FAST INTRA MODE DECISION IN HIGH EFFICIENCY VIDEO CODING

by

HARSHDEEP BRAHMASURY JAIN

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

Foremost, I would like to express my sincere gratitude to Dr. K. R. Rao from the bottom of my heart for being a guide, mentor and a constant source of encouragement throughout my thesis. He has been and will be a great source of inspiration for his zeal, resourcefulness and ideas. I am very grateful for the opportunity to work under his guidance.

I would like to thank Dr. Jonathan Bredow and Dr. Alan Davis for serving on my committee. I would like to thank Dr. Zhan Ma, Senior Standards Researcher, Samsung Research America Dallas whose research has been an inspiration for my work.

I would also like to thank my MPL lab-mates: Abhijith Jagannath, Kushal Shah and Vinoothna Gajula for providing valuable inputs throughout my research. Also I thank Uday Giridhar for his support and inputs.

Finally and most importantly, I would like to acknowledge my parents, Sudha Brahmasury and Brahmasury Anantarajaiah and my adorable sister Nisha for their unwavering love and support throughout my entire life.

July 8, 2014

Abstract

FAST INTRA MODE DECISION IN HIGH EFFICIENCY VIDEO CODING

Harshdeep Brahmasury Jain, MS

The University of Texas at Arlington, 2014

Supervising Professor: K. R. Rao

In this thesis a CU early termination algorithm with a fast intra prediction algorithm is proposed that terminates complete full search prediction for the CU and replaced by CU early termination algorithm which determines the complexity of the CU block then on sent decision is made to further split or non-split the CU. This is followed by a PU mode decision to find the optimal modes prediction mode from 35 prediction modes. This includes a two-step process: firstly calculating the Sum of Absolute Differences (SAD) of all the modes by down sampling method and secondly applying a three step search algorithm to remove unnecessary modes. This is followed by early RDOQ (Rate Distortion Optimization Quantization) termination algorithm to further reduce the encoding time. Experimental results based on several video test sequences suggest a decrease of about 35%-48% in encoding time is achieved with implementation of the proposed CU early termination algorithm and fast intra mode decision algorithm for intra predication mode decision with negligible degradation in peak signal to noise ratio (PSNR). Metrics such as BD-bitrate (Bjøntegaard Delta bitrate), BD-PSNR (Bjøntegaard Delta Peak Signal to Noise Ratio) and RD curve (Rate Distortion) are also used.

Table of Contents

List of Illustrations

List of Tables

Chapter 1

Introduction

1.1 Significance

Innovations in the communication systems have been tremendous in the last decade. Technology in communication systems has transformed from having only analog television via cable, satellite with availability of only a few channels or mobile phones that can only make voice calls or internet connections that are slow, mostly connected through a dial up modem connected via telephone lines.

Today the world has transformed into the so called “digital age” or “electronic age”, where mobile phones are called smart phones because they not only make phone calls but are also used for web browsing, sending emails, watching/capturing videos, transfer data, navigation purposes and take pictures. Digital television sets have become more compact with availability of regional and international channels with HD (High Definition) quality. Data is stored on re-writable DVDs, Blu-ray discs and hard disks which are light weight, portable with huge space for storage [2]. In this fast growing world of communications, data compression is still one of the most essential components in any multimedia system. Modern data compression techniques offer the possibility to store or transmit the vast amounts of data necessary to represent digital videos and images in an efficient and robust way.

Compression is the process of removing redundant information and representing data with fewer bits than the original information would use. It is useful because it helps to reduce the consumption of expensive resources such as data storage on hard disks and transmission bandwidths [2]. Hence, research is still going on in the field of compression techniques to enable real-time data transmission using less resources. Compression techniques are categorized as lossless or lossy. Lossless compression is possible because most of the real-world data has statistical redundancy. If the data has been losslessly compressed, the original data can be recovered with no loss. Lossless compression exploits statistical redundancy and represents data with more fidelity and less error [2]. It is beneficial in areas like text compression and audio compression. Lossy compression involves some information loss, so the data cannot be recovered exactly. It is applied in areas where data distorion is tolerable like video compression, image compression and some types of audio compression. Lossy image compression is used in digital cameras, to increase the storage capacity with less degradation of picture quality. Similarly lossy video compression is used on DVDs, Blu-ray disks [5], Internet telephony using MPEG-2 [2], H.264 [2] and HEVC (High Efficiency Video Coding) [1].

Video sequences contain a significant amount of statistical and subjective redundancies within, and between frames. The ultimate goal of a video source coding is bit rate reduction for storage and transmission by exploring both statistical (spatial) and subjective (temporal) redundancies and to encode a “minimum set” of information using entropy coding techniques [3]. The volume of data in multimedia signals is very high. For example, to represent 2 minutes of CD-quality music (44,100 samples per second, 16 bits per sample) requires more than 84 million bits. For video signals to represent 1 second of video without compression (using CCIR 601 format) [2], more than 20 Mbytes or 160 Mbits is required [2]. This data indicates the importance of compression for multimedia signals.

Multimedia consumer applications have a very large market. The revenues involved in digital TV broadcasting and DVD, Blu-ray distributions are substantial. Thus standardization of video coding is essential. Standards simplify inter-operability between encoders and decoders from different manufacturers, they make it possible for different vendors to build platforms that incorporate video codecs, audio codecs, security and rights management and they all interact in well-defined and consistent ways. There are numerous video compression standards, both open source and proprietary, depending on the applications and end-usage. Figure 1-1 shows the evolution of the video codec standards from the 90’s till today.

The High Efficiency Video Coding (HEVC) standard is the most recent joint video project of the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG) standardization organizations, working together in a partnership known as the Joint Collaborative Team on Video Coding (JCT-VC) [1]. However, an increasing diversity of services, the growing popularity of HD video, and the emergence of beyond- HD formats (e.g., 4k×2k or 8k×4k resolution) [10] are creating even stronger needs for coding efficiency superior to H.264/MPEG-4 AVC’s capabilities. The need is even stronger when higher resolution is accompanied by stereo or multiview capture and display. Moreover, the traffic caused by video applications targeting mobile devices and tablets PCs, as well as the transmission needs for video-on-demand services, are imposing severe challenges on today’s networks. An increased desire for higher quality and resolutions is also arising in mobile applications [1].

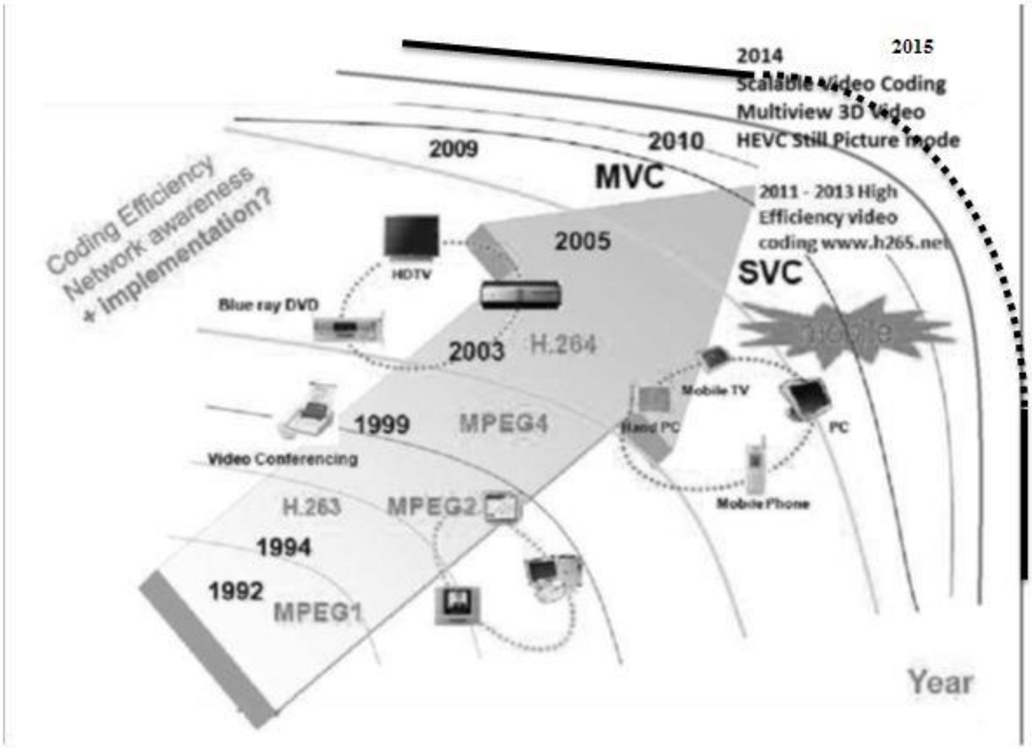


Figure 1-1 Evolution of video coding standards [11]

1.2 Why Is Complexity Reduction Important In HEVC/H.265 ?

HEVC/H.265 has very efficient compression methods, which allow it to compress video much more efficiently than older standards and provide more flexibility for application to a wide variety of network enviornments. To achieve highly efficient compression, the computational cost associated with it is also very high. This is the reason why, these increased compression efficiencies cannot be exploited across all application domains. Resource constrained devices such as cell phones and other embedded systems use simple encoders or simpler profiles of the codec to tradeoff compression efficieny and quality for reduced complexity [3]. Video coding standards specify the decoding process and bitstream syntax of the compressed video. The encoding process or the process of producing a standard compliant video is not specified. This approach leaves room for innovation in the encoding algorithm development. The work in this thesis focuses on coding unit early termination algorithm and fast intra mode decision to decrease the encoder complexity for the intra prediction modes of HEVC.

1.3 Outline Of The Research

The research presented here proposes a reduced complexity HEVC encoder by making use of HM 13.0 reference software [4]. A new technique is implemented for reducing encoding complexity in HEVC. The results show reduction in complexity in terms of encoding time for different videos sequences, with acceptable loss in the PSNR and bit-rates.

1.4 Thesis Outline

Chapter 2 provides details of various blocks in HEVC encoder along with brief explanation of encoding process. Chapter 3 discusses present intra-prediction technique along with various encoder complexity reduction algorithms present for coding unit and prediction unit blocks along with proposed implemetation method for reducing complexity using coding unit early termination with fast intra mode decision for HEVC. Chapter 4 discusses the simulations and the results for different formats of test sequences. Chapter 5 outlines the conclusions and further research. The configuration files used by the HM 13.0 [4] software of HEVC encoder for the generation of the bitstreams are also provided.

Chapter 2

High Efficiency video coding

HEVC is the latest video standard introduced by the Joint Collaborative Team on Video Coding (JCT-VC) in January, 2013 which contains three profiles namely; main (8-bit), main10 (10-bit) and still frame. Here only the main (8-bit) profile is considered since it is most widely used profile. The HEVC standard is designed to achieve multiple goals, including coding efficiency, ease of transport system integration, data loss resilience and implementation using parallel processing architectures. The HEVC standard has been designed to address essentially, all the existing applications of the H.264/MPEG-4 AVC standard [1] and to particularly focus on two key issues: increased video resolution and increased use of parallel processing architectures [1]. The major achievements of the HEVC standard in comparison with the H.264 [1] standard are flexible prediction modes, larger transform block sizes, better partitioning options, improved interpolation and deblocking filters, prediction, signaling of modes and motion vectors and support efficient parallel processing [1]. HEVC has been designed to address essentially all existing applications of H.264/MPEG-4 AVC and to particularly focus on two key issues: increased video resolution and increased use of parallel processing architectures. The HEVC syntax should be generally suited for other applications and not specifically to two applications mentioned above [1]. This is not the result of optimizing a single step in the encoding process, but a combined result of optimization of many processes together.

The HEVC extension [12] also includes extended-range formats with increased bit depth and enhanced color component sampling, scalable coding, and 3-D/stereo/multi-view video coding (the latter including the encoding of depth maps for use with advanced 3-D displays) [1]. As more and more emphasis is laid on video streaming and playback of HD and beyond HD quality, the HEVC standard is a great improvement with respect to the previous standards. The basic design of the HEVC standard remained the same as that of the H.264/AVC i.e., the block based hybrid coding approach which efficiently exploits the temporal statistical dependencies and the spatial statistical dependencies [1].

The block diagrams of the HEVC encoder and decoder are shown in figures 2-1 and 2-2 respectively.

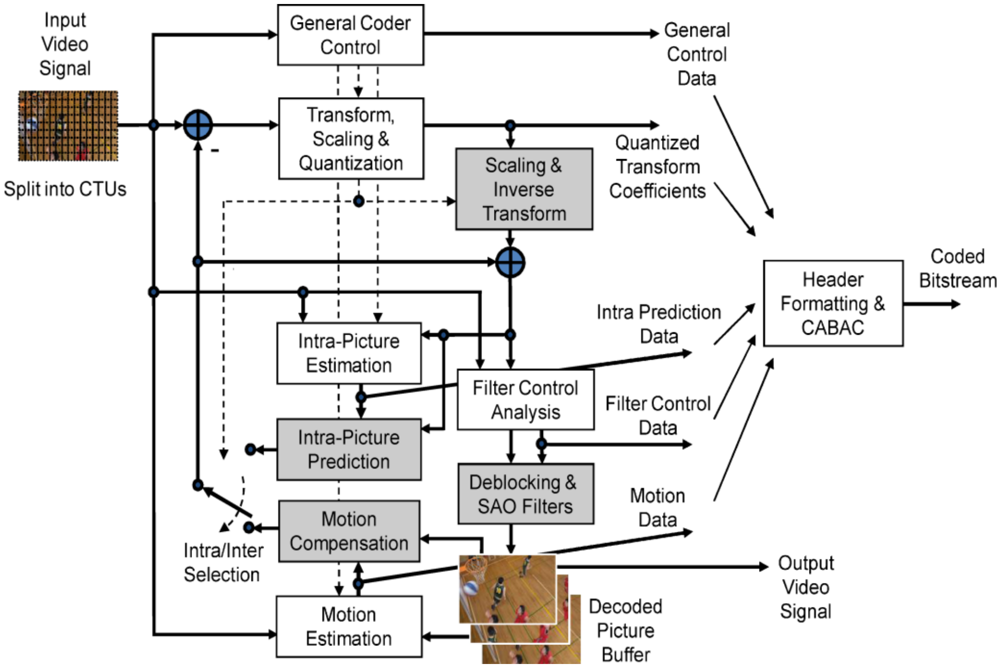


Figure 2-1 HEVC encoder block diagram [1]

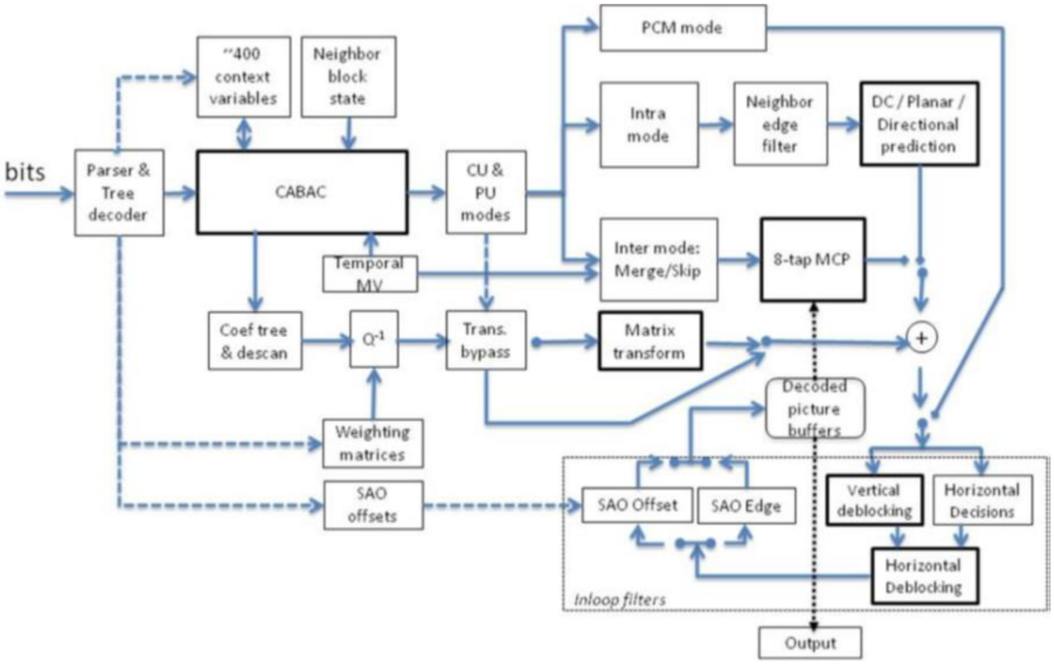


Figure 2-2 HEVC decoder block diagram [13]

2.1 HEVC Coding Design And Feature Highlights

The video coding layer of the HEVC standard employs the same hybrid approach used in all video compression standards since H.261 [1]. The HEVC standard is designed to achieve multiple goals, including coding efficiency, ease of transport system integration and data loss resilience, as well as implementability using parallel processing architectures [1].

The HEVC standard employs adaptive and flexible quad-tree coding block partitioning structure which enables efficient use of large multiple sizes of prediction, coding, transform block employs improved intra prediction, adaptive motion parameter prediction, new loop filter and an enhanced context-adaptive binary arithmetic coding (CABAC) as entropy coding method [16].

2.1.1 Video Coding Layer And Structure Of Encoder

The HEVC standard is a block-based hybrid-coding scheme. One of the major contributors to its higher compression performance is the introduction of larger block structures with flexible subpartitioning mechanisms. The basic block in the standard HEVC is known as the largest coding unit (LCU) and can be recursively split into smaller coding units (CUs), which in turn can be split into small prediction units (PUs) and transform units (TU) [14]. Figure 2-3 shows an example of partitioning a 64 X 64 LCU to various sizes of CUs.

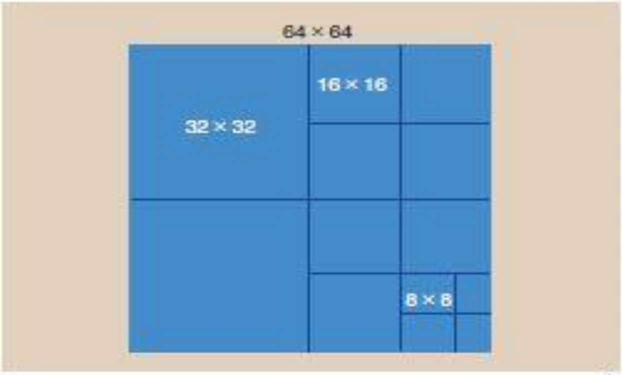


Figure 2-3 Partitioning of a 64 X 64 LCU to various sizes of CU [14]

The quad-tree block partitioning is based on a coding tree unit (CTU) structure as shown in figure 2-5 which is analogous to the macro block in previous standards. Video is a packet or sequence of frames and in the HEVC standard each coded video frame is partitioned into tiles, slices and CTUs. CTUs are subdivided into square regions called coding units (CU). CUs are predicted using intra or inter prediction where the first frame at each random access point of a video sequence is coded using only intra prediction so that it has no dependence on other pictures. The remaining frames are mostly coded by inter prediction, then residual is transformed using transform units and encoded using CABAC [22] [21]

[22]. The HEVC code uses YCbCr color space with a 4:2:0 color format with 8 bps (bits per color sample). Y is symbol for luma component, Cb is symbol for the blue chroma component and Cr is symbol for the red chroma component [20] as shown in figure 2-4 [20].

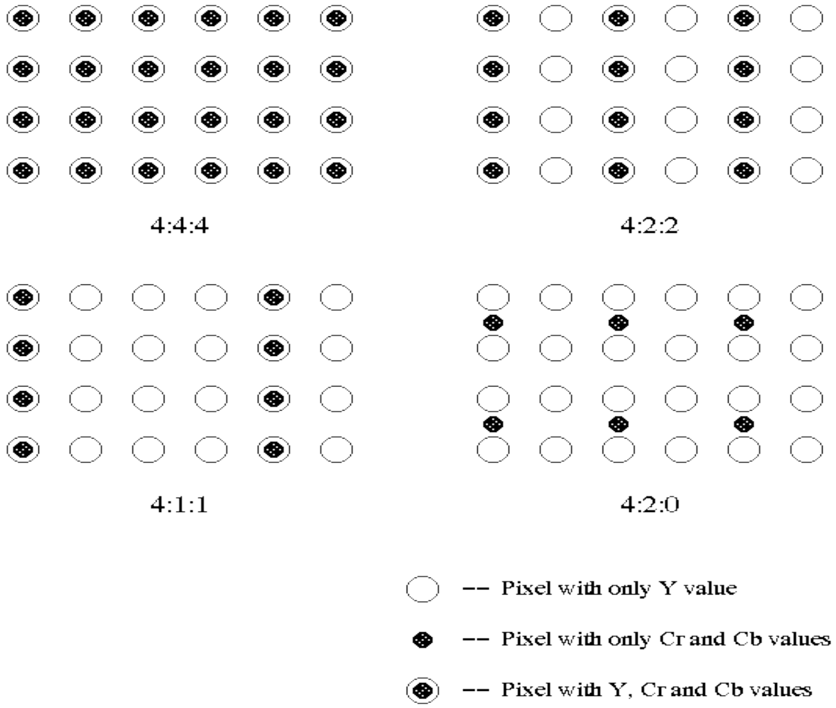


Figure 2-4 Formats for YUV components [16]

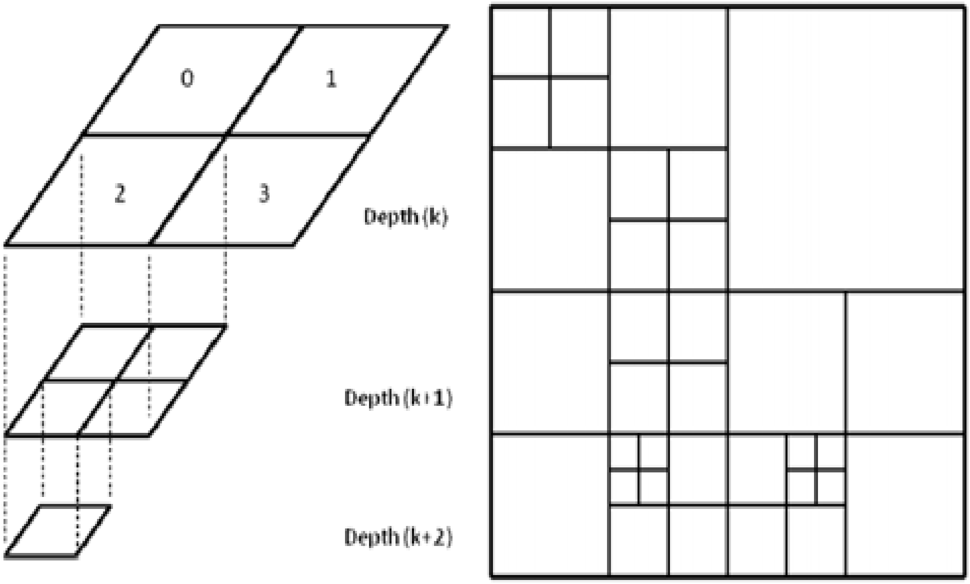


Figure 2-5 Quad tree CU structure in HEVC [20] [1]

As the picture resolution of videos increases from standard definition to HD and beyond, the chances are that the picture will contain larger smooth regions, which can be encoded more effectively when large block sizes. This is the reason that the HEVC standard supports larger encoding blocks than H.264/AVC, while it also has a more flexible partitioning structure to allow smaller blocks to be used for more textured and in general uneven regions [14].

Each CU can be further split into smaller units, which form the basis for prediction. These units are called PUs. Each CU may contain one or more PUs, and each PU can be as large as their root CU or as small as 4x4 in luma block sizes. While an LCU can recursively split into smaller and smaller CUs, the splitting of a CU into PUs is nonrecursive. PUs can be symmetric or asymmetric. Symmetric PUs can be square or rectangular and are used in both intraprediction and interprediction. In particular, a CU of size 2Nx2N can be split into two symmetric PUs of size Nx2N or 2NxN or four PUs of size NxN. Asymmetric PUs are used only for interprediction. This allows partitioning, which matches the boundaries of the objects in the picture [14]. Figure 2-6 shows the intra and inter frame prediction modes for the HEVC standard.

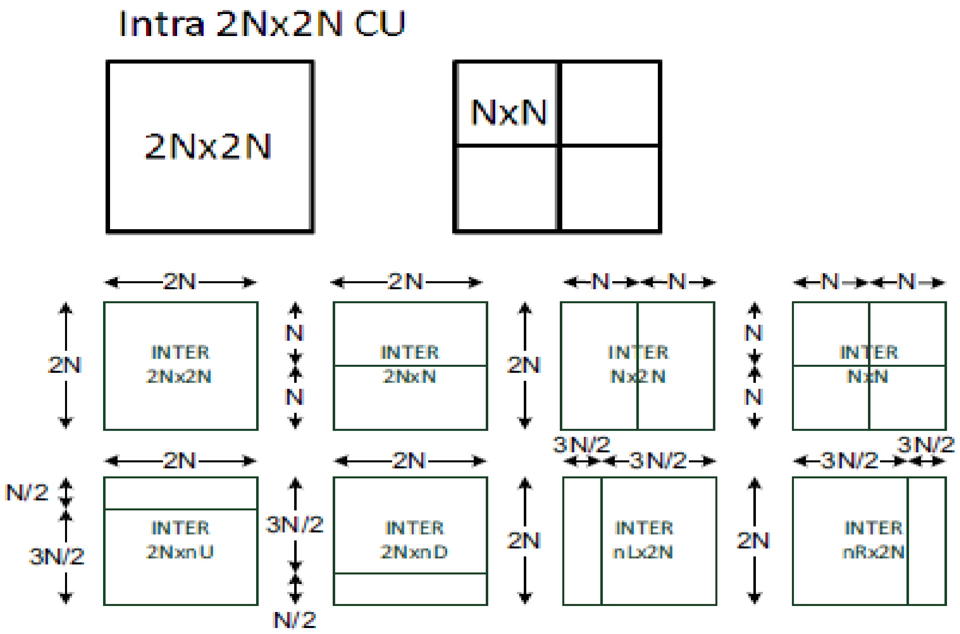


Figure 2-6 Intra and Inter frame prediction modes for HEVC [20]

A transform unit (TU) is the basic unit for the transform and quantization processes. The size and the shape of the TU depend on the size of the PU. Figure 2-8 shows the partitioning of a 32x32 CU into PUs and TUs. The size of square-shape TUs can be as small as 4x4 or as large as 32x32 and nonsquare TUs can have sizes of 32x8, 8x32, 16x4, or 4x16 luma samples as shown in figure 2-7. Each CU may contain one or more TUs, each square CU may split into smaller TUs in a quad-tree segmentation structure [14].

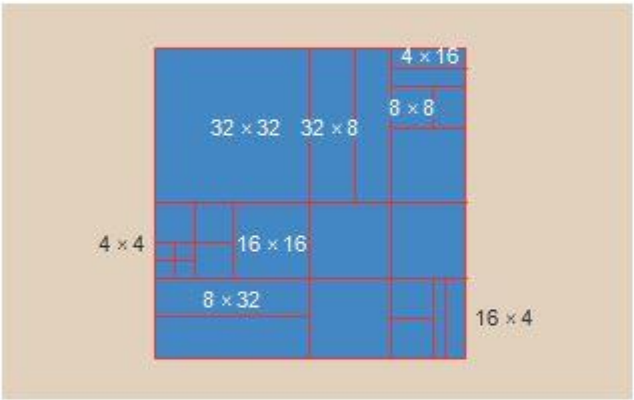


Figure 2-7 Arrangement of TUs in a CU [14]

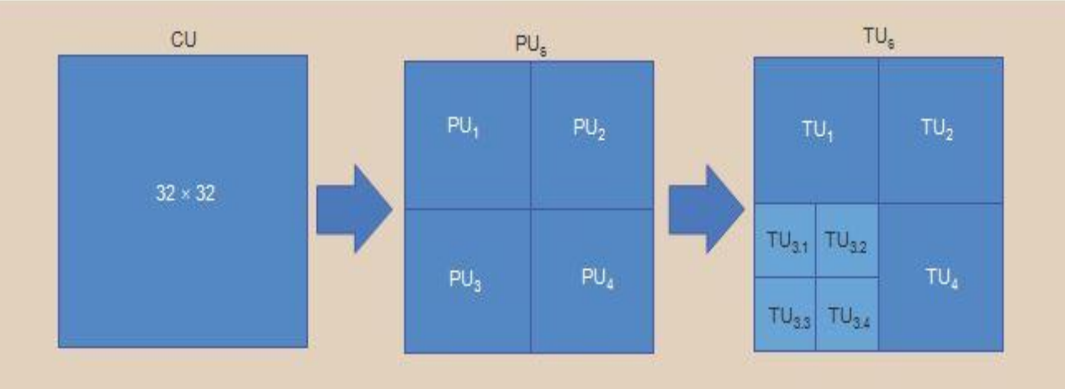


Figure 2-8 Partitioning of 32x32 CU into PUs and TUs [14]

Similarly, starting at the level of a CU, a CB (Coding Block) can have one Transform Block (TB) of the same size as the CB or be split into smaller TBs [1] [21] [22] as shown in figures 2-9, 2-10 and 2-11.

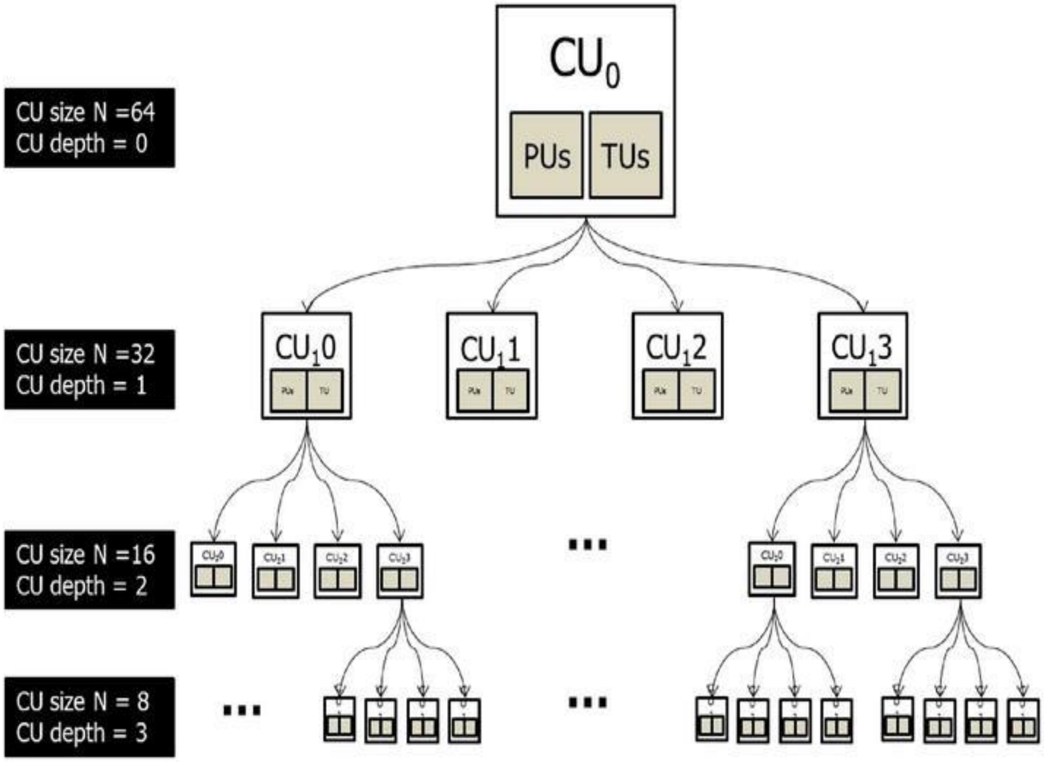


Figure 2-9 Splitting Coding unit into prediction units and transform units [23]

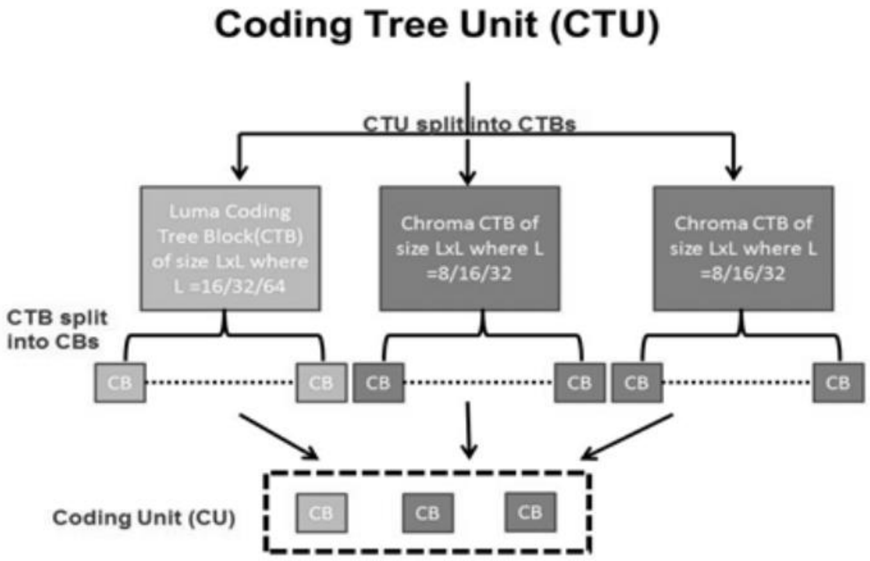


Figure 2-10 Splitting Coding tree unit into Coding Blocks [1]

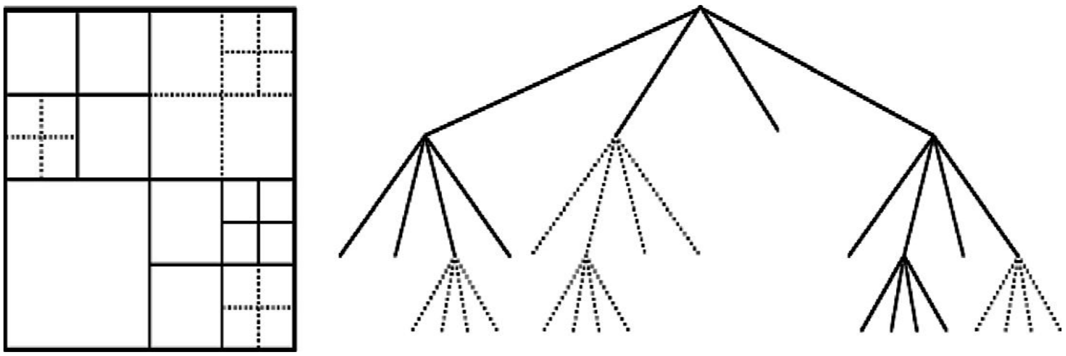


Figure 2-11 CTB with its partitioning and corresponding quad tree [1]

2.1.2 Slice And Tiles

The HEVC standard introduced tiles as a means to support parallel processing, with more flexibility than the normal slices in the H.264/AVC standard [2] but considerably lower complexity than the Flexible Macroblock Ordering (FMO) standard. Tiles are specified by vertical and horizontal boundaries with intersections that partition a picture into rectangular regions. Figure 2-12 shows an example of tile partitions that contain slices. The spacing of the row and column boundaries of tiles need not be uniform. This offers greater flexibility and can be useful for error resilience applications. In each tile, LCUs are processed in a raster scan order. Similarly, the tiles themselves are processed in a raster scan order within a picture.

The HEVC standard also supports slices, similar to slices found in the H.264/AVC standard, but without FMO. Slices and tiles may be used together within the same picture. To support parallel processing, each slice in HEVC can be subdivided into smaller slices called entropy slices. Each entropy slice can be independently entropy decoded without reference to other entropy slices. Therefore, each core of a CPU can handle an entropy-decoding process in parallel [14]. Figure 2-12 shows the tile partitions containing slices.

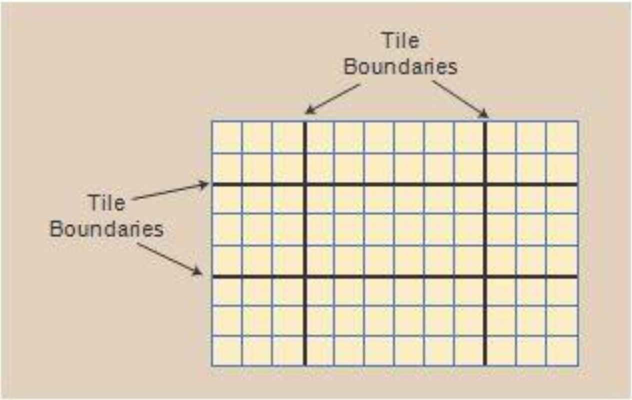


Figure 2-12 A picture partitioned into nine tiles [14]

The slices are processed in the order of a raster scan. A picture may be split into one or several slices as shown in figure 2-13 so that a picture is a collection of one or more slices. Slices are self-contained in the sense that, given the availability of the active sequence and picture parameter sets, their syntax elements can be parsed from the bit stream and the values of the samples in the area of the picture that the slice represents can be correctly decoded without the use of any data from other slices in the same picture.

Tiles are self-contained and independently decodable rectangular regions of the picture. The main purpose of tiles is to enable the use of parallel processing architectures for encoding and decoding. Multiple tiles may share header information by being contained in the same slice. Alternatively, a single tile may contain multiple slices. A tile consists of a rectangular arranged group of CTUs as shown in figure 2-13.

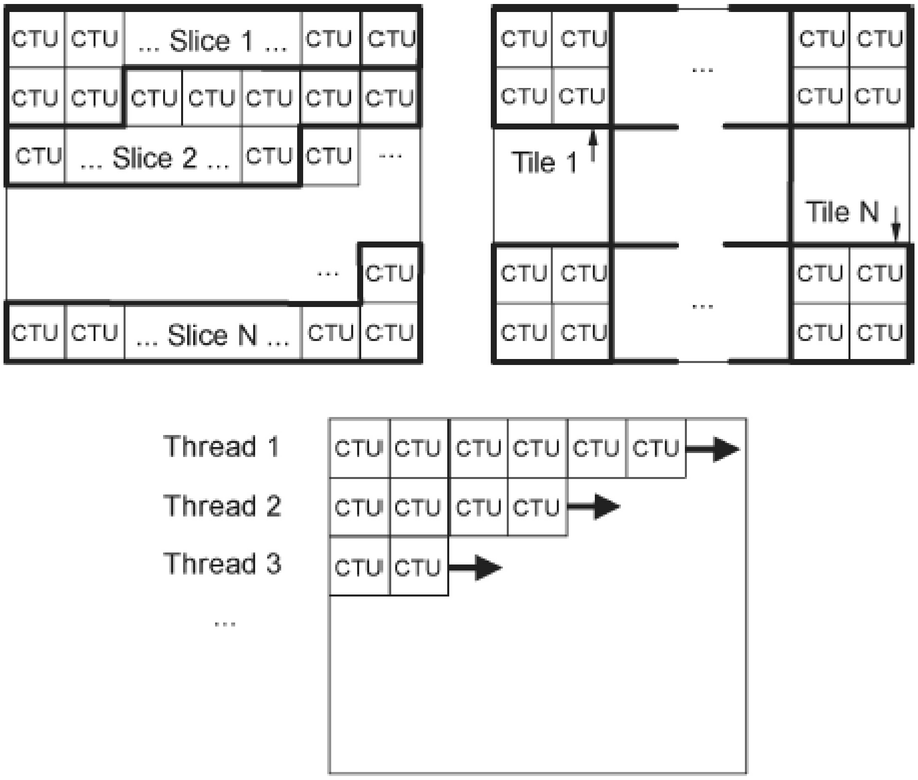


Figure 2-13 Subdivision of picture into slice and Tiles [1]

2.2 HEVC Encoder Description

There are five major parts of the HEVC encoder which are discussed in the following section.

2.2.1 Intra-Picture Prediction

Intra-picture prediction operates according to the TB size and previously decoded boundary samples from spatially neighboring TBs which are used to form the prediction signal. Directional prediction with 33 different directional orientations is defined for (square) TB sizes from 4×4 up to 32×32. The possible prediction directions are shown in figure 2-14. Alternatively, planar prediction and DC prediction can also be used. For chroma the horizontal, vertical, planar, and DC prediction modes can be explicitly signaled, or the chroma prediction mode can be indicated to be the same as the luma prediction mode [1].

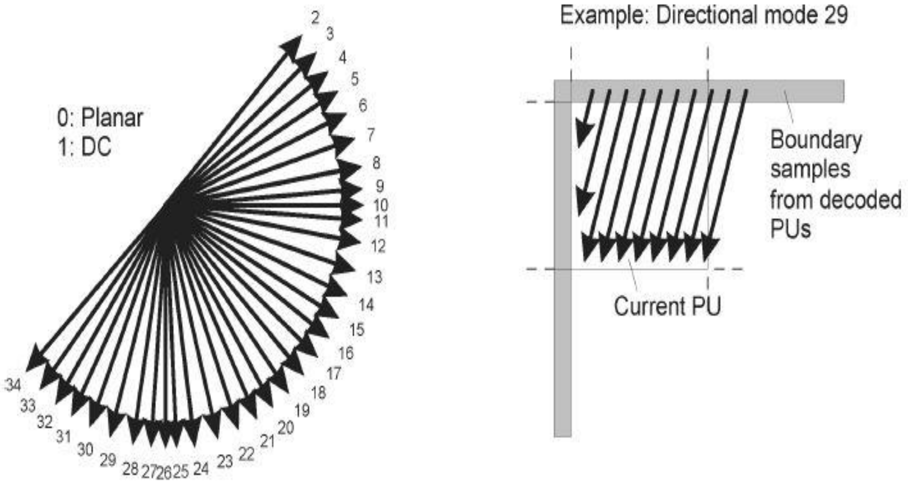


Figure 2-14 Mode decision for intra picture prediction [1]

The HEVC standard also includes a planar intra-prediction mode which is useful for predicting smooth picture regions. In planar mode, the prediction is generated from the average of two linear interpolations.

2.2.2 Inter-Picture Prediction

Compared to intra-picture predicted CBs, the HEVC standard supports more PB partition shapes for inter-picture predicted CBs. The partitioning modes of PART_2N×2N, PART_2N×N and PART_N×2N as shown in Figure 2-15 indicate the cases when the CB is not split, split into two equal-size PBs horizontally, and split into two equal-size PBs vertically, respectively. PART−N×N specifies that the CB is split into four equal size PBs, but this mode is only supported when the CB size is equal to the smallest allowed CB size. In addition, there are four partitioning types that support splitting the CB into two PBs having different sizes: PART−2N×nU, PART−2N×nD, PART−nL×2N, and PART−nR×2N (U=up, D=down, L=left and R=right) as shown in figure 2-5. These types are known as asymmetric motion partitions [1].

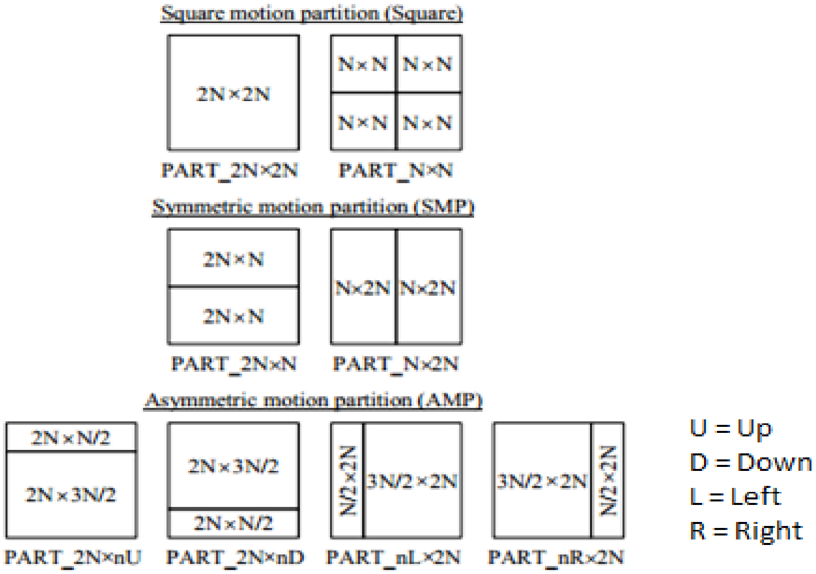


Figure 2-15 Partition modes in HEVC inter-prediction [22]

2.2.3 Transform, Scaling And Quantization

The HEVC standard uses transform coding of the prediction error residual in a similar manner as in prior standards. The residual block is partitioned into multiple square TBs. The supported transform block sizes are 4×4, 8×8, 16×16, and 32×32 [1]. Pre-scaling operation is not needed when using HEVC code since the rows of the transform matrix are a close approximations of values of uniformly scaled basis functions of the orthonormal DCT (Discrete Cosine Transform) [1]. Uniform reconstruction quantization (URQ) is used in the HEVC standard, with quantization scaling matrices supported for the various transform block sizes [1]. The range of the QP values is defined from 0 to 51, and an increase by 6 doubles the quantization step size such that the mapping of QP values to step sizes is approximately logarithmic.

2.2.4 Entropy Coding

A new and improved CABAC (Context Adaptive Binary Arithmetic Coding) is used for the entropy coding of the bitstreams. This coding has improved speed, compression and requires less memory then entropy coding used in the H.264/AVC standard (figure 2-16). Instead of doing the normal CABAC re-initialization for every CTB row, the context state from the second CTU in the previous row is used to start the processing of a brand new CTB row (figure 2-17), and thus taking huge advantage of parallel processing.

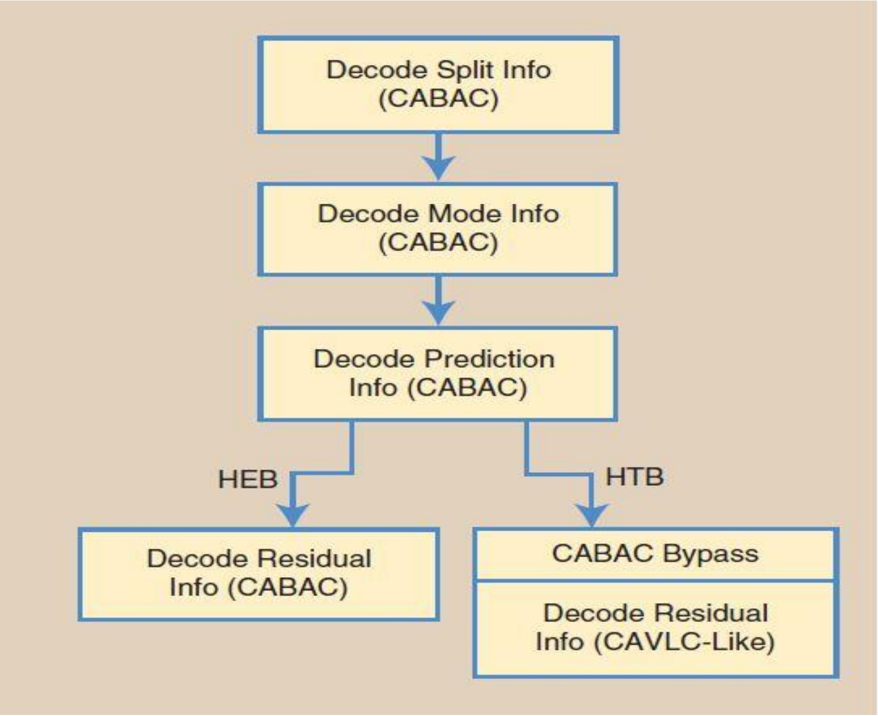


Figure 2-16 HEVC entropy coding [14]

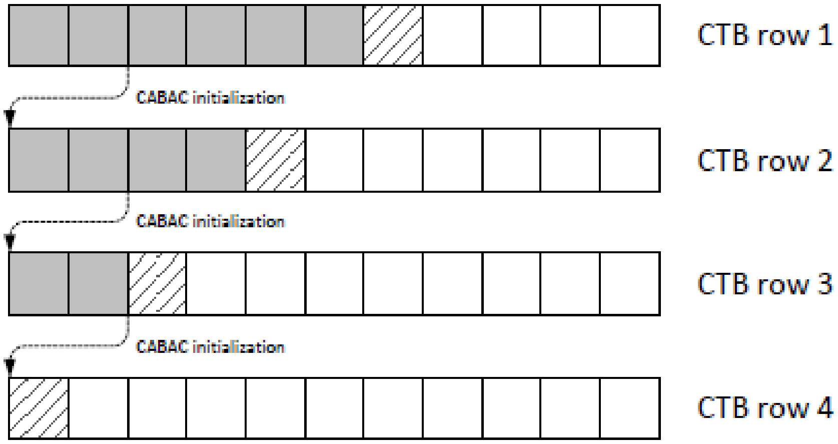


Figure 2-17 Example of waveform processing [1]

2.2.5 In-Loop Filtering

In the HEVC standard, two processing steps, namely a deblocking filter (DBF) followed by a sample adaptive offset (SAO) filter are applied to the reconstructed samples before writing them into the decoded picture buffer in the decoder loop. The DBF is intended to reduce the blocking artifacts due to block-based coding. The deblocking filter is applied to all samples adjacent to a PU or TU boundary except the case when the boundary is also a picture boundary, or when deblocking is disabled across slice or tile boundaries. It should be noted that both PU and TU boundaries should be considered since PU boundaries are not always aligned with TU boundaries in some cases of interpicture-predicted CBs. Syntax elements in the SPS and slice headers control whether the deblocking filter is applied across the slice and tile boundaries [1]. The SAO is a process that modifies the decoded samples by conditionally adding an offset value to each sample after the application of the deblocking filter, this is based on values in look-up tables transmitted by the encoder [1].

2.3 Summary

This chapter outlines the coding tools of the HEVC codec. The intent of the HEVC is to create a standard capable of providing good video quality at substantially lower bit rates than previous standards. Chapter 3 outlines the description of intra-prediction mode decision and the proposed early termination CU with fast intra mode decision algorithm.

Chapter 3

Intra-Prediction And Fast Intra Mode Decision

3.1 Intra Prediction Introduction

In order to generate prediction for a current block, the encoder will have information about decoded pixels in the row above this block and a column to the left of this block. Using this information, the encoder can predict the value of the current block and subsequently quantize and transform the residual for transmission. This is the basic idea of intra prediction. The word “intra” indicates that the considered frame uses only pixels within itself for the prediction process.

3.2 Intra Prediction In Detail

Intra picture prediction operates according to the TB size, and previously decoded boundary samples from spatially neighboring TBs are used to form the prediction signal. Directional prediction with 33 different directional orientations is defined for TB sizes from 4×4 up to 32×32. Alternatively, planar prediction and DC can also be used. For chroma, the horizontal, vertical, planar, and DC prediction modes can be explicitly signaled, or the chroma prediction mode can be indicated to be the same as the luma prediction mode. Each CB can be coded by one of several coding types, depending on the slice type. Similar to H.264/MPEG-4 AVC [2], intra picture predictive coding is supported in all slice types. HEVC supports various intra picture predictive coding methods referred to as Intra−Angular, Intra−Planar, and Intra−DC. Figure 3-1 shows the luma intra prediction modes of HEVC [1].

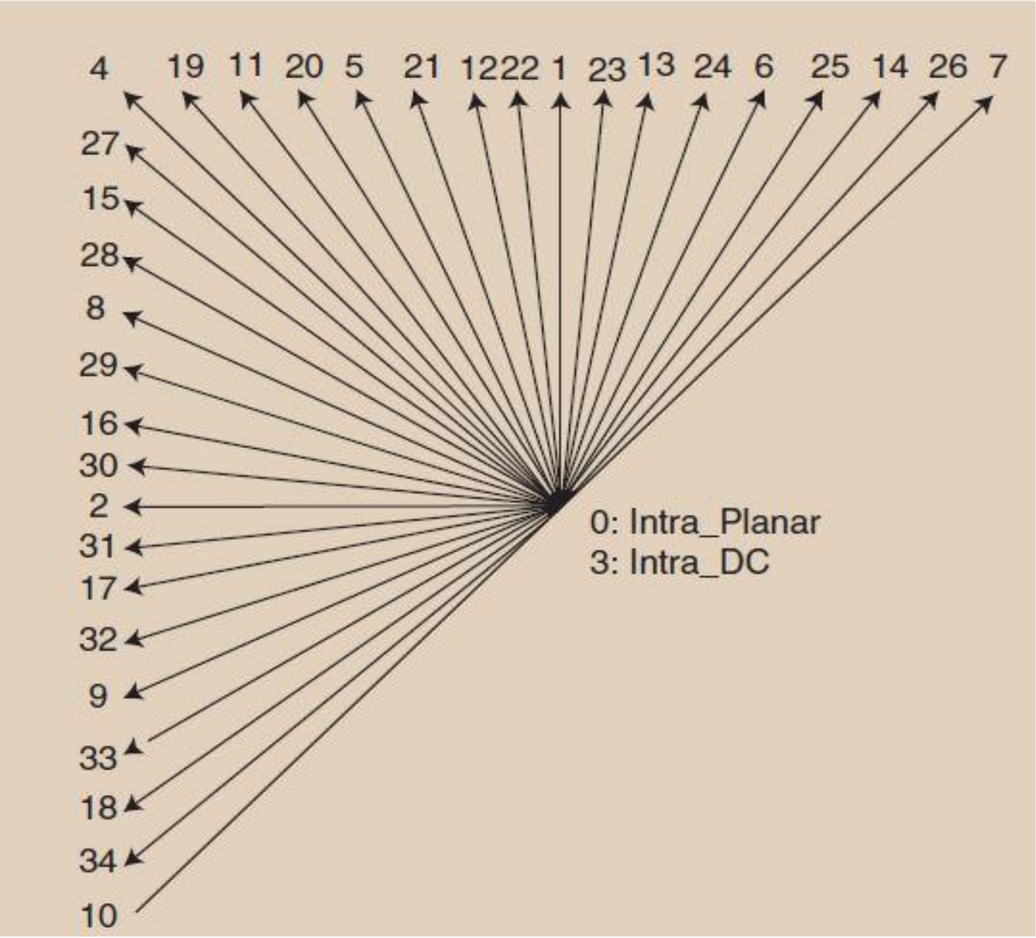


Figure 3-1 Luma intra prediction modes of HEVC [14]

3.3 PB Partitioning

An intra-picture predicted CB of size M×M may have one of two types of PB partitions which are referred to as PART−2N×2N and PART−N×N, the first indicates that the CB is not split and the second indicates that the CB is split into four equal-sized PBs. However, it is possible to represent the same regions that would be specified by four PBs by using four smaller CBs when the size of the current CB is larger than the minimum CU size. The HEVC design only allows the partitioning type PART−N×N to be used when the current CB size is equal to the minimum CU size. This means that the PB size is always equal to the CB size when the CB is coded using an intrapicture prediction mode and the CB size is not equal to the minimum CU size. Although the intrapicture prediction mode is established at the PB level, the actual prediction process operates separately for each TB [1]. Table 3-1 shows the luma intraprediction modes supported by different PU sizes.

3.4 Intra Angular Prediction

The intrapicture prediction of the HEVC standard similarly operates in the spatial domain, but is extended significantly mainly due to the increased size of the TB and an increased number of selectable prediction directions. The HEVC standard supports a total of 33 prediction directions denoted as Intra−Angular[k] where ‘k’ is a mode number from 2 to 34. The angles are intentionally designed to provide denser coverage for near-horizontal and near-vertical angles and coarser coverage for near-diagonal angles to reflect the observed statistical prevalence of the angles and the effectiveness of the signal prediction processing. When using an Intra Angular mode, each TB is predicted directionally from spatially neighboring samples that are reconstructed before being used for this prediction. For a TB of size N×N, a total of 4N+1 spatially neighboring samples may be used for the prediction, as shown in figure 3-1. When available from preceding decoding operations, samples from lower left TBs can be used for prediction in HEVC in addition to samples from TBs at the left, above, and above right of the current TB [1].

The prediction process of the Intra Angular modes can involve extrapolating samples from the projected reference sample location according to a given directionality. To remove the need for sample-by-sample switching between reference row and column buffers, for Intra−Angular[k] with k in the range of 2–17, the samples located in the above row are projected as additional samples located in the left column and with k in the range of 18–34, the samples located at the left column are projected as samples located in the above row. To improve the intrapicture prediction accuracy, the projected reference sample location is computed with 1/32 sample accuracy. Bilinear interpolation is used to obtain the value of the projected reference sample using two closest reference samples located at integer positions [1].

3.5 Intra−Planar and Intra−DC Prediction

In addition to Intra−Angular prediction that targets regions with strong directional edges, HEVC supports two alternative prediction methods, Intra−Planar and Intra−DC. Intra−DC prediction uses an average value of reference samples for the predictions average values of two linear predictions using four corner reference samples are used in Intra−Planar prediction to prevent discontinuities along the block boundaries [1].

3.6 Reference Sample Smoothing

In HEVC, the reference samples used for the intrapicture prediction are sometimes filtered by a three-tap [1 2 1]/4 smoothing filter which applies smoothing operations more adaptively, according to the directionality, the amount of detected discontinuity, and the block size Intra−Angular[k] with k = 2, 18, or 34, use the reference sample smoothing. For 16×16 blocks, the reference samples are filtered for most of the directions except the near-horizontal and near-vertical directions, k in the range of 9–11 and 25–27. For 32×32 blocks, all directions except the exactly horizontal (k = 10) and exactly vertical (k = 26) directions use the smoothing filter, and when the amount of detected discontinuity exceeds a threshold, bilinear interpolation from three neighboring region samples is applied to form a smooth prediction. The Intra−Planar mode also uses the smoothing filter when the block size is greater than or equal to 8×8, and the smoothing is not used for the Intra−DC case [1].

3.7 Boundary Value Smoothing

To remove discontinuities along block boundaries, in three modes, Intra−DC (mode 1) and Intra−Angular[*k*] with *k* = 10 or 26 (exactly horizontal or exactly vertical), the boundary samples inside the TB are replaced by filtered values, when the TB size is smaller than 32 × 32. For the Intra−DC mode, both the first row and column of samples in the TB are replaced by the output of a two-tap [3 1]/4 filter fed by their original value and the adjacent reference sample. In horizontal prediction, the boundary samples of the first column of the TB are modified such that half of the difference between their neighbored reference sample and the top-left reference sample is added. This makes the prediction signal smoother when large variations in the vertical direction are present. In vertical prediction, the same is applied to the first row of samples [1].

3.8 Reference Sample Substitution

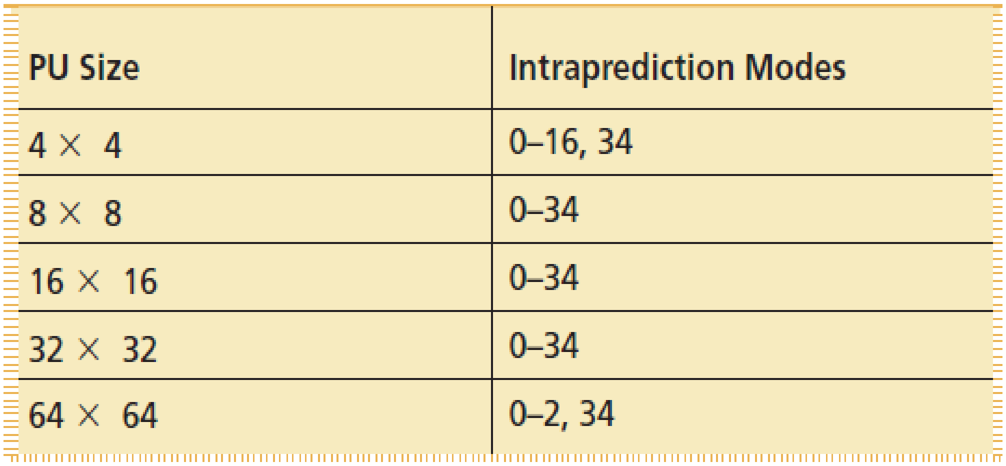
The neighboring reference samples are not available at the slice or tile boundaries. In addition, when a loss-resilience feature known as constrained intra prediction is enabled, the neighboring reference samples inside any inter picture-predicted PB are also considered not available in order to avoid letting potentially corrupted prior decoded picture data propagate errors into the prediction signal [1]. HEVC allows the use of other intra picture prediction modes after substituting the non-available reference sample values with the neighboring available reference sample values.

3.9 Mode Coding

HEVC supports a total of 33 Intra−Angular prediction modes (Fig. 3-1) and Intra−Planar and Intra−DC prediction modes for luma prediction for all block sizes. Due to the increased number of directions, HEVC considers three most probable modes (MPMs) when coding the luma intra picture prediction mode.

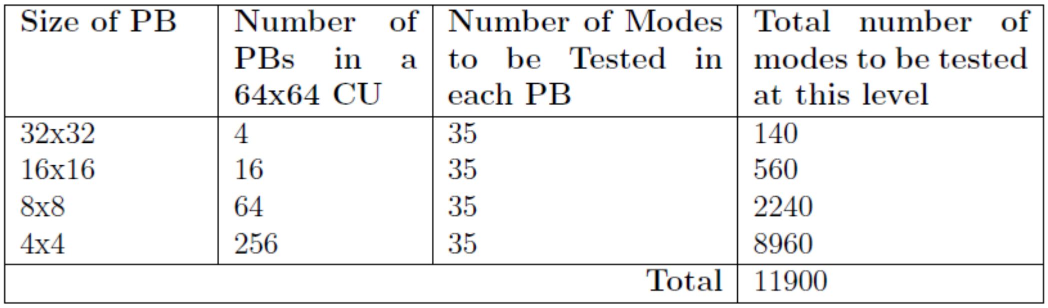
Among the three most probable modes, the first two are initialized by the luma intra picture prediction modes of the above and left PBs if those PBs are available and are coded using an intrapicture prediction mode. Any unavailable prediction mode is considered to be intra−DC. The PB above the luma CTB is always considered to be unavailable in order to avoid the need to store a line buffer of neighboring luma prediction modes. When the first two most probable modes are not equal, the third most probable mode is set equal to Intra−Planar, Intra−DC, or Intra−Angular, according to which of these modes, in this order, is not a duplicate of one of the first two modes. When the first two most probable modes are the same, if this first mode has the value Intra−Planar or Intra−DC, the second and third most probable modes are assigned as Intra−Planar, Intra−DC, or Intra−Angular, according to which of these modes, in this order, are not duplicates [1]. When the first two most probable modes are the same and the first mode has an Intra−Angular value, the second and third most probable modes are chosen as the two angular prediction modes that are closest to the angle (i.e., the value of k) of the first. In the case that the current luma prediction mode is one of three MPMs, only the MPM index is transmitted to the decoder. Otherwise, the index of the current luma prediction mode excluding the three MPMs is transmitted to the decoder by using a 5-b fixed length code. For chroma intra picture prediction, HEVC allows the encoder to select one of five modes: Intra−Planar, Intra−Angular, Intra−Angular, Intra−DC, and Intra−Derived. The intra derived mode specifies that the chroma prediction uses the same angular direction as the luma prediction. With this scheme, all angular modes specified for luma in HEVC can, in principle, also be used in the chroma prediction, and a good tradeoff is achieved between prediction accuracy and the signaling overhead. The selected chroma prediction mode is coded directly [1]. Table 3-2 shows the HEVC encoder complexity for CU and PB blocks.

Table 3-1 Luma intraprediction modes supported by different PU sizes [14]



PU Size	Intraprediction Modes
4 × 4	0–16, 34
8 × 8	0–34
16 × 16	0–34
32 × 32	0–34
64 × 64	0–2, 34

Table 3-2 Current Problem-Complexity for HEVC [33]



Size of PB	Number of PBs in a 64x64 CU	Number of Modes to be Tested in each PB	Total number of modes to be tested at this level
32x32	4	35	140
16x16	16	35	560
8x8	64	35	2240
4x4	256	35	8960
		Total	11900

3.10 Proposed Solution - Fast Intra Coding

A large number of researchers have proposed various techniques for making the intra prediction process faster. [20–28]

A three step method is proposed as a solution. In CU splitting, decision is made whether to split the current CU further by analysing the CU texture characteristics. In PU partition, down sampling prediction followed by three – step search is exploited [24]. In the last step the early RDOQ termination is implemented [25].

3.10.1 CU Early Termination

When the CU texture is complex the CU is split into smaller sub units to find the best size and when the CU texture is flat, the CU is not divided further into sub – units. This has already been proved [12].

In the first stage, to decrease the computational complexity, the down-sampling method is exploited by applying a 2:1 down sampling filter by a simple average operator to the current CU and other CU have the similar operation as shown in figure 3-2.

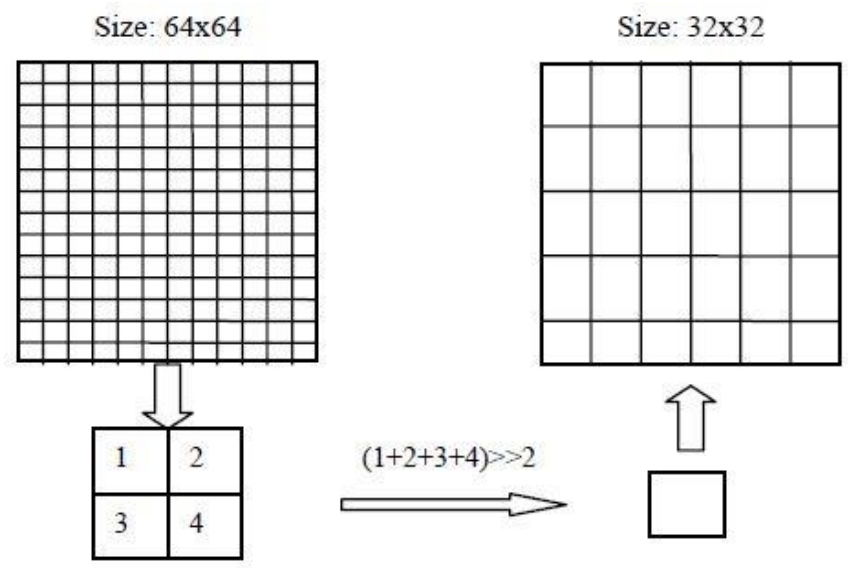


Figure 3-2 Simple averaging based on down-sampling on 64x64 CU [27]

After the down sampling, the complexity of the original LCU can be calculated by the following formula,

$$E_{com} = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left[p(i,j) - \frac{1}{N}\frac{1}{N}\left(\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}p(i,j)\right)\right]$$

where E_{com} represents the texture complexity, N is the size of the current CU, p(i, j) is the pixel and (i, j) is the coordinate in CU.

Depending on the texture calculation, two thresholds are set with a tradeoff on coding quality and complexity reduction as Thres1 and Thres2. The CU is split when the complexity is greater than Thres1 and when complexity is less than Thres2, the CU is not split further. If the complexity is between the Thres1 and Thres2, HEVC reference software is referred [4].

3.10.2 PU Mode Decision

In the second stage, PU modes decision is obtained by calculating the Sum of Absolute Differences (SAD) which is performed by down sampling and then by applying similar three step search algorithm. The detailed operation is as follows.

1) List of candidates are created, S1={0,1,2,6,8,12,16,20,24,28,30,32,34} from the 35 prediction modes (figure 3-1) and then 5 optimal modes by SAD is check on S1, suppose 5 modes are S2={0,3,12,16,34}.
2) From the three-step algorithm [27], list S2 is extended on to the 2-distance neighbors and S3={2,10,20,32} for both the modes 0 and 2 and then S1, S2, S3 are checked for optimal modes S4={8,14,24}. Suppose modes of upper and left PUs are S5={1,6}, then checking optimal modes and if the optimal two modes are S6={2,6}.
3) Then 1-distance neighbors of S6 are S7={3,5,9} then we choose the best M modes as the candidates for RDOQ.

3.10.3 Early RDOQ Termination

In the third stage, there are M modes selected from the result of the second step which are put into a group, Ψ, that go through the RDOQ process to get the best mode, mopt. An early RDOQ termination is proposed for further encoder time reduction. For each intra mode $m \in \Psi$, its overall cost J(m) as the combination of SATD cost and associated mode index bits consumption is calculated. Within Ψ there is a mode with minimal J_{min} defined as rough best mode mopt_rough. If mopt_rough is Planar or DC mode, all other modes in Ψ are skipped. If mopt_rough != 0 or 1 , and |m−mopt_rough| > 3, such mode m is skipped also; Meanwhile, if $J(m) > \alpha J_{min}$, mode m will not be checked and $\alpha = 1.08$ is considered. After such early termination procedure, all the remaining modes are checked by RDOQ.

3.11 Summary

This chapter introduced, different angular prediction modes and CU splitting and the PU predicition modes. CU early temination with the fast intra prediciton algorithm is explained. Chapter 4 outlines simulations and results for different resolution test sequences (Appendix A) for unmodified and modified HM version code base [4].

Chapter 4

Results

4.1 Test Conditions

In order to evaluate the performance of the proposed intra prediction algorithm, the algorithm is implemented on the recent HEVC reference software (HM 13.0) [4]. The intra main profile is used for coding with the intra period set as 1 and frame rate set at 30 fps. The proposed algorithm is evaluated with 4 QPs of 22, 27, 32 and 37 using the following test sequences recommended by JCT-VC [35]. A frame of each test sequence is shown in Appendix A.

Table 4-1 Test Sequences Used

No.	Sequence Name	Resolution	Type	No. of frames
1.	RaceHorses	416x240	WQVGA	30
2.	BasketballDrillText	832x480	WVGA	30
3.	SlideEditing	1280x720	SD	30
4.	Kimono	1920x1080	HD	30
5.	PeopleOnStreet	2560x1600	WQHD	30

4.2 Encoder Complexity Reduction

With the proposed CU early termination algorithm, encoder complexity in terms of encoding time for the test sequences is reduced by 35-48% as compared to the unmodified encoding HM13.0 [4]. The following test results (figures 4-1 to 4-5) show the difference in encoding time of the original HM13.0 and the proposed for different quantization parameter (QP) values as suggested by JCTVC [35].

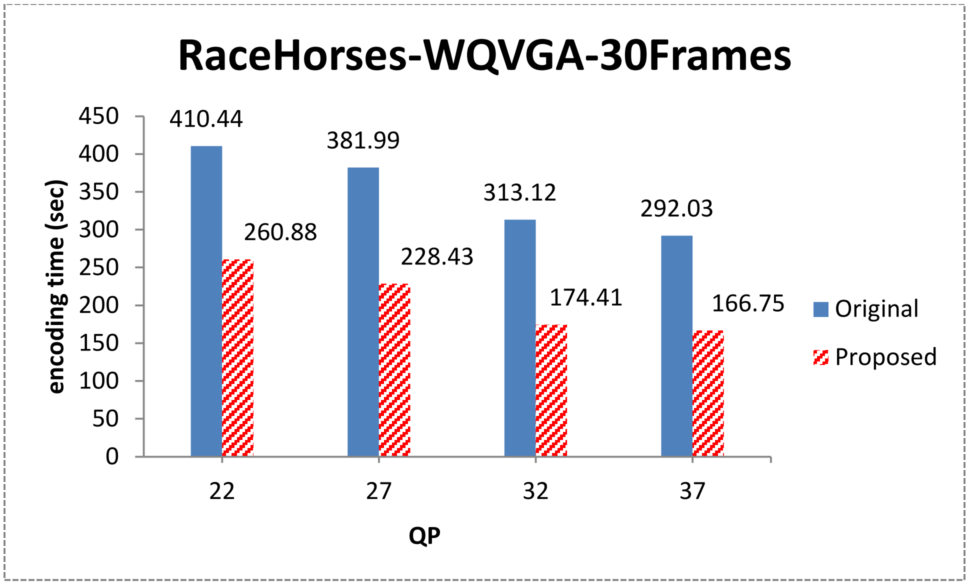


Figure 4-1 Encoding time vs. quantization parameter for Racehorses

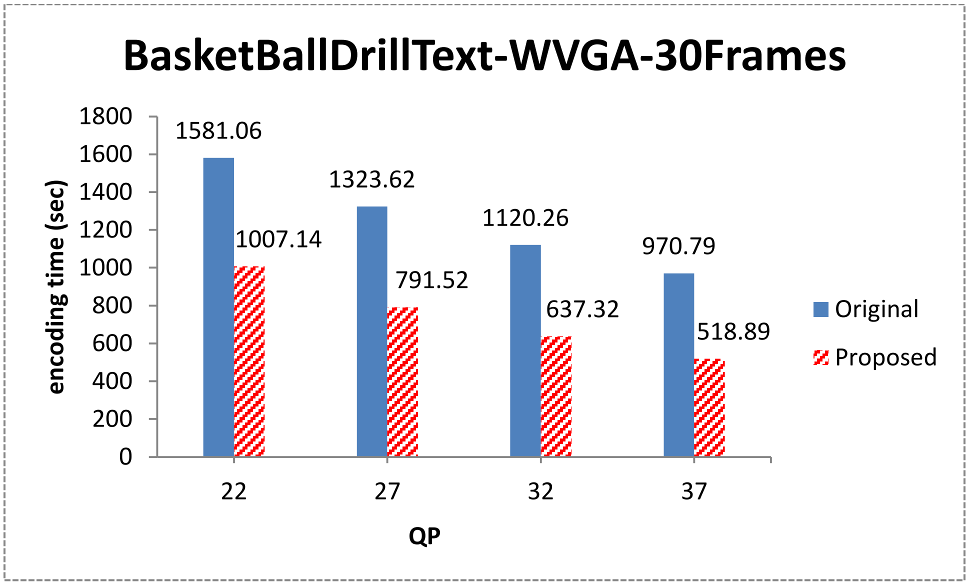


Figure 4-2 Encoding time vs. quantization parameter for BasketBallDrillText

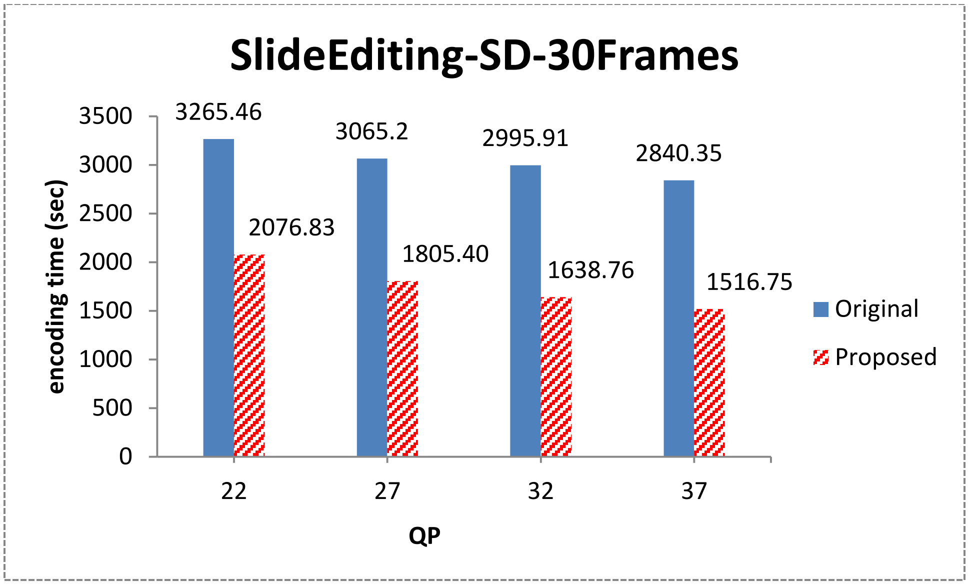


Figure 4-3 Encoding time vs. quantization parameter for SlideEditing

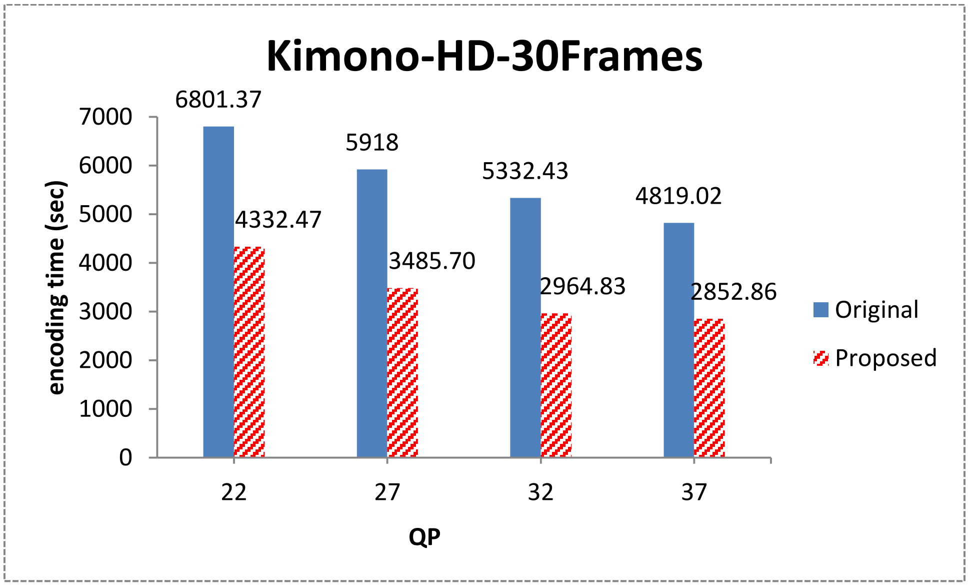


Figure 4-4 Encoding time vs. quantization parameter for Kimono

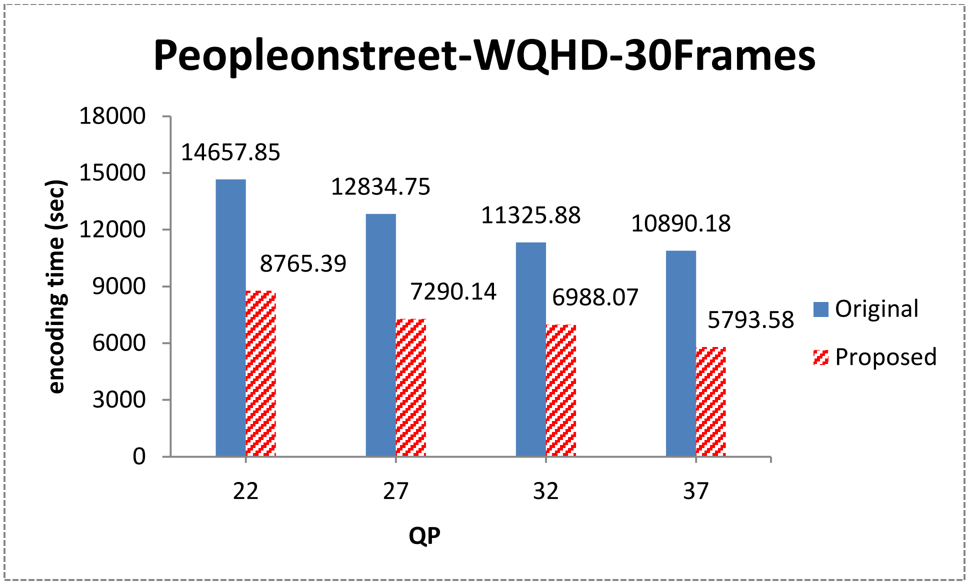


Figure 4-5 Encoding time vs. quantization parameter for PeopleOnStreet

4.3 BD-PSNR

To objectively evaluate the coding efficiency of video codecs, Bjøntegaard Delta PSNR (BD-PSNR) was proposed [36]. Based on the rate-distortion (R-D) curve fitting, BD-PSNR is able to provide a good evaluation of the R-D performance [36]. BD-PSNR is a curve fitting metric based on rate and distortion of the video sequence. However, this does not take into account the complexity of the encoder, but the BD metric tells a lot about the quality of the video sequence [30] [31]. Ideally, BD-PSNR should increase and BD-bitrate should decrease. The following results show a plot of BD-PSNR versus the quantization parameter (QP). It can be observed from figures 4-6 to 4-10 that there is a slight drop in PSNR using BD metrics for the proposed algorithm for the HM13.0 in the range of 0.25 dB to 0.48 dB.

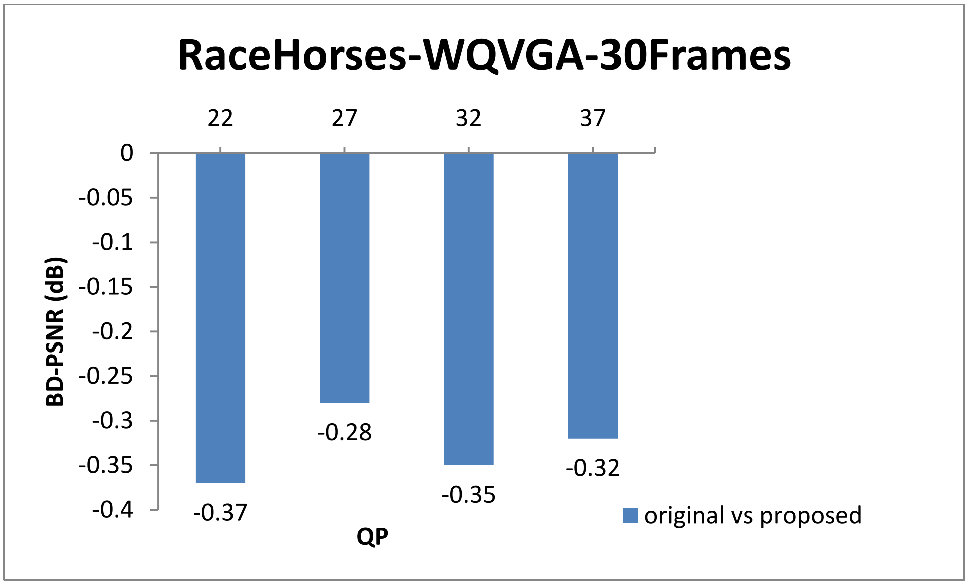


Figure 4-6 BD-PSNR vs. quantization parameter for RaceHorses

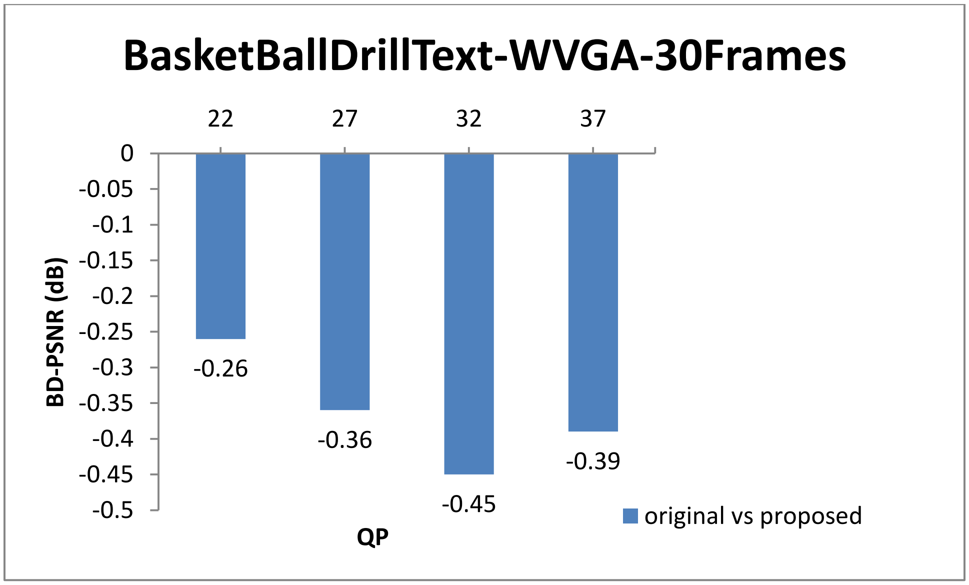


Figure 4-7 BD-PSNR vs. quantization parameter for BasketBallDrillText

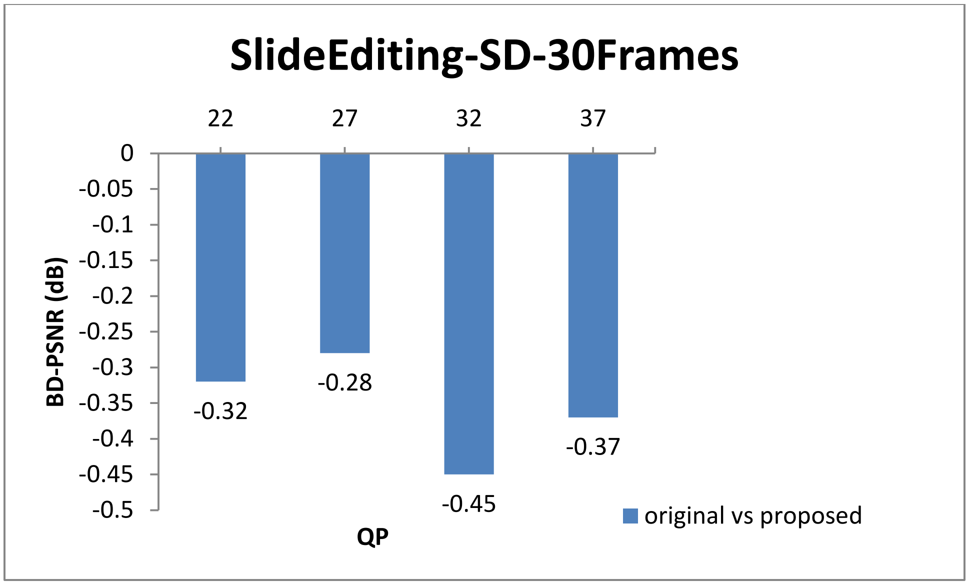


Figure 4-8 BD-PSNR vs. quantization parameter for SlideEditing

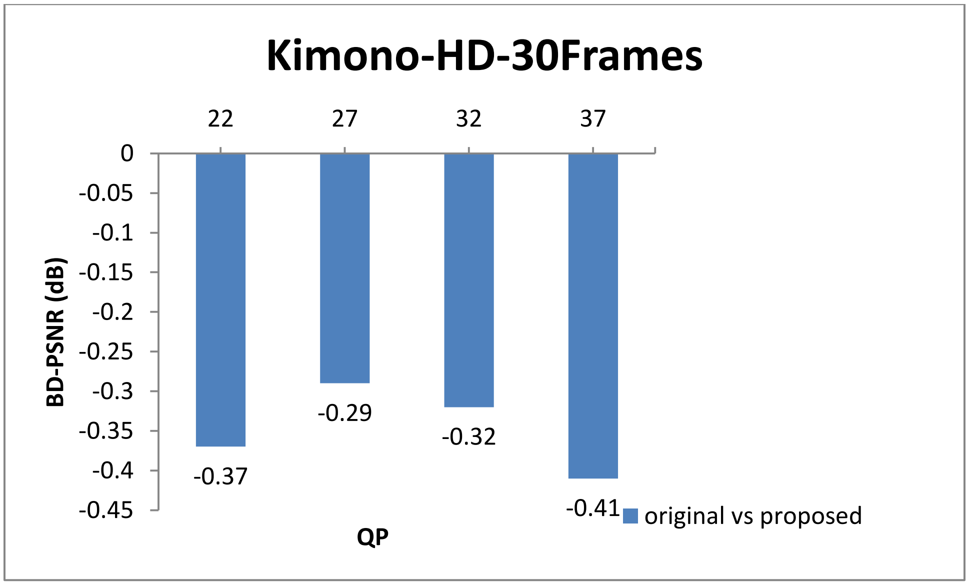


Figure 4-9 BD-PSNR vs. quantization parameter for Kimono

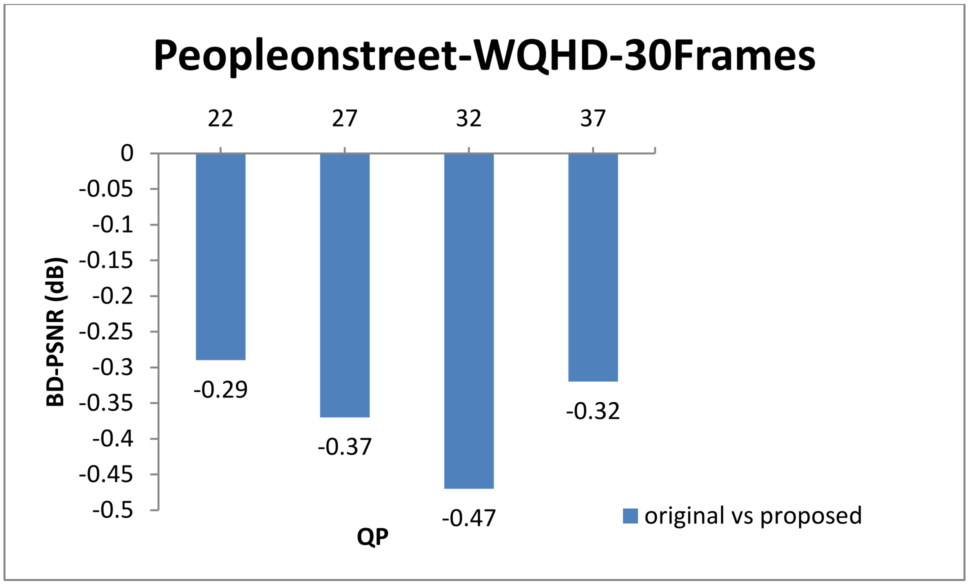


Figure 4-10 BD-PSNR vs. quantization parameter for PeopleonStreet

4.4 BD-Bitrate

BD-bitrate is a metric similar to the BD-PSNR metric which determines the quality of encoded video sequence along with the of measure the output bitstream of encoded video sequence. It can be observed from figures 4-11 to 4-15 that there is a slight increase in bitrate using the BD metrics for the proposed algorithm for HM13.0 in the range of 6 kbps to 11 kbps (b-Bit).

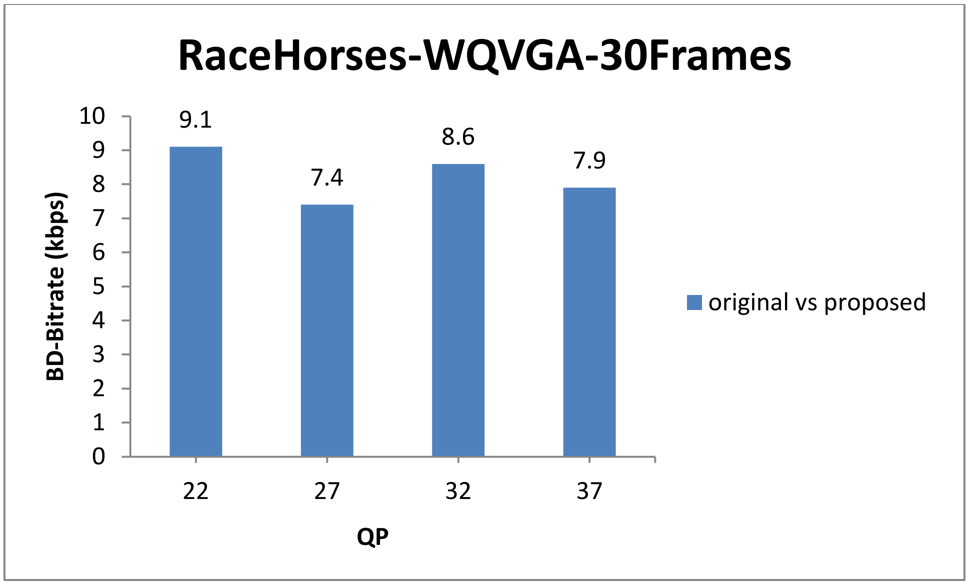


Figure 4-11 BD-bitrate vs. quantization parameter for RaceHorses

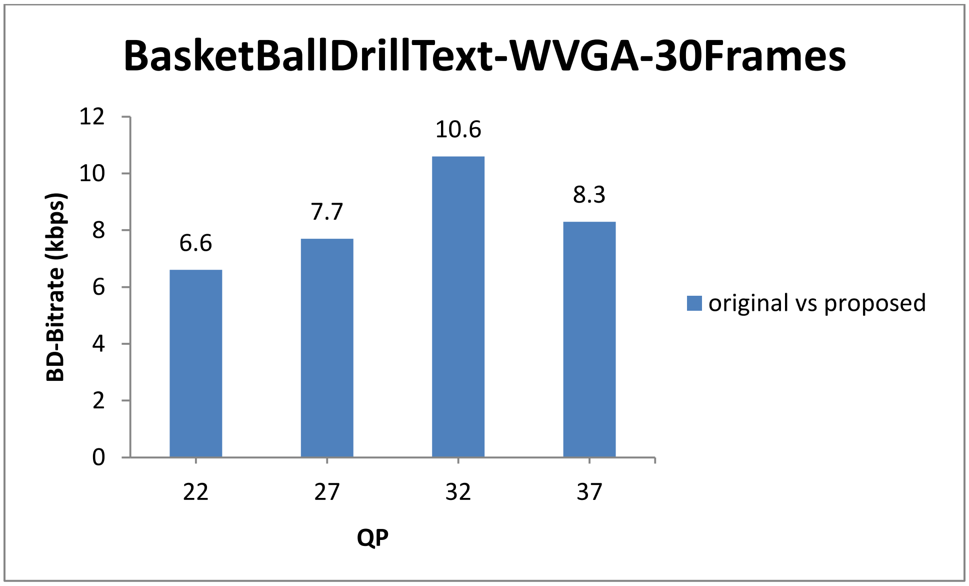


Figure 4-12 BD-bitrate vs. quantization parameter for BasketballDrillText

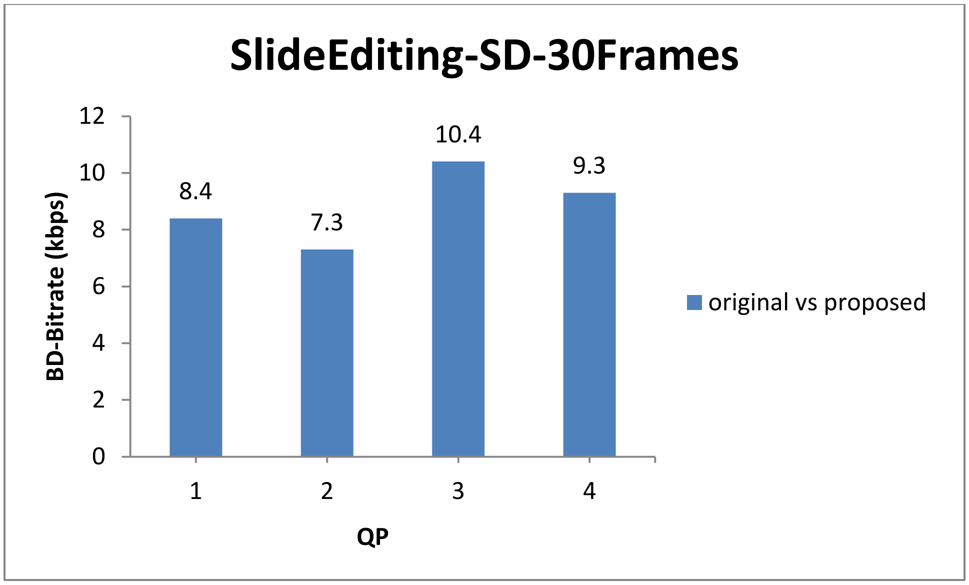


Figure 4-13 BD-bitrate vs. quantization parameter for SlideEditing

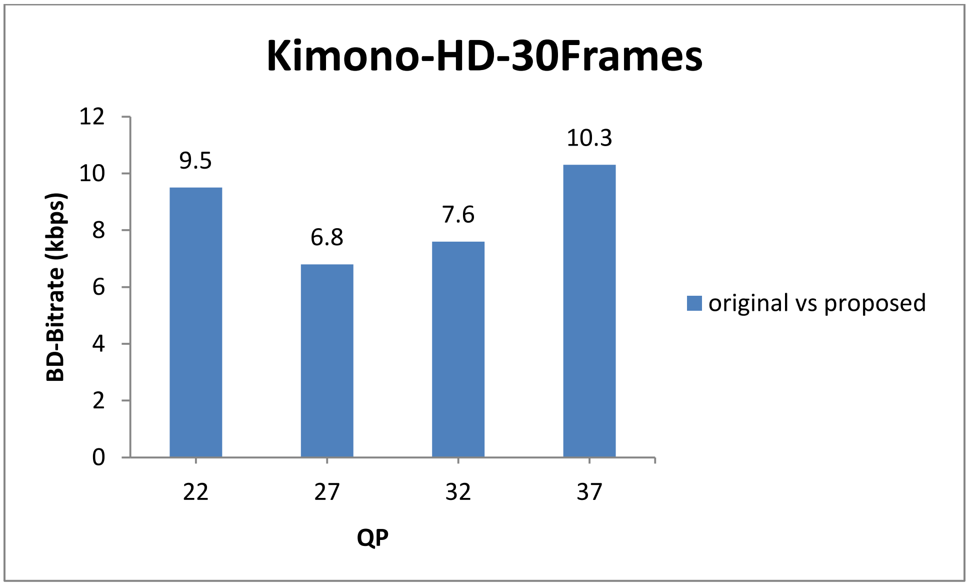


Figure 4-14 BD-bitrate vs. quantization parameter for Kimono

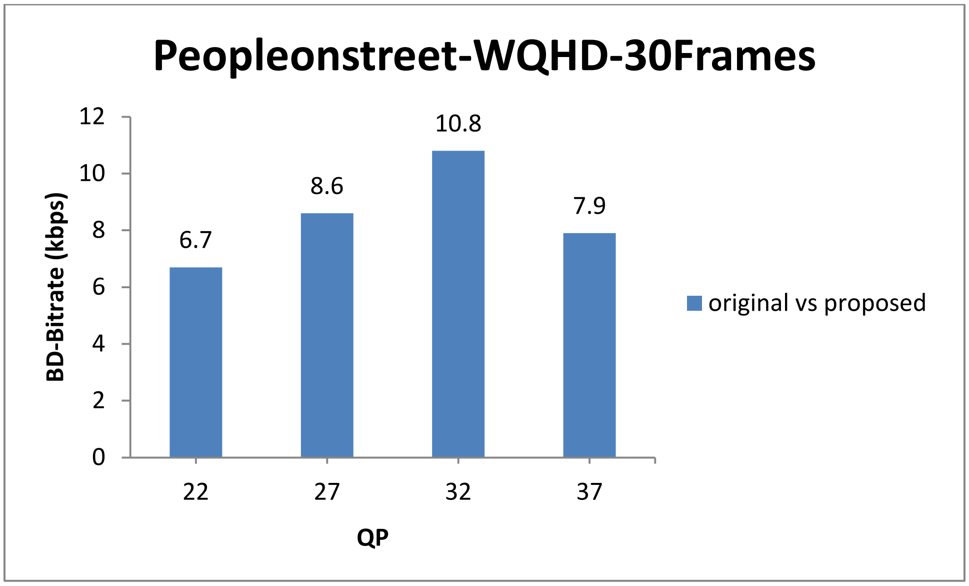


Figure 4-15 BD-bitrate vs. quantization parameter for PeopleOnstreet.

4.5 Rate Distortion Plot (RD Plot)

The proposed algorithm has negligible PSNR loss and bitrate increase. Figure 4-16 to 4-20 show bitrate-PSNR graphs for the various test sequences. It can be seen that performance is similar to the original HM13.0 encoder.

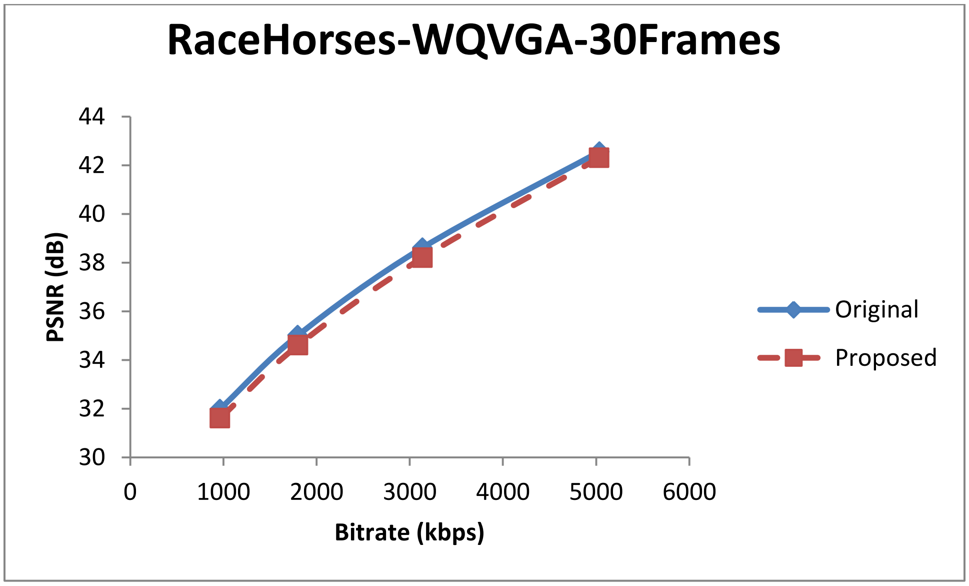


Figure 4-16 PSNR vs. bitrate for RaceHorses

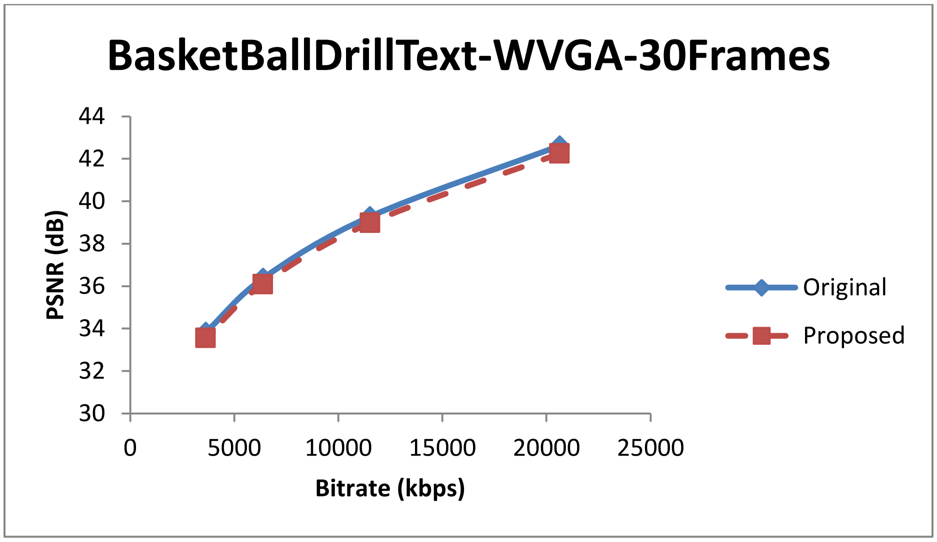


Figure 4-17 PSNR vs. bitrate for BasketBallDrillText

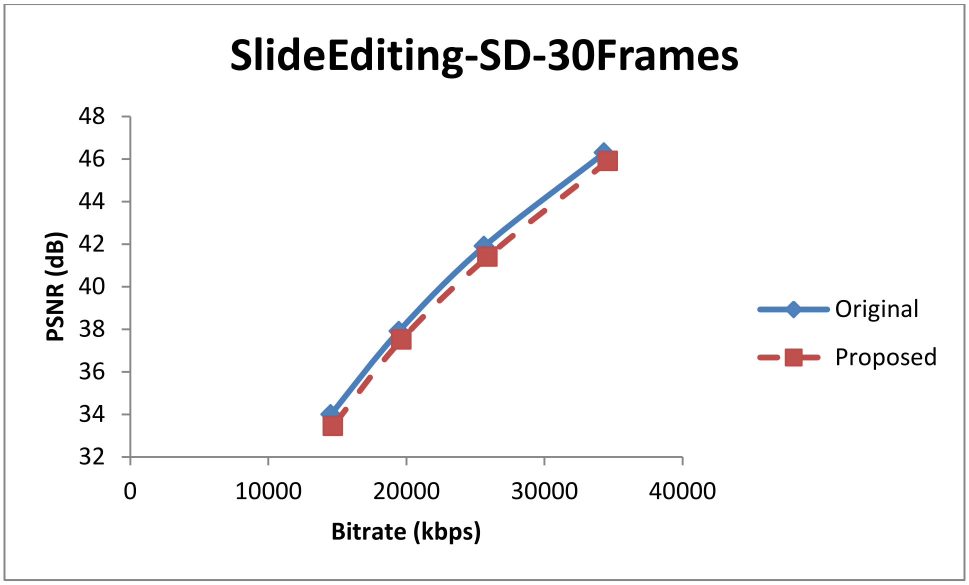


Figure 4-18 PSNR vs. bitrate for SlideEditing

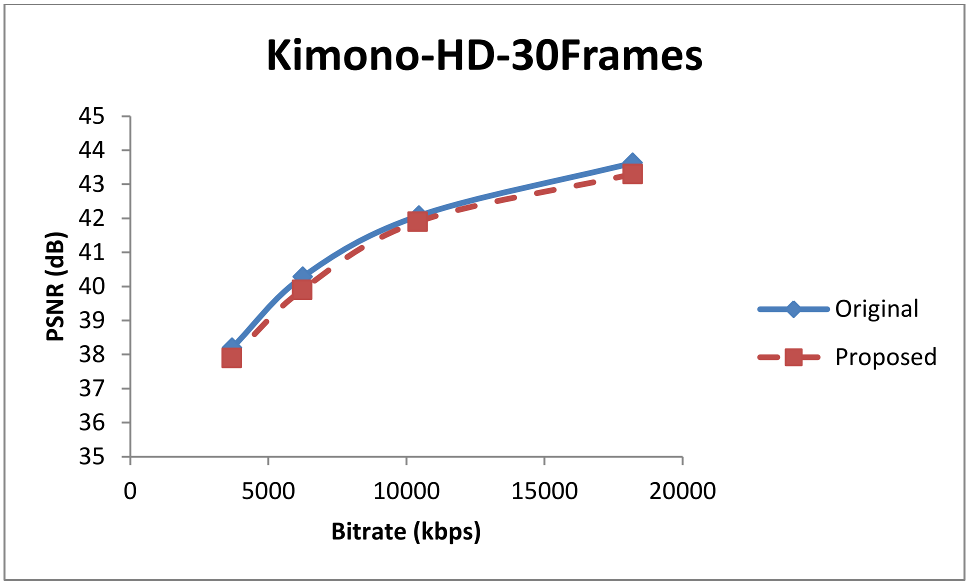


Figure 4-19 PSNR vs. bitrate for Kimono

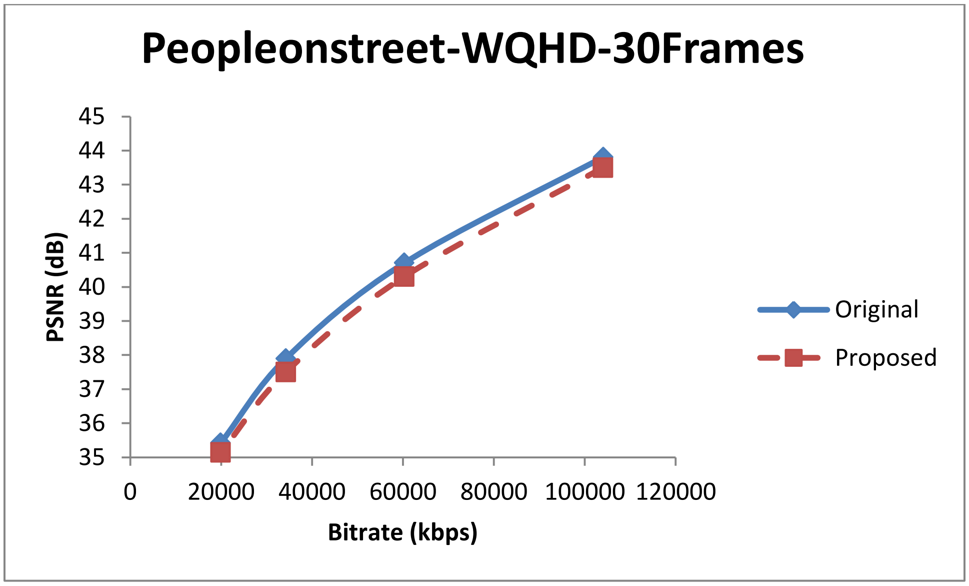


Figure 4-20 PSNR vs. bitrate for Peopleonstreet

4.6 Bitstream Size Gain

Figures 4-21 to 4-25 show the encoded bitstream size for the original HM13.0 and the proposed HM13.0 encoded for different quantization parameter values. It can be observed that there is only 1% to 4% increase in bitstream size.

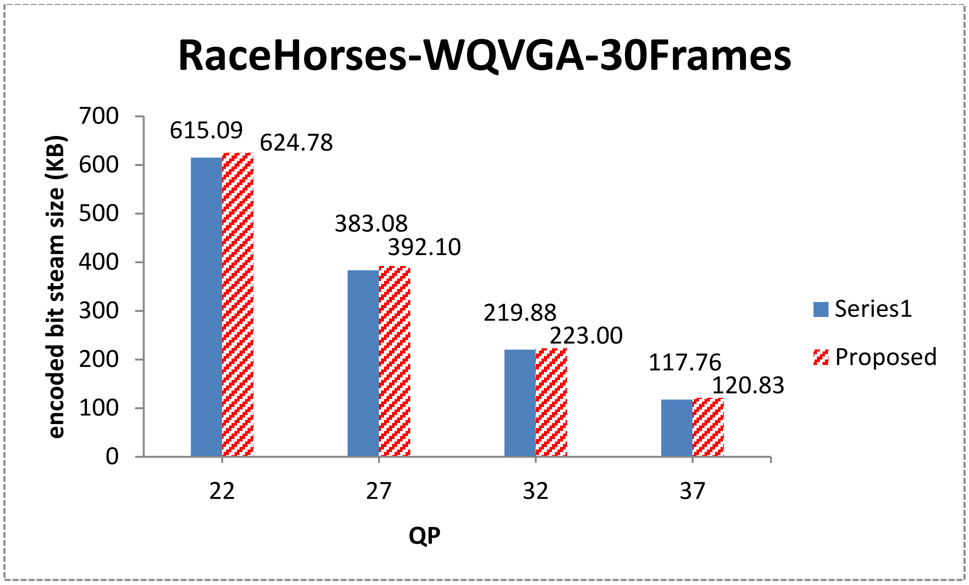


Figure 4-21 Encoded bitstream size vs. quantization parameter for RaceHorses

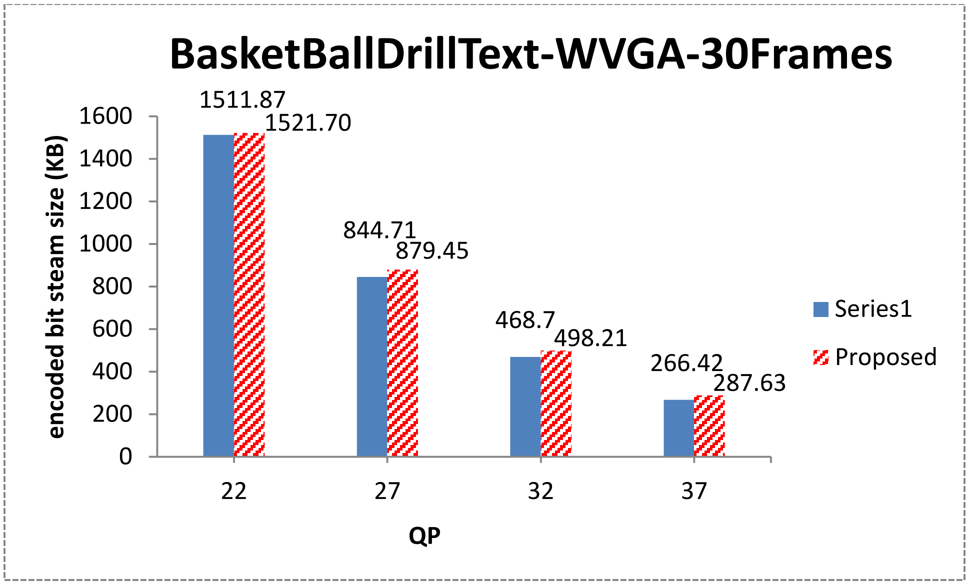


Figure 4-22 Encoded bitstream size vs. quantization parameter for BasketballDrilltext

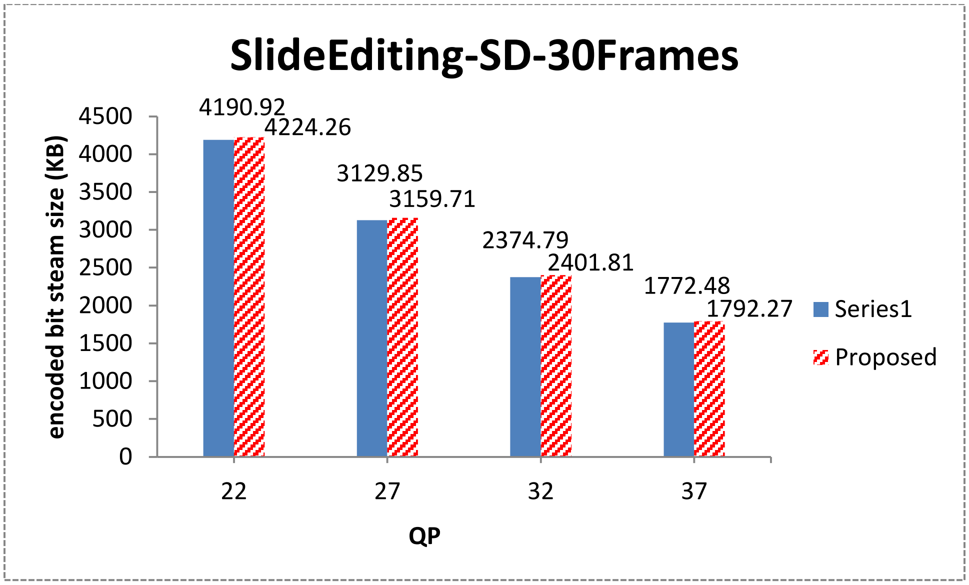


Figure 4-23 Encoded bitstream size vs. quantization parameter for SlideEditing

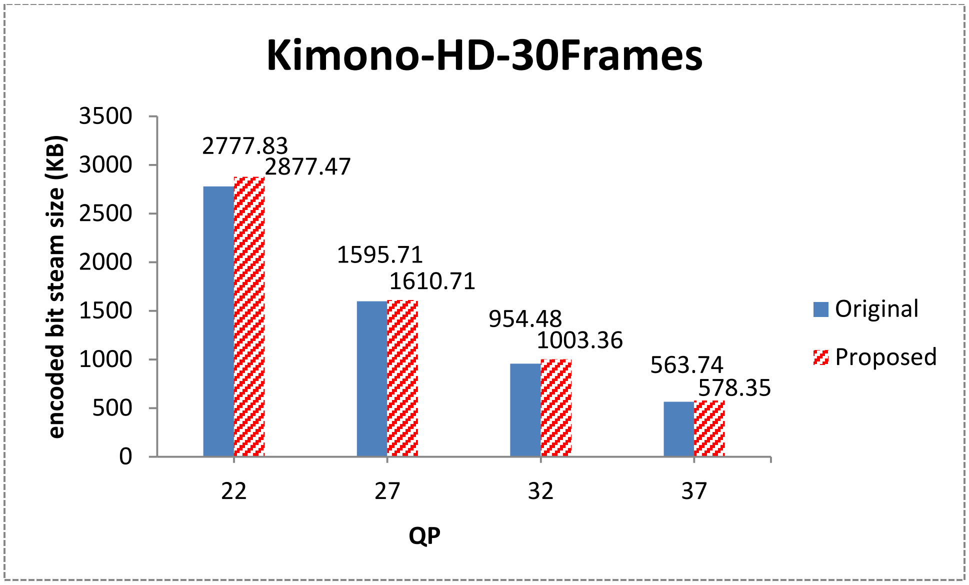


Figure 4-24 Encoded bitstream size vs. quantization parameter for Kimono

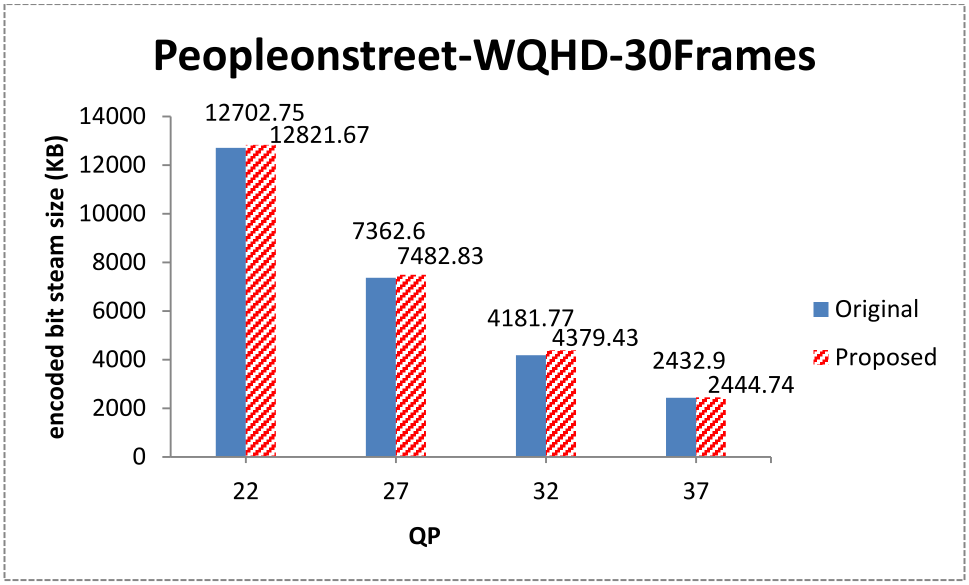


Figure 4-25 Encoded bitstream size vs. quantization parameter for Peopleonstreet

4.7 Percentage Decrease In Encoding Time

Figures 4-26 thru 4-30 show 35-48% decrease in encoding time which shows the decrease in the complexity of the encoder with proposed CU early termination algorithm as compared to the original HM13.0 algorithm [4].

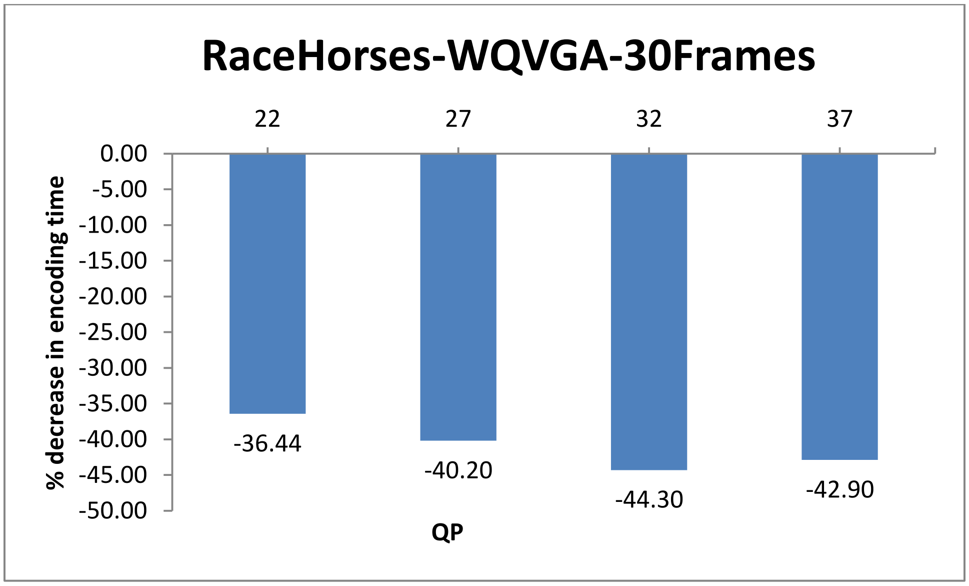


Figure 4-26 % decrease in encoding time vs. quantization parameter for RaceHorses

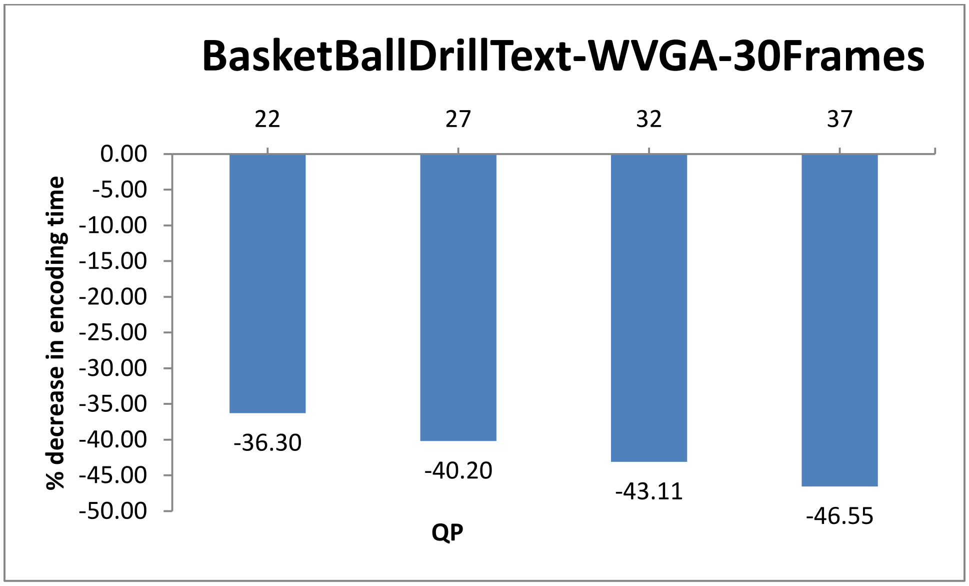


Figure 4-27 % decrease in encoding time vs. quantization parameter for BasketBalldrillText

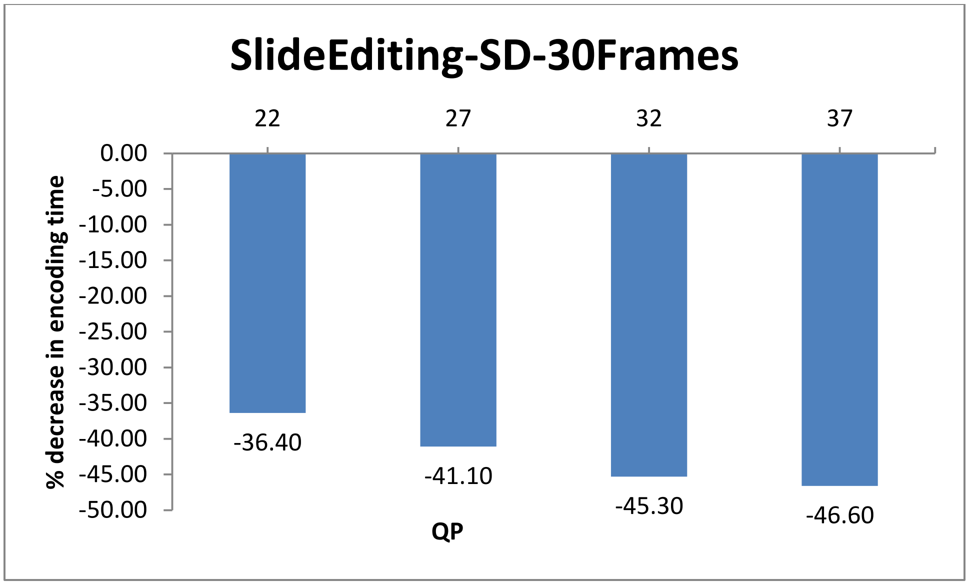


Figure 4-28 % decrease in encoding time vs. quantization parameter for SlideEditing

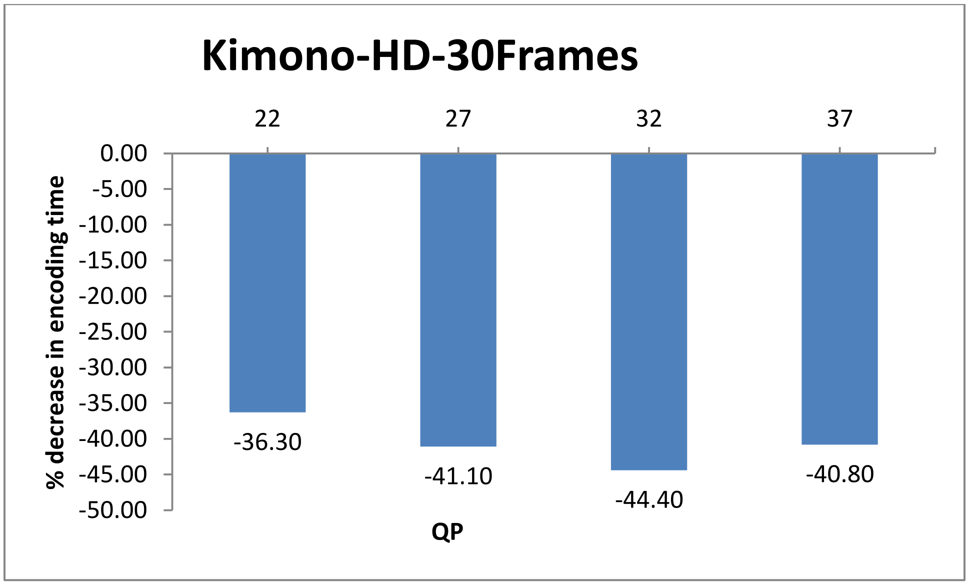


Figure 4-29 % decrease in encoding time vs. quantization parameter for Kimono

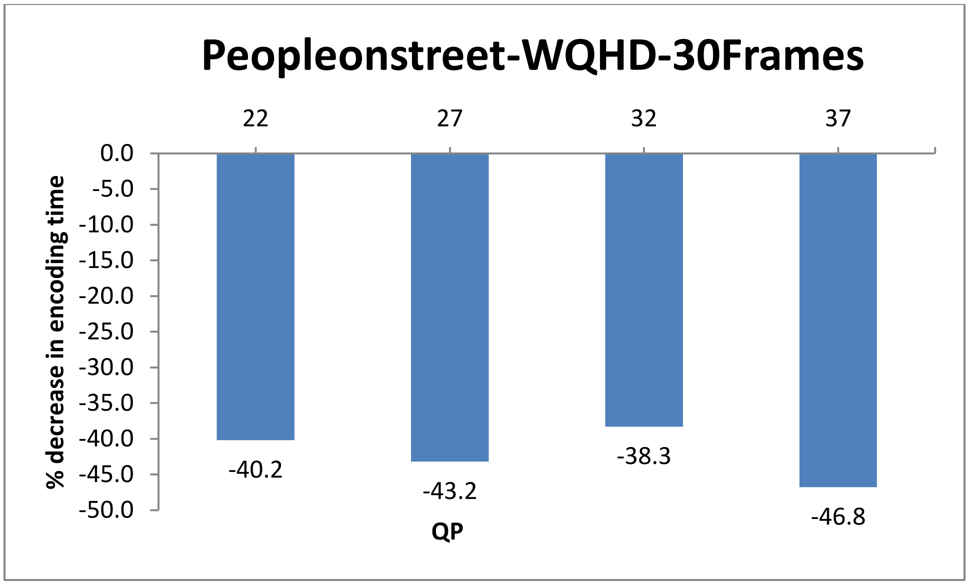


Figure 4-30 % decrease in encoding time vs. quantization parameter for Peopleonstreet

4.8 Summary

In this chapter, various results with graphs are described with and without implementation of the CU Early termination algorithm and the fast intra mode decision using various metrics such as encoding time, BD-PSNR, BD-bitrate, and bitstream size. In chapter 5, conclusions and future work are discussed.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this thesis a CU early termination algorithm and fast intra mode decision algorithm are proposed to reduce the computational complexity of the HEVC encoder, which includes three strategies, i.e., CU early termination, PU mode decision and early RDOQ termination. The results of comparative experiments demonstrate that the proposed algorithm can effectively reduce the computational complexity (encoding time) by 35-48% on average as compared to the HM 13.0 encoder [4], while only incurring a slight drop in the PSNR and a negligible increase in the bitrate and encoding bitstream size for different values of the quantization parameter based on various standard test sequences [29]. The results of simulation also demonstrate negligible decrease in BD-PSNR [30] i.e. 0.25 dB to 0.48 dB as compared to the original HM13.0 software and 6 kbps to 11 kbps increase in the BD-bitrate [31].

5.2 Future Work

There are many other ways to explore in the CU early termination and fast intra prediction in the intra prediction area as suggested by research [25][33]. Many of these methods can be combined with this method, or if needed, one method may be replaced by a new method and encoding time gains can be explored.

Similar algorithms can be developed for fast inter-prediction in which the RD cost of the different modes in inter-prediction are explored, and depending upon the adaptive threshold [34], mode decision can be terminated resulting in less encoding time and reduced complexity combining with the above proposed algorithm.

Tan et al [37] proposed a fast RQT algorithm for both intra and inter mode coding in order to reduce the encoder complexity. In [37], for all intra case, 13% encoding time can be saved, However, BD-Rate just increases by 0.1%. For random access and low delay constraints it reduces by up to 9% encoding time with 0.3% BD-Rate performance degradation. This method can be integrated with the proposed algorithm to increase the encoding time.

Tian et al [38] proposed a PU size decision algorithm to speed up the intra coding. In this method, two-stage is applied. In the pre-stage, filtering the unnecessary PU by analyzing the texture complexity of the LCU and its four sub-blocks secondly, skipping the small PU candidates by referring the neighboring PU. The simulation results show that proposed method can speed up by average of 44.91%, with only PSNR degradation less than 0.04dB. This method can be combined with the proposed algorithm.

The Bayesian decision [39] rule can be applied to calculate the CU size, and then this information can be combined with the proposed method to achieve further encoding time gains.

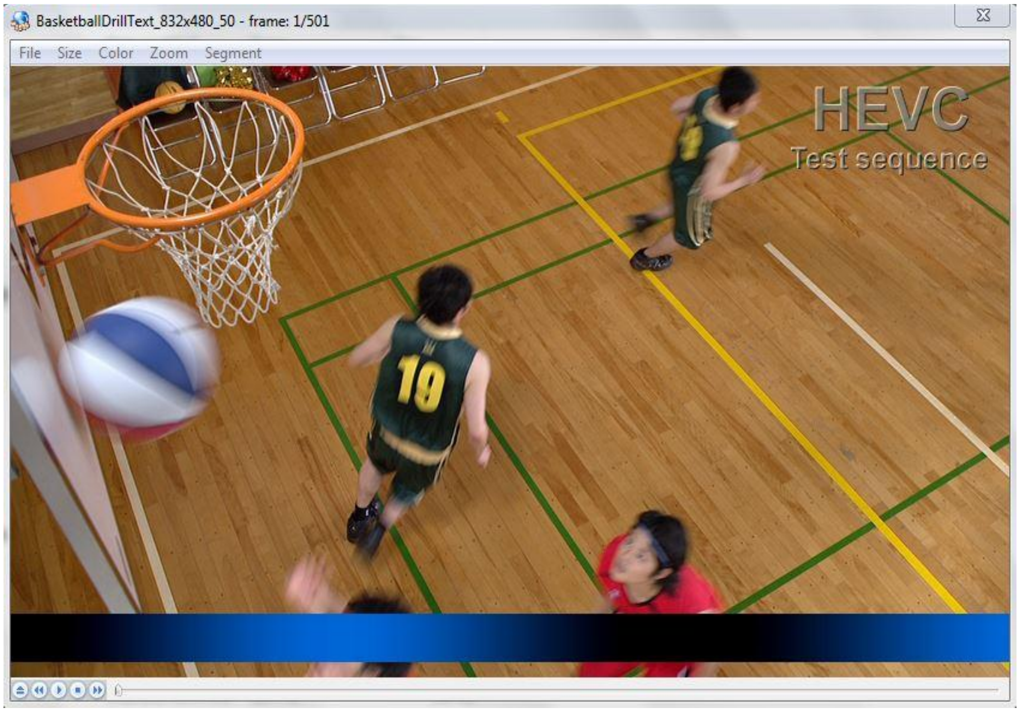
Complexity reduction can also be achieved through hardware implementation of a specific algorithm which requires much computation. The FPGA implementation can be useful to evaluate the performance of the system on hardware in terms of power consumption and encoding time.

Appendix A

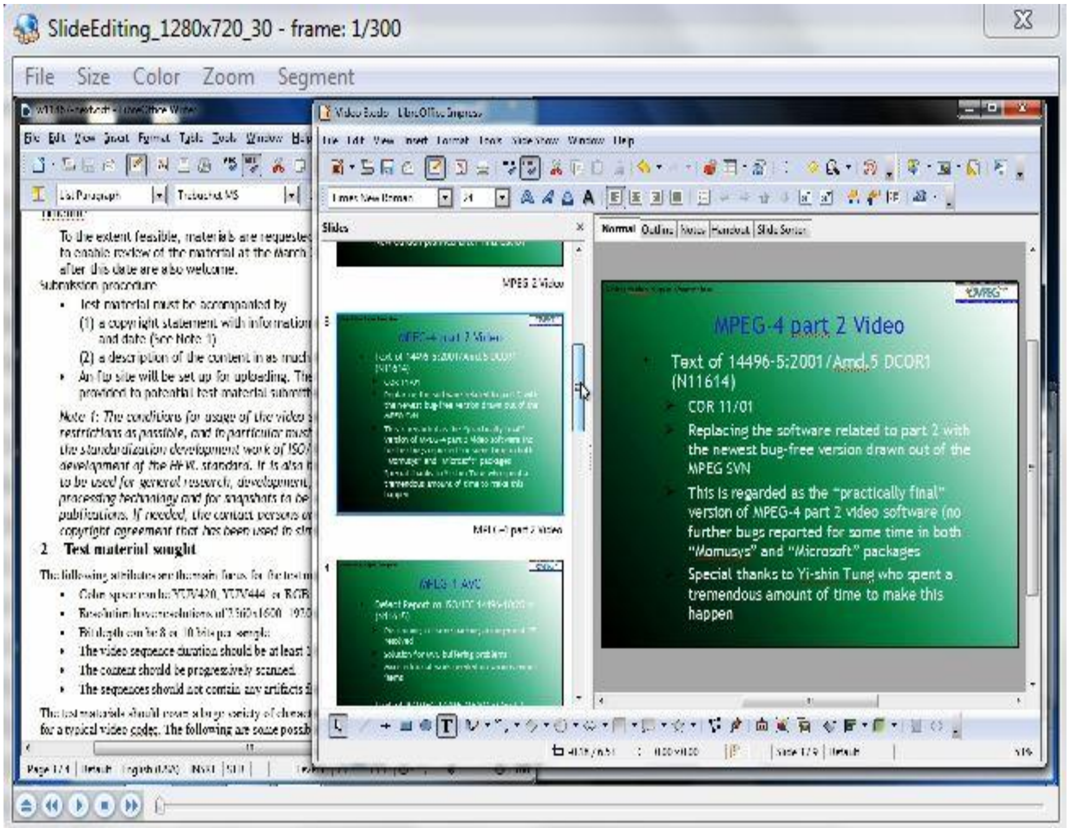
Test Sequences [29]

A.1 Racehorses

A.2 BasketBallDrillText



A.3 SlideEditing



To the extent feasible, materials are requested

after this date are also welcome.

Submission procedure

- Test material must be accompanied by
  - (1) a copyright statement with information
  - and date (see Note 1)

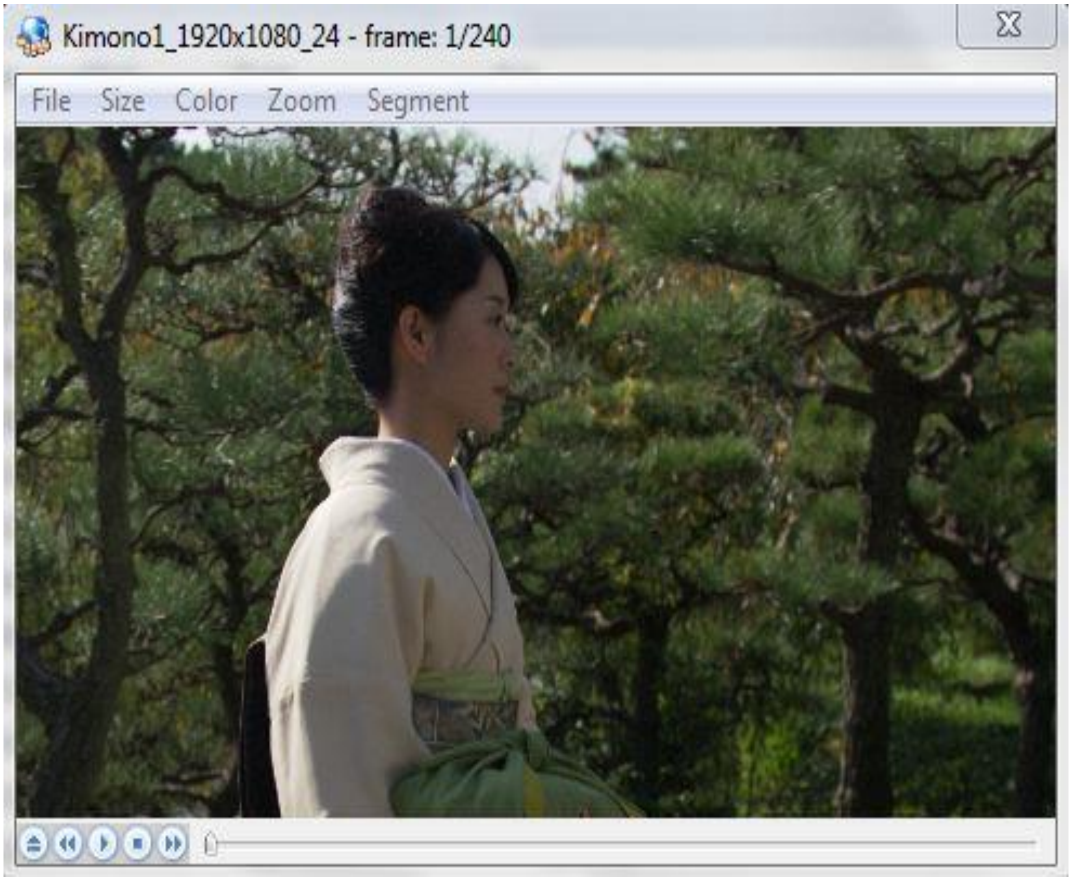
2 Test material sought

- The content should be progressively scanned.

MPEG-4 part 2 Video

- Text of 14496-5:2001/Amd.5 DCOR1 (N11614)
  - COR 11/01
  - Replacing the software related to part 2 with the newest bug-free version drawn out of the MPEG SVN
  - This is regarded as the "practically final" version of MPEG-4 part 2 video software (no further bugs reported for some time in both "Momusys" and "Microsoft" packages
  - Special thanks to Yi-shin Tung who spent a tremendous amount of time to make this happen

A.4 Kimono


Kimono1_1920x1080_24 - frame: 1/240
File
Size
Color
Zoom
Segment

A.5 PeopleOnStreet

Appendix B

Test Conditions

The code revision used for this work is revision HM13.0 [4]. The work was done using an Intel Core i5 processor running at 2.50 GHz, with Microsoft Windows 7 64 bit version running with 6 GB of RAM.

Appendix C

BD-PSNR And BD-Bitrate [30] [31]

BD-PSNR (Bjontegaard – PSNR) and BD-bit rate (Bjontegaard – bit rate) metrics are used to compute the average gain in PSNR and the average per cent saving in bit rate between two rate-distortion graphs respectively and is an ITU-T approved metric [30]. 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 [31].

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

% BJONTEGAARD    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 verifyed 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
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 (fixed 20130515)
      min_int = max([ min(lR1); min(lR2) ]);
      max_int = min([ max(lR1); max(lR2) ]);


      % 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,lR1,3);
      p2 = polyfit(PSNR2,lR2,3);

      % 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

BD - Bjontegaard Delta

CABAC – Context Adaptive Binary Arithmetic Coding

CB – Coding Block

CBF – Coding Block Flag

CTU – Coding Tree Unit

CTB – Coding Tree Block

CU – Coding Unit

DCT – Discrete Cosine Transform

DST – Discrete Sine Transform

ECU – Early Coding Unit

ET – Early Termination

FDIS – Final Draft International Standard

HD– High Definition

HEVC – High Efficiency Video Coding

HM – HEVC Test Model

ICCE – International Conference on Consumer Electronics

ISO – International Standards Organization

ITU – International Telecommunications Union

JCT-VC - Joint Collaborative Team on Video Coding

MPEG – Moving Picture Experts Group

MPM – Most Probable Modes

PU – Prediction Unit

QP – Quantization Parameter

RDOQ – Rate Distortion Optimization Quantization

SATD –Sum of Absolute Transform Differences

SD – Standard Definition

VCEG – Video Coding Experts Group

VPS – Video Parameter Set

WQHD – Wide Quarter High Definition

WQVGA – Wide Quarter Video Graphics Array

WVGA – Wide Video Graphics Array

References

1. G. Sullivan et al., “Overview of the high efficiency video coding (HEVC) standard”, “IEEE Transactions on Circuits and Systems for Video Technology, Vol. 22, No. 12, pp 1649-1668, December 2012
2. I. E. Richardson, “The H.264 advanced video compression standard”, 2nd Edition, Wiley 2010.
3. G. Escribano et al, “Video encoding and transcoding using machine learning,” MDM/KDD’08, Las Vegas, NV, USA, August 24, 2008.
4. HM13.0, HEVC code: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-13.0/
5. Open source article, “Blu-ray discs”, http://www.blu-ray.com/info/
6. Open source article, ”Coding of moving pictures and audio” http://mpeg.chiariglione.org/standards/mpeg-2/mpeg-2.htm
7. R. Schafer and T. Sikora, “Digital video coding standards and their role in video communications”, Proceedings of the IEEE, Vol 83, pp. 907-923, Jan. 1995.
8. Open source article, “Studio encoding parameters of digital television for standard 4:3 and wide screen 16:9 aspect ratios”, http://www.itu.int/rec/R-REC-BT.601/
9. HEVC software manual:https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/branches/HM-13.0-dev/doc/software-manual.pdf
10. K. Iguchi et al, “HEVC encoder for super hi-vision”, IEEE ICCE, pp. 61-62, Las Vegas, NV, Jan. 2014.
11. MPL Website: http://www-ee.uta.edu/dip/
12. G J. Sullivan et al, “Standardized Extensions of HEVC”, IEEE Journal of Selected topics in Signal Processing, Vol.7, No.6, pp.1001-1016, Dec. 2013.

13. 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.
14. M.T.Pourazad et al, “HEVC: The new gold standard for video compression”, IEEE consumer electronics magazine, vol.1, no.7, pp.36-46, July 2012.
15. F. Bossen et al, “HEVC Complexity and Implementation Analysis”, IEEE Trans. on circuits and systems for video technology, vol.22, no.12, pp.1685-1696, Dec. 2012.
16. Basics of video: http://lea.hamradio.si/~s51kq/V-BAS.HTM
17. B. Bross et al, “HM9: High Efficiency Video Coding (HEVC) Test Model 9 Encoder Description”, JCTVC-K1002v2, October 2012. [Online]. Available: http://phenix.it-sudparis.eu/jct/doc_end_user/current_document.php?id=6807
18. B. Bross et al, “High Efficiency Video Coding (HEVC) text specification draft 9 (SoDIS)”, JCTVC-K1003v13, October 2012. [Online]. Available: http://phenix.it-sudparis.eu/jct/doc_end_user/current_document.php?id=6803
19. W. Jiang, H. Ma and Y. Chen, “Gradient based fast mode decision algorithm for intra prediction in HEVC”, 2nd International Conference on Consumer Electronics, Communications and Networks, pp. 1836-1840, 21-23 April 2012.
20. S. Lui et al, “Video Prediction Block Structure and the Emerging High Efficiency Video Coding Standard”, IEEE proceedings on Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific, pp. 1-4, 2012.
21. G. Sullivan et al, “Efficient quadtree coding of images and video”, IEEE Transactions on Image Processing, vol. 3, pp. 327-331, May 1994.

22. P. Helle et al, “Block merging for quadtree-based partitioning in HEVC”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, pp. 1720-1731, May 2012.
23. K. Choi et al, “Fast coding unit decision method based on coding tree pruning for high efficiency video coding”, Proc. SPIE Optical Engineering, vol. 51, 030502, March 2012.
24. R. Li et al, ”A new three-step search algorithm for block motion estimation”, IEEE Transactions on Circuits and Systems for Video Technology, vol.4, pp.438-442, Aug.1994.
25. H. Zhang and Z. Ma, “Fast intra prediction for high efficiency video coding”, Advances in Multimedia Information Processing, 13th Pacific-Rim Conference on Multimedia, Proceedings, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 7674 LNCS, pp. 568-577, 4-6 Dec. 2012.
26. G. Tian et al, “Content adaptive prediction unit size decision algorithm for HEVC intra coding”, Picture Coding Symposium (PCS), pp.405-408, May 2012.
27. H. Lei and Z. Yang, “Fast Intra Prediction Mode Decision for High Efficiency Video Coding”. 2nd International Symposium on Computer, Communication, Control and Automation, Nov 2013.
28. H. Zhang and Z. Ma, “Fast intra prediction for high efficiency video coding”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 24, No. 4, pp 660-668, April 2014.
29. HEVC test sequences: ftp://ftp.tnt.uni-hannover.de/testsequences
30. G. Bjontegaard, “Calculation of average PSNR differences between RD-curves”, Q6/SG16, Video Coding Experts Group (VCEG), 2-4, April 2001.

31. BD metrics code - http://www.mathworks.com/matlabcentral/fileexchange/27798-bjontegaardmetric/content/bjontegaard.m
32. 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.
33. S. Vasudevan and K. R. Rao “Combination method of fast HEVC encoding” IEEE ECTICON 2014, Korat, Thailand, May 2014.
34. K. Shah, “Reducing the complexity of inter prediction mode decision for HEVC”, website: http://www-ee.uta.edu/Dip/Courses/EE5359/KushalShah_Thesis.pdf, Thesis, University of Texas at Arlington, UMI Dissertion Publishing, April 2014.
35. JCT-VC documents: http://www.itu.int/en/ITU-T/studygroups/2013-2016/16/Pages/video/jctvc.aspx
36. X. Li et al, “Rate-Complexity-Distortion evaluation for hybrid video coding”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 21, pp. 957 - 970, July 2011.
37. Y. H. Tan et al, ”On residual quad-tree coding in HEVC” 2011 IEEE 18th International Conference on Image Processing (ICIP), pp. 3469-3472, Sept. 2011.
38. G. Tian et al,”Content adaptive prediction unit size decision algorithm for HEVC intra coding”, Picture Coding Symposium (PCS) , pp.405-408, May 2012.
39. X. Shen et al, “Fast coding unit size selection for HEVC based on Bayesian decision rule”, IEEE Picture Coding Symposium (PCS), pp. 453-456, May 2012.
40. Special issue on emerging research and standards in next generation video coding, IEEE Transactions on Circuits and Systems for Video Technology (CSVT), vol. 22, pp. 1646-1909, Dec. 2012.
41. R. H. Gweon et al, “Early termination of CU encoding to reduce HEVC complexity”, document JCTVC-F045, Torino, Italy, July 2011.

42. A.S. Motra et al, “Fast intra mode decision for HEVC video encoder”, 2012 20th International Conference on Software, Telecommunications and Computer Networks, SoftCOM, 11-13 Sept. 2012.
43. L. Zhao et al, “Fast mode decision algorithm for intra prediction in HEVC”, Conference Article no.6115979, IEEE Visual Communications and Image Processing, 6-9 Nov. 2011.
44. T.L. Da Silva et al, “HEVC Intra Coding Acceleration based on Tree-Inter level Mode Correlation”, IEEE and SPA, 26-28 Sept. 2013.
45. S.W. Teng et al, “Fast mode decision algorithm for residual quadtree coding in HEVC”, Conference Article no.6116062, IEEE Visual Communications and Image Processing, 6-9 Nov. 2011.

Biographical Information

Harshdeep Brahmasury Jain was born in Tumkur, Karnataka, India in 1990. After completing his schooling at Kendriya Vidyalaya, Tumkur in 2005, he went on to obtain his Bachelors of Engineering in Electronics and Telecommunication from Visvesvaraya Technological University, India from 2007-2011. After that he worked for Indian Institute of Science (IISC), Bangalore as a Research Assistant.

He joined the University of Texas at Arlington, USA to pursue his M.S in Electrical Engineering in Fall 2012. This was around the time he joined the Multimedia Processing Lab. He worked as Graphics Software Intern in Intel Corporation, Folsom, CA and subsequently will join Intel Corporation after graduation.