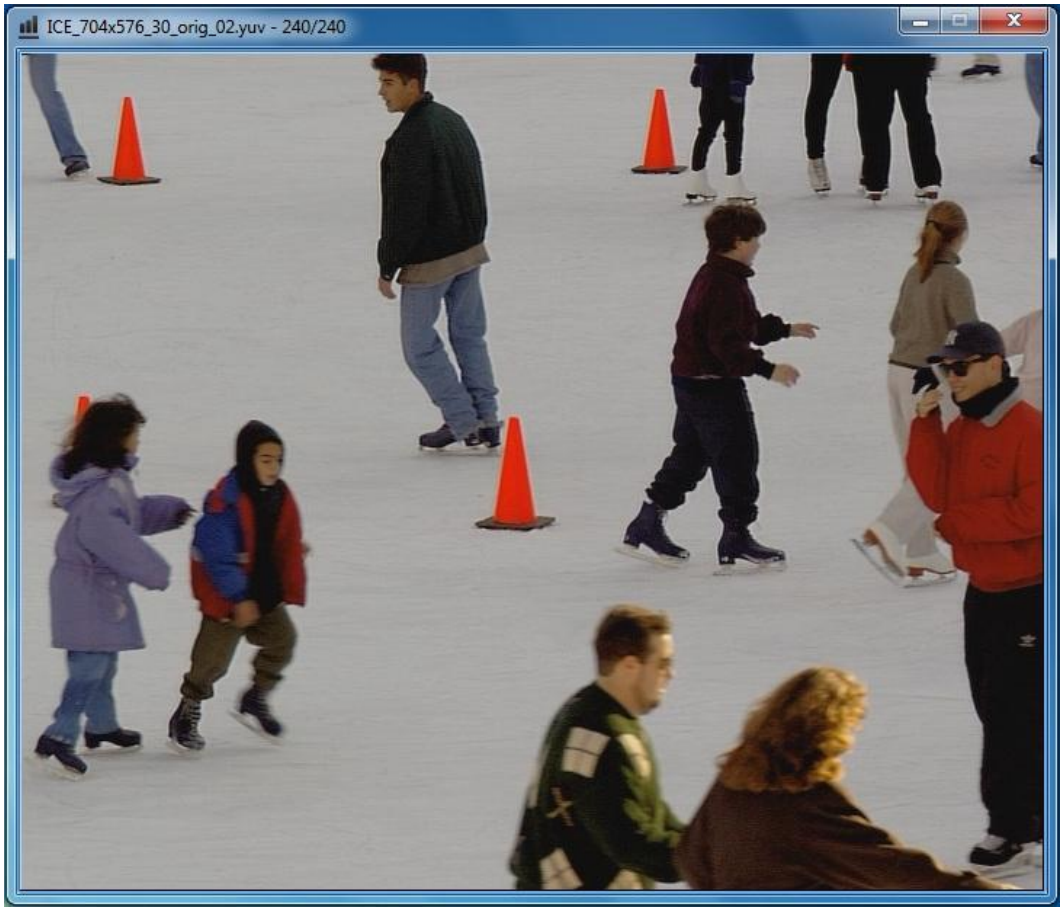
A.1 City

A.2 Crew

A.3 Harbour



A.4 Ice



# Appendix B

## Test Conditions

This appendix explains the platform used for running the simulations.

The code revision used for this work is revision HM 10.0-dev-SHM [30]. The work was carried out using an Intel(R) Xeon(R) CPU E3-1220 V2 with Microsoft Windows 7 64bit version running with 8 GB RAM at a speed of 3.1GHz.

# Appendix C

## BD-PSNR and BD-Bitrate [48][49][40]

This appendix defines the BD-PSNR and BD-bitrate metrics and lists the MATLAB code for their implementation.

BD-PSNR (Bjontegaard – PSNR) and BD-bit rate (Bjontegaard – bit rate) metrics are used to compute the average gain in PSNR and the average percent saving in bit rate between two rate-distortion graphs respectively and is an ITU-T approved metric [48]. This method was developed by Bjontegaard and is used to gauge compression algorithms from a visual aspect in media industry and referenced by many multimedia engineers. The MATLAB code is available online [49]

```
function avg_diff = bjontegaard(R1,PSNR1,R2,PSNR2,mode)
%BJONTEGAARD Bjontegaard metric calculation
% 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
%
% 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
% convert rates in logarithmic units
lR1 = log(R1);
```

```
lR2 = log(R2);
switch lower(mode)
case 'dsnr'
% PSNR method
p1 = polyfit(lR1,PSNR1,3);
p2 = polyfit(lR2,PSNR2,3);
% integration interval
min_int = min([lR1; lR2]);
max_int = max([lR1; lR2]);
% find integral
p_int1 = polyint(p1);
p_int2 = polyint(p2);
int1 = polyval(p_int1, max_int) - polyval(p_int1, min_int); 10
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,lR1,3);
p2 = polyfit(PSNR2,lR2,3);
% integration interval
min_int = min([PSNR1; PSNR2]);
max_int = max([PSNR1; 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

List of Acronyms

This appendix lists the acronyms used

AVC – Advanced Video Codec.

AMVP – Advanced motion vector prediction.

BL – Base Layer.

BO – Band Offset.

CABAC – Context Adaptive Binary arithmetic coding.

CAVLC – Context Adaptive Variable Length Coding.

CB – Coding Block.

CIF – Common Intermidiate Format.

CTB – Coding Tree Block.

CTU – Coding Tree Unit.

CU – Coding Unit.

DC – Direct Current.

DCT – Discrete Cosine Transform.

Diff – Difference.

DPB – Decoded Picture Buffer.

DST – Discrete Sine Transform.

DM – Direct Mode.

ED – Entropy Decoder.

EL – Enhancement Layer.

EO – Edge Offset.

Filt – Filter.

FIR – Finite Impulse Response.

GOP – Group of pictures.

HD – High Definition.

HDTV – High Definition Television.

HEB – High Efficiency Binarization.

HEVC – High Efficiency Video Coding.

HTB – High Throughput Binarization.

IEC – International Electro-technical Commission.

IP – Intra Prediction.

IQ – Inverse Quantization.

IT – Inverse Transform.

ITU-T – International Telecommunication Union-Telecommunications standardization sector.

ISO – International Standardization Organization.

JCT-VC – Joint Collaborative Team on Video Coding.

LCU – Largest Coding Unit.

LM – Linear Mode.

LP – Loop Filtering.

MANE – Media Aware Network Elements.

MC – Motion compensation.

MPEG – Moving Picture Experts Group.

MV – Motion Vector.

PB – Prediction Block.

PDA – Personal Digital Assistant.

PU – Prediction Unit.

QCIF – Quarter Common Intermediate Format.

QP – Quantization Parameter.

QVGA – Quarter Video Graphics Array.

ROI – Region Of Interest.

SAO – Sample Adaptive Offset.

SHVC – Scalable High efficiency Video Coding.

SNR – Signal to Noise Ratio.

SVC – Scalable Video Coding.

TB – Transform Block.

TU – Transform Unit.

TV – Television.

VCEG – Video Coding Experts Group.

VCL – Video Coding Layer.

VGA – Video Graphics Array.

1-D – 1 Dimensional.

2-D - 2 dimensional.

3D – 3 Dimensional.

4CIF – 4x CIF.

References


[1] I. Richardson, “The H.264 Advanced video Compression Standards”, Wiley, 2010.

[2] F. Wang, “Parallelization of Software MPEG Compression”, blog. [online]. Available: http://www.evl.uic.edu/fwang/mpeg.html (accessed on June 4th 2014).

[3] Video Coding for Low Bit Rate Communication, ITU-T Rec. H.263, Nov. 1995 (and subsequent editions).

[4] Generic Coding of Moving Pictures and Associated Audio Information- Part 2: Video, ITU-T Rec. H.262 and ISO/IEC 13818-2 (MPEG 2 Video), ITU-T and ISO/IEC JTC 1, Nov. 1994.

[5] A. Uhl, “Compression Technologies and Multimedia Data Formats”, Lecture Notes, Department of Computer Sciences, University of Salzburg. [online]. Available: http://www.cosy.sbg.ac.at/~uhl/ctmdf.pdf (accessed on June 4th 2014).

[6] K.R. Rao, D.N. Kim and J.J. Hwang, “Video Coding Standards: AVS China, H.264/MPEG-4 Part10, HEVC, VP6, DIRAC and VC-1”, Springer, 2014.

[7] J. Chen et al, “Design of Digital Video Coding Systems a complete compressed domain approach”, Marcel Dekker, 2002.

[8] G.J. Sullivan et al, “Overview of the High Efficiency Video Coding (HEVC) standard”, IEEE Trans. on CSVT, vol.22, Issue 12, pp. 1649-1668, Dec. 2012.

[9] N. Ling, “High efficiency video coding and its 3D extension: A research perspective”, 2012 7th IEEE Conference on Industrial Electronics and Applications (ICIEA), pp. 2150-2155, July 2012.

[10] B. Bross et al, “High efficiency video coding (HEVC) text specification draft 10”, 12th Meeting Geneva, January 2013. [online]. Available: http://phenix.it-sudparis.eu/jct/doc_end_user/current_document.php?id=7243 (accessed on June 9th 2014).

[11] M.T. Pourazad et al, “HEVC: The New Gold Standard for Video Compression”, IEEE consumer electronics magazine, pp. 36 – 46, July 2012. [online]. Available: http://dml.ece.ubc.ca/doc/HEVC_2012.pdf (accessed on June 8 2014).

[12] Moto, “HEVC- What are CTU, CU, CTB, CB, PB and TB?”, CODE: Sequoia, worgpress.com site, blog.[online]. Available: http://codesequoia.wordpress.com/2012/10/28/hevc-ctu-cu-ctb-cb-pb-and-tb/ (accessed on June 9th 2014).

[13] S. Riabstev, “Detailed overview of HEVC/H.265”, [online]. Available: https://app.box.com/s/rxxxzr1a1lnh7709yvih (accessed on June 12 2014).

[14] A. Luthra and P. Topiwala, “Overview of the H.264/AVC video coding standard”, Proceedings of SPIE- The International Society for Optical Engineering, vol.5203, pp. 417-431, Applications of Digital Image Processing XXVI, Aug. 2003.

[15] X. Zhang et al, “New chroma intra prediction modes based on linear model for HEVC” 19th IEEE International Conference on Image Processing (ICIP), pp. 197-200, Oct. 2012.

[16] H. Schwarz, D. Marpe, and T. Wiegand, “Overview of Scalable Video Coding Extension of the H.264/AVC Standard”, IEEE Trans. on CSVT, vol. 17, No.9, pp.1051-8215, Sept. 2007.

[17] H. Schwarz and T. Wiegand, “The Scalable Video Coding Amendment of the H.264/AVC Standard”, csdn.net, world pharos, blog.[online].Available: http://blog.csdn.net/worldpharos/article/details/3369933 (accessed on June 20th 2014).

[18] I. Unanue et al, “A Tutorial on H.264/SVC Scalable Video Coding and its Tradeoff between Quality, Coding Efficiency and Performance”, Recent Advances on

Video Coding.[online]. Available: http://www.doc88.com/p-516795349043.html (accessed on June 20 2014).

[19] "Digital Video Broadcasting", blog. [online]. Available: http://www.ntu.edu.sg/home/elpchau/research%20video%20broadcast.htm (accessed on June 21 2014).

[20] A.Eleftheriadis, M.R. Civanlar and O. Shapiro, "Multipoint videoconferencing with scalable video coding", J. Zhejiang Univ. Sci. A, vol. 7, no. 5, pp. 696-705, May 2006.

[21] T. Schierl and T. Wiegand,"Mobile video transmission using SVC", IEEE Trans. on CSVT, vol. 17, no. 9, pp. Sept. 2007.

[22] M. Wien, R. Cazoulat, A. Hutter and P. Amon, "Real time system for adaptive video streaming based on SVC", IEEE Trans on CSVT, vol. 17, no. 9, pp. 1227-1237, Sept. 2007.

[23] M. Vochin et al, "Scalability analysis of a media aware network element", 20th EUSIPCO 2012 proceedings, pp. 2213-2217, Aug. 2012.

[24] F. Ziliani,"The importance of scalability in video surveillance architectures" IEEE International Symposium on ICDP, pp. 29-32, June 2005.

[25] G.J. Sullivan et al, "Standardized extensions of High Efficiency Video Coding (HEVC)", IEEE J-STSP, vol. 7, no. 6, pp. 1001 – 1016, Dec. 2013.

[26] J. Chen et al, "Scalable high efficiency video coding draft 3", 14th JCT-VC meeting: Vienna, Austria, Document JCTVC-N1008, July 25 – Aug. 2 2013.

[27] E. Alshina et al, "Suggested up-sampling filter design for tool experiments on HEVC scalable extension", 11th JCT-VC meeting: Shanghai, china, Document JCTVC-K0378, Oct. 10 – 19, 2012.

[28] T. Hinz et al, “An HEVC Extension for Spatial and Quality Scalable Video Coding”, Proceedings of SPIE, vol. 8666, pp. 866605-1 to 866605-16, Feb. 2013.

[29] C.A.Segall and G.J. Sullivan, “Spatial Scalability within the H.264/AVC Scalable Video Coding Extension”, IEEE Trans. on CSVT, vol. 17, pp. 1121-1135, Sept. 2007.

[30] SHVC software and software manual: The source code for the software and its manual is available in the following SVN repository.[online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_SHVCSoftware/ (accessed on June 26th 2014).

[31] Test sequences for scalable video coding. [online]. Available: ftp://ftp.tnt.uni-hannover.de/pub/svc/testsequences/ (accessed on June 26, 2014).

[32] J. Boyce et al, “Information for HEVC scalability extension”, JCT-VC, Document JCTVC-G078, Nov. 2011.[online]. Available: http://phenix.int-evry.fr/jct/doc_end_user/current_document.php?id=3327 (accessed on July 2, 2014).

[33] P.Heke et al, “A Scalable video coding extension of HEVC”, IEEE DCC, pp. 201-210, Mar. 2013.

[34] A. Abdelazim, W,Masri and B. Noaman, “Motion Estimation Optimization tools for Emerging High Efficiency Video Coding (HEVC)”, Proc. Of SPIE-IS&T Electronic Imaging, vol. 9029, pp. 051-058, Jan. 2014.

[35] C-L. Su, T-M. Che and C-Y. Huang, “Cluster-Based Motion Estimation Algorithm With Low Memory and Bandwidth Requirements for H.264/AVC Scalable Extension”, IEEE Trans. on CSVT, vol. 24, no. 6, pp. 1016-1024, June 2014. (Good references in the reference section of this paper).

[36] C.M. Huang et al, “Error Resilience supporting bi-directional frame recovery for video streaming”, Proceedings of IEEE CIP, vol.1, pp.537-540, 2004.

[37] X. Li et al, “Rate-Complexity-Distortion evaluation for hybrid video coding”, IEEE Trans. on CSVT, vol. 21, pp. 957 - 970, July 2011.

[38] SHVC bitstream layer parser.[online]. Available: http://r2d2n3po.tistory.com/70

[39] All JCT-VC documents can be accessed. [online]. Available: http://phenix.int-evry.fr/jct/doc_end_user/current_meeting.php?id_meeting=154&type_order=&sql_type=document_number

[40] K. Shah, “Reducing the complexity of Inter-prediction mode decision for HEVC”, M.S. Thesis, University of Texas at Arlington, UMI Dissertation Publishing, April 2014. [online]. Available: http://www-ee.uta.edu/Dip/Courses/EE5359/KushalShah_Thesis.pdf (accessed on July 1st 2014).

[41] S.Vasudevan, “Fast intra prediction and fast residual quadtree encoding implementation in HEVC”, M.S. Thesis, University of Texas at Arlington, UMI Dissertation Publishing, Nov. 2013. [online]. Available http://www-ee.uta.edu/Dip/Courses/EE5359/index.html (accessed on July 2nd 2014).

[42] D.P. Kumar, “Intra Frame Luma Prediction using Neural Networks in HEVC”, M.S. Thesis, University of Texas at Arlington, UMI Dissertation Publishing, May 2013. [online].Available: http://www-ee.uta.edu/Dip/Courses/EE5359/Dilip_Thesis_Document.pdf (accessed on July 2nd 2014).

[43] A. Hassan Thungaraj, “Encoder complexity reduction with selective motion merge in HEVC”, M.S. Thesis, University of Texas at Arlington, UMI Dissertation

Publishing, Aug. 2013. [online].Available: http://www-ee.uta.edu/Dip/Courses/EE5359

[44] B. Lee et al, “A fast mode selection scheme in inter-layer prediction of H.264 scalable extension coding” ISBMSB, pp. 1-5, Mar. 2008.

[45] S. Lee and S.J. Park, “On improving the fast mode decision of enhancement layer in scalable video coding extensions of H.264/AVC”, International journal of Control and Automation, Vo. 5, No. 3, pp. 207-215, Sept. 2012.

[46] S.V. Leuven et al, “ Generic techniques to reduce SVC enhancement layer encoding complexity”, IEEE Trans. on Consumer Electronics, vol. 57, issue 2, pp. 827-832, June 2011.

[47] X. Lu and G.R. Martin, “Fast mode decision algorithm for H.264/AVC scalable video coding extension”, IEEE Trans. on CSVT, vol. 23, issue 5, pp. 846-855, May 2013.

[48] G. Bjontegaard, “Calculation of average PSNR differences between RD-curves”, Q6/SG16, Video Coding Experts Group (VCEG), 2-4 April 2001.

[49] BD metrics code. [online]. Accessed: http://www.mathworks.com/matlabcentral/fileexchange/27798-bjontegaardmetric/content/bjontegaard.m (accessed on July 10th 2014).

[50] K. Iguchi et al, “HEVC encoder for super hi-vision”, IEEE ICCE, pp. 61-62, Las Vegas, NV, Jan. 2014.

[51] H. Schwarz et al, “ Extension of High Efficiency Video Coding (HEVC) for multiview video and depth data”, 19th IEEE ICIP, pp. 205-208, Sept. 30 – Oct. 3, 2012.

## Biographical Information

Karuna Gubbi Shivashankar Sastri was born in Bangalore, Karnataka, India in 1990. After completing the schooling at Poorna Prajna Education Centre, Bangalore in 2006, she went to obtain her bachelor's degree in engineering in Medical Electronics from Dayanada Sagar College of Engineering in Bangalore in 2012.

She joined the University of Texas at Arlington to pursue her master's degree in Electrical Engineering in 2013. This was around the time she joined the Multimedia Processing Lab.