HDR VIDEO COMPRESSION USING H.265 (HEVC) MAIN 10 PROFILE

by

SRIKANTH VASIREDDY

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

May 2016

ACKNOWLEDGEMENTS

First and foremost, I would like to thank Dr. K.R Rao for being a guide, mentor and constant source of encouragement throughout my thesis. He has been a pillar of support throughout my Master's degree program and constant source of inspiration for me to pursue this research work. I would like to thank Dr.Ioannis D. Schizas and Dr. Howard Russell for serving as members of my graduate committee.

I would like to express my gratitude to Dr. Pankaj Topiwala for giving me an internship opportunity to carry out the research work. I would like to thank my mentor Madhu Peringassery Krishnan for guiding and supporting me throughout the work. He has always been there to discuss and provide suggestions.

I would like to thank my MPL lab mates: Tuan Ho, Shiba Kuanar, Deepak and Rohith for providing valuable inputs throughout my research.

Finally, I thank my parents, family members and friends for their love, encouragement and support.

April 14, 2016

ABSTRACT

HDR VIDEO COMPRESSION USING H.265 (HEVC) MAIN 10 PROFILE

Srikanth Vasireddy, MS

The University of Texas at Arlington, 2016

Supervising Professor: K. R. Rao

Increasing the resolution of the video has been the main interest in the television industry for a long time. More recently the industry is studying techniques to increase the range of luminance and color representation in videos. This is achieved through High Dynamic Range/ Wide Color Gamut (HDR/WCG) videos by providing visual quality of experience to the end consumers close to that of real life compared to the Standard Dynamic Range (SDR) videos.In order to deliver high-quality video services to consumers, efficient compression techniques are required to store or transmit a video especially for high resolution video formats whilst maintaining an 'acceptable' level of video quality. One such technology is High Efficiency Video Coding (HEVC) which is the current state-of-art video codec providing a higher compression capability and is widely being adopted by lot of users.

The primary focus of the thesis is to investigate different approaches for HDR video compression and implement a scheme using a pre/post processing scheme with the HM-16.7 software as a part of this research. In this study, the quality of HDR video using proposed scheme with the original HDR video, both compressed by the HEVC standard is evaluated using different objective metrics (tPSNR-X, tPSNR-Y, tPSNR-XYZ, tOSNR-XYZ, PSNR_DE100 and PSNR_L100) for different test sequences. Also the visual quality tests are performed for different test sequences. The thesis concludes with a discussion and possible directions for future work.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1. Motivation

Today's multimedia environment is growing with:

- Increasing HD -1920x1080 and UHD (4Kx2K, 8Kx4K) video content
- State of the art video coding technologies such as High Efficiency Video Coding (HEVC) [2]
- Multiple dimensions to increase the data rate
- Consumer internet video traffic which will be 80% of all consumer internet traffic in 2019 [25].

In order to deliver high quality video services to consumers, efficient compression techniques are required to store or transmit a video especially for HD and UHD resolutions whilst maintaining an 'acceptable' level of video quality. Thus, the research mainly aims at compression of the HD video content using the current state- of-art video codec HEVC which is widely being adopted by lot of users.

1.2. Basics of Video Compression

In today's technological world, all of us are surrounded with devices varying from handheld mobiles to powerful high end devices which are capable of capturing, processing, transmitting, receiving and displaying videos and multimedia files. 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 significant role in our day to day life which relies on video coding technologies.

Video is nothing but a sequence of images displayed at a constant interval and Image is collection of pixels where each pixel comprises of brightness and color components [2]. The main attributes of a video are Height, Width, Frame rate and Pixel values.

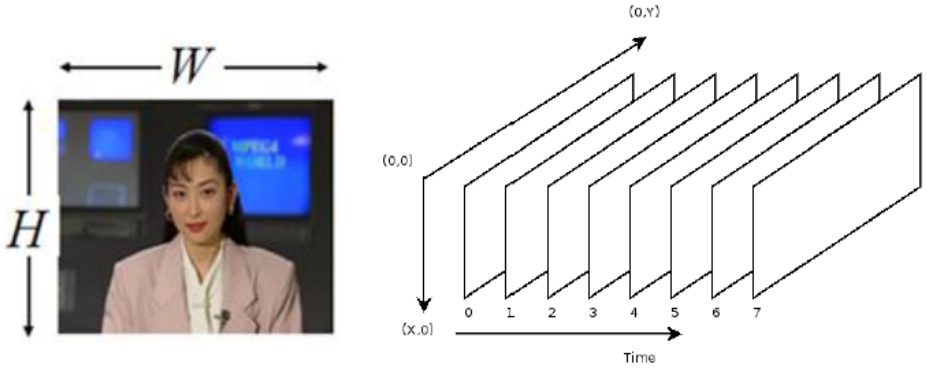


Figure 1-1 : Video and its attributes [1] [7]

The common color spaces for digital image and video representation are:

- RGB color space – Each pixel is represented by three numbers indicating the relative proportions of red, green and blue colors
- YC_bC_r color space – Y is the luminance component, a monochrome version of color image. Y is a weighted average of R, G and B:

  $Y=k_r R + k_g G + k_b B$, where k are the weighting factors.

The color information is represented as color differences or chrominance components, where each chrominance component is difference between R, G or B and the luminance Y.

As the human visual system is less sensitive to color than the luminance component, YC_bC_r has advantages over RGB space. The amount of data required to represent the chrominance component reduces without impairing the visual quality [1].The sampling patterns are shown in Figure 1-2.

The popular patterns of sampling [1] are:

- *4:4:4* – The three components Y: C_r: C_b have the same resolution, which is for every 4 luminance samples there are 4 C_r and 4 C_b samples.
- *4:2:2* – For every 4 luminance samples in the horizontal direction, there are 2 C_r and 2 C_b samples. This representation is used for high quality video color reproduction.
- *4:2:0* – The C_r and C_b each have half the horizontal and vertical resolution of Y. Figure1-3 represents the 4:2:0 sampling pattern. This is popularly used in applications such as video conferencing, digital television and DVD storage.

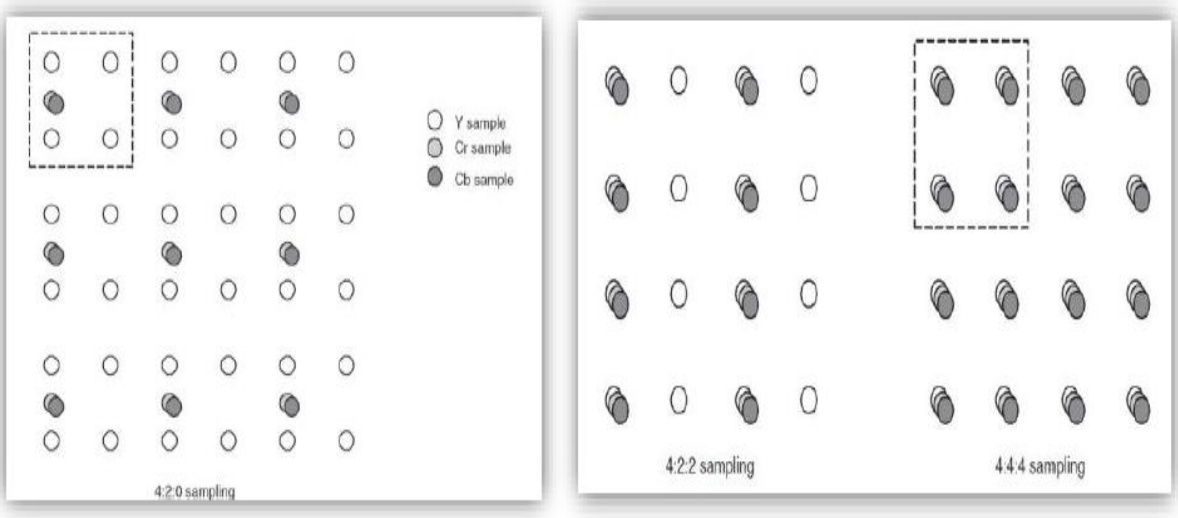


Figure 1-2 : 4:2:0, 4:2:2 and 4:4:4 sampling patterns [1]

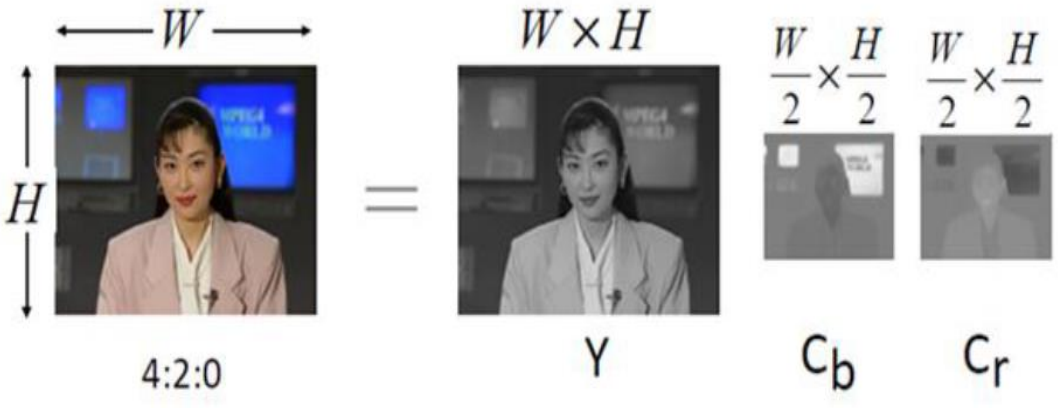


Figure 1-3: 4:2:0 sub-sampling pattern [7]

Representing video material in a digital form requires a large number of bits and the volume of data generated by digitizing a video signal occupies huge space for its storage and high bandwidth for its transmission which implies higher bit rates are required for uncompressed video. Hence compression of video plays a very crucial role. The compression must be such that it reduces the size of the original video without degrading its quality. The objective of any compression scheme is to represent the data in a compact form. Representation of data in reduced number of bits is achieved by removing redundant information from the video sequence.

Table 1-1: Compression Strategies [7]

Information Type	Compression Tool
Spatial Redundancy	Intra prediction
Perceptual Redundancy	Quantization
Statistical Redundancy	Entropy Coding
Temporal Redundancy	Inter prediction

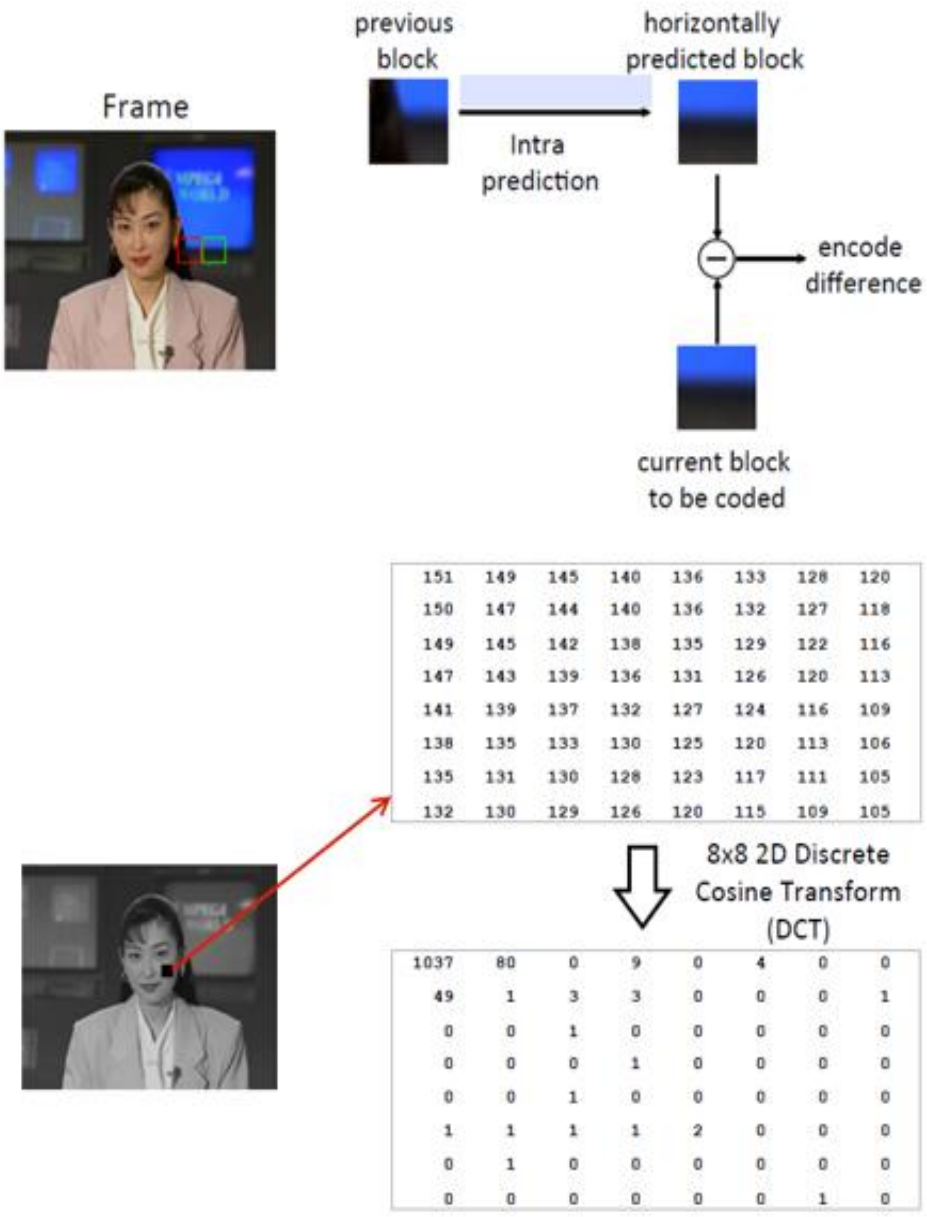


Figure 1-4: Spatial redundancy removal using intra prediction & block transforms [7]

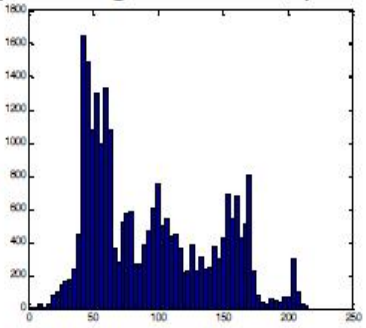
Original frame

Image obtained by retaining 36 DCT low frequency coefficients for each 8x8 block

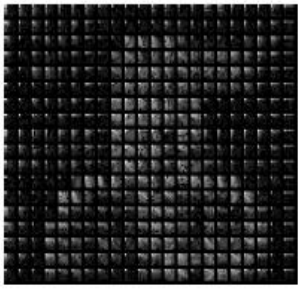
Figure 1-5: Perceptual redundancy removal using quantization [7]

- Original image: 8 bits/pixel, Entropy coding: 7.14 bits/pixel

Histogram



- Results more dramatic when entropy coding is applied on transformed and quantized image: 1.82 bits/pixel



Histogram

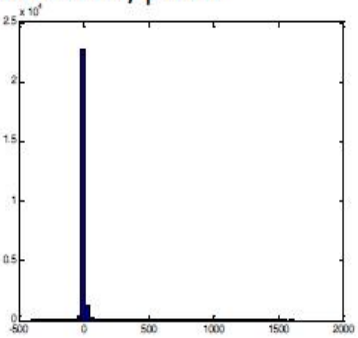


Figure 1-6: Statistical redundancy removal using entropy coding technique [7]

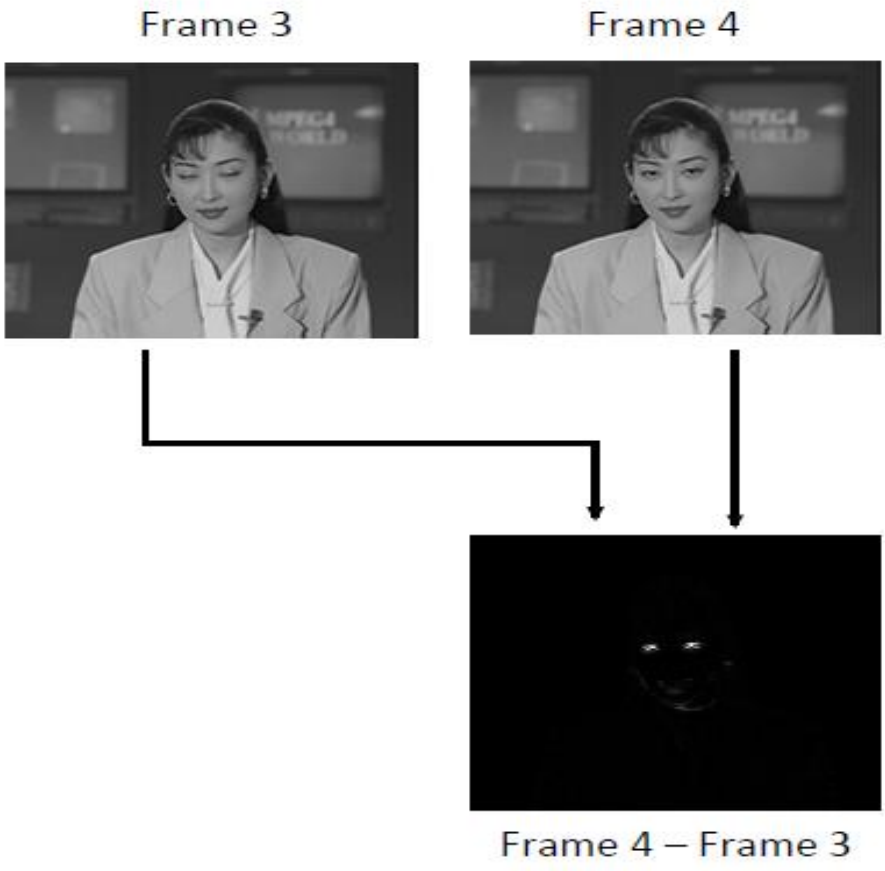


Figure 1-7: Temporal redundancy removal using frame difference [7]

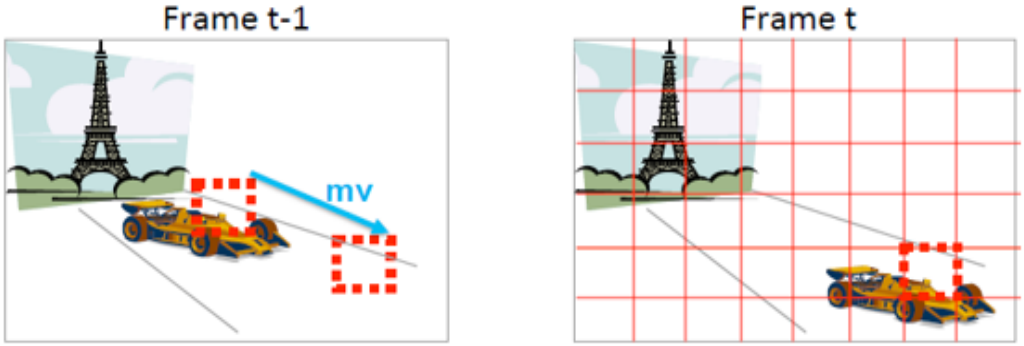


Figure 1-8: Temporal redundancy removal using motion compensated prediction [7]

A picture or frame will belong to one of the I-picture, P-picture or B-picture categories. I – pictures or intra predicted frame is the one in which picture or current frame is coded without reference to other pictures or other frames. P-pictures and B-pictures are said to be inter-coded using motion-compensated prediction from reference frames. P- pictures make use of a reference frame (the P-picture or I- picture preceding the current P -picture) i.e. picture is predicted from one prior coded picture whereas B-pictures make use two reference frames (the P- and/or I-pictures before and after the current frame) i.e. picture is coded from two prior coded pictures. The difference between predicted frame and actual frame carries less information and is coded to achieve compression. Intra picture (I) and Inter picture (P, B, b) are shown in Figure 1-1 [2] [7] [9].

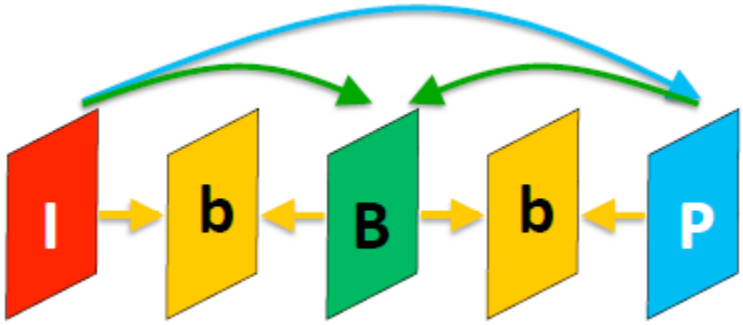


Figure 1-9: I-, P- and b- frames [7]

Is compression really necessary once transmission and storage capacities have increased to a sufficient level to cope with uncompressed video! It is true that both the storage and transmission capacities continue to increase. However, an efficient and well-designed video compression system gives very significant performance advantages for visual communication at both low and high transmission bandwidths [1].

1.3. Video Coding Standards

There have been several video coding standards introduced by organizations like the International Telecommunication Union - Telecommunication Standardization Sector (ITU-T), Moving Picture Experts Group (MPEG) and the Joint Collaborative Team on Video Coding (JCT-VC). Each standard is an improvement over the previous standard. Video compression standards ensure inter-operability between encoder and decoder. They usually support multiple use cases and applications by introducing different levels and profiles. With every standard, the general thumb of rule has been to retain the same video quality by being able to reduce the bit rate by 50%. Figure 1-10 shows the evolution of video coding standards over the years. [23]

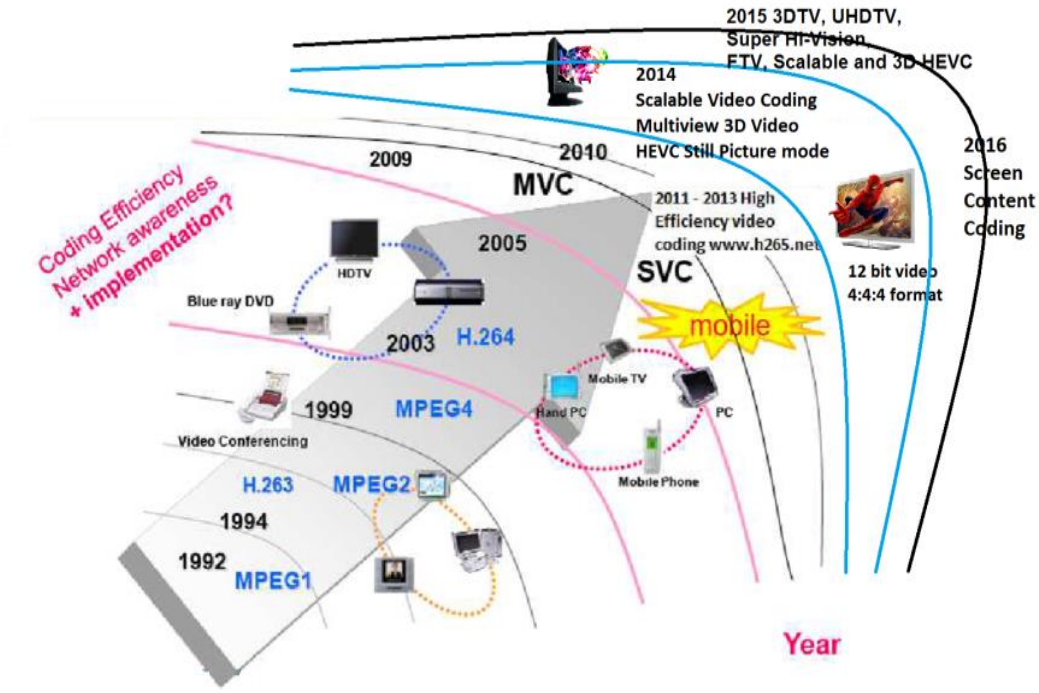


Figure 1-10: Evolution of video coding standards [23]

1.4. Thesis Outline

The thesis is organized as follows: Chapter 2 provides the overview of H.265 also known as High Efficiency Video Coding (HEVC) standard [2]. Chapter 3 presents the concepts of High Dynamic Range (HDR) and Wide Colour Gamut (WCG), Encoding and decoding chain process, highlights the different methods for HDR video compression, and discusses about the proposed scheme and evaluation methodology. The results of proposed scheme and its discussion are provided in Chapter 4.The conclusions and an insight into further work are given in Chapter 5.

Chapter 2

HIGH EFFICIENCY VIDEO CODING (HEVC)

2.1 Introduction

High Efficiency Video Coding (HEVC) [2] is an international standard for video compression developed by working group of ISO/IEC Moving Picture Experts Group (MPEG) and ITU-T Video Coding Experts Group (VCEG). Its predecessor H.264/ MPEG-4 AVC [5] was used in many application domains including broadcast of high definition (HD) TV signals over satellite, cable, and terrestrial transmission systems, video content acquisition and editing systems, camcorders, security applications etc., and it replaced all the other previous video compression standards[14]. 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) requires an efficient coding standard superior to H.264/MPEG-4 AVC [5] and current state-of-art video coding standard HEVC [1] addresses all these issues.

The main goal of HEVC is to significantly improve compression performance compared to existing standards such as H.264/MPEG-4 AVC [5] in the range of 50% bit rate reduction at similar visual quality [1] and to focus on two key issues : increased video resolution formats and increased use of parallel processing architectures. It primarily targets consumer applications as pixel formats are limited to 4:2:0 8-bit and 4:2:0 10-bit. Other goals of HEVC are to provide ease of transport system integration and data loss resilience.It supports resolutions up to 4K and 8K, bit depths of 8, 10, 12 and 16 bits per sample.

2.2 H.265 Encoder and Decoder

The video coding layer of HEVC employs the same hybrid approach (inter-/intra-picture prediction and 2-D transform coding) used in all video compression standards since H.261. A source video, consisting of a sequence of video frames, is encoded or compressed by a video encoder to create a compressed video bit stream. The compressed bit stream is stored or transmitted. A video decoder decompresses the bit stream to create a sequence of decoded frames [18].Figure 2-1 represents the encoder and decoder workflow. Figure 2-2 depicts the block diagram of a hybrid video encoder, which could create a bitstream conforming to the HEVC standard [13] whereas Figure 2-3 depicts the block diagram for HEVC decoder.

The video encoder performs the following steps:

- Partitioning each picture into multiple units
- Predicting each unit using inter or intra prediction, and subtracting the prediction from the unit
- Transforming and quantizing the residual (the difference between the original picture unit and the prediction)
- Entropy encoding transform output, prediction information, mode information and headers

The video decoder performs the following steps:

- Entropy decoding and extracting the elements of the coded sequence
- Rescaling and inverting the transform stage
- Predicting each unit and adding the prediction to the output of the inverse transform
- Reconstructing a decoded video image

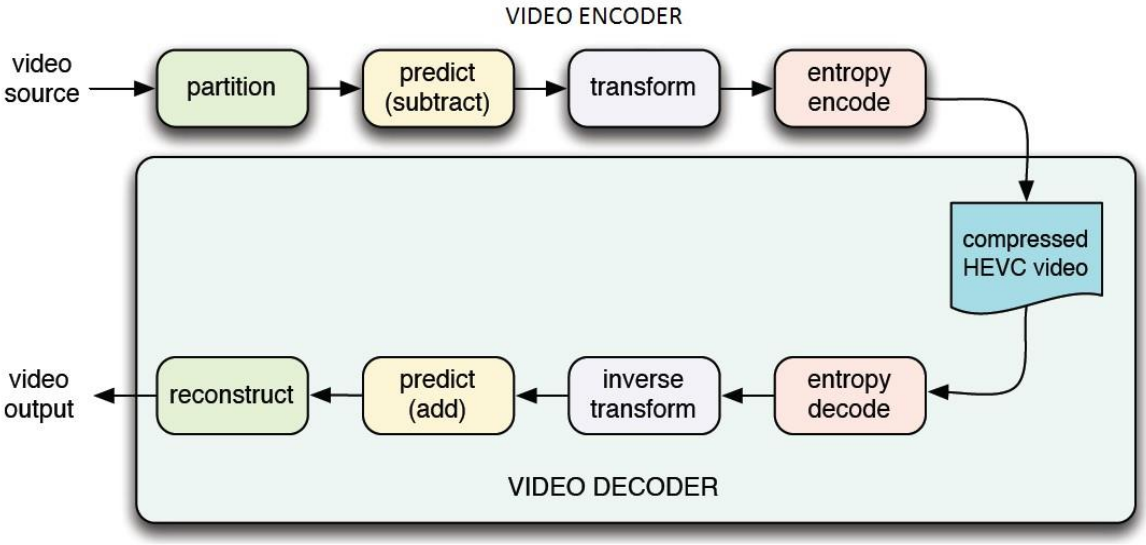


Figure 2-1: Block Diagram of HEVC CODEC [18]

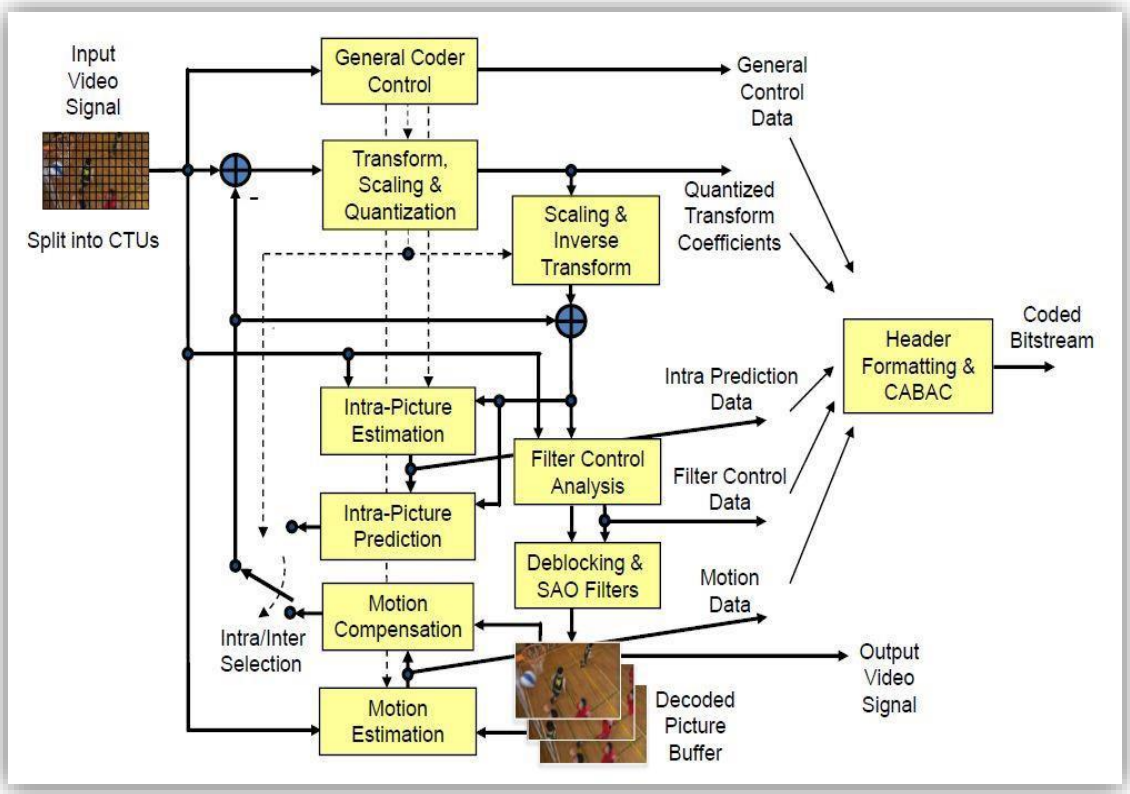


Figure 2-2: Block Diagram of HEVC video encoder [2]

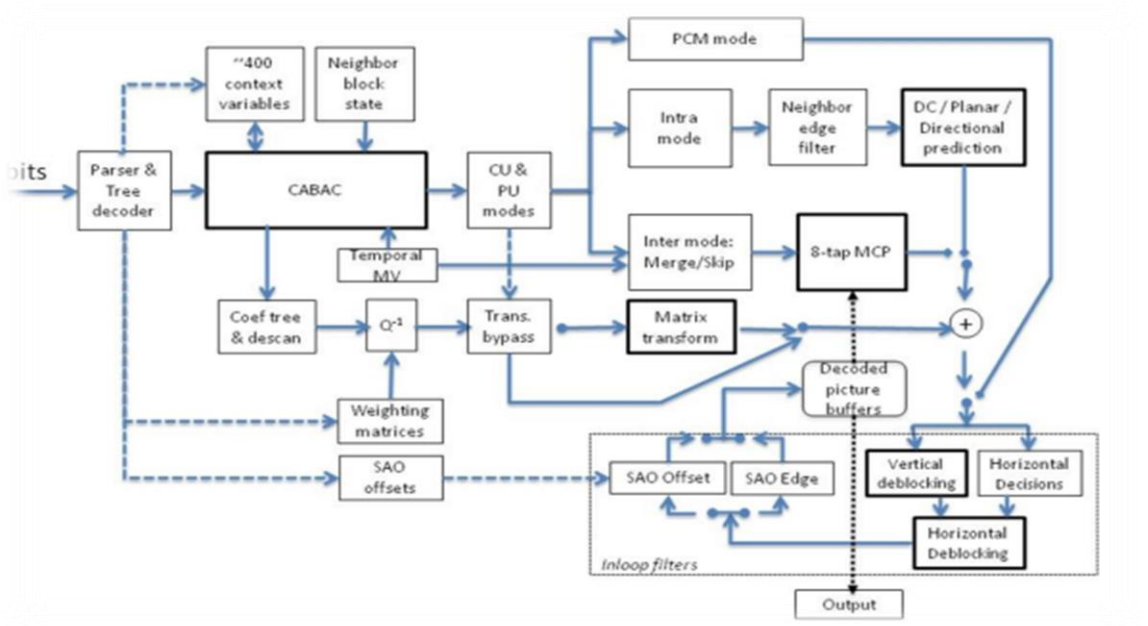


Figure 2-3: Block Diagram of HEVC Decoder [16]

2.3 Features of HEVC

2.3.1 Picture Partitioning

HEVC supports highly flexible partitioning of a video sequence. Each frame of the sequence is split up into square-shaped Coding Tree Blocks (CTBs), and its size varies from 16x16 to 64x64 pixels. One Luma CTB and two chroma CTBs with syntax elements form the Coding Tree Unit (CTU), which is the basic processing unit in HEVC. Each of the CTB is predicted from previously coded data. After prediction, any residual information is transformed and entropy encoded.

Each coded video frame, or picture, is partitioned into tiles and/or slices, which are further partitioned into coding tree units (CTUs) which are shown in Figure 2-4 and Figure 2-5. The CTU is the basic unit of coding, analogous to the macro-block in earlier standards, and can be up to 64x64 pixels in size.

A coding tree unit(CTU) can be subdivided into square regions known as coding units

(CUs) using a quad- tree structure. CU size ranges from 8x8 to 64x64 pixels. Each CU is partitioned into Prediction Units (PUs) which is predicted using inter or intra prediction and transformed using one or more transform units.

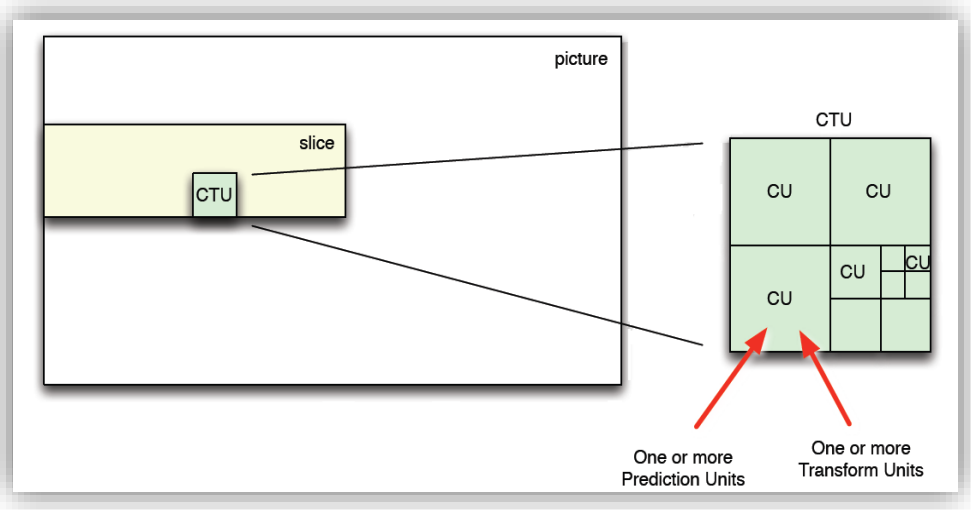
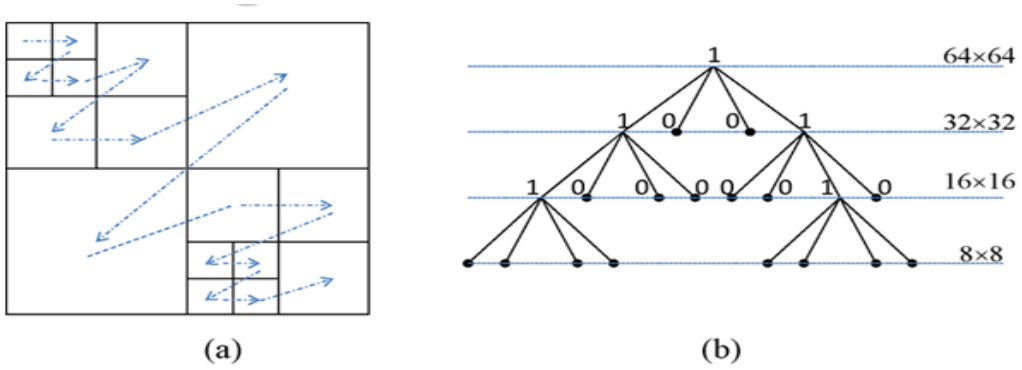


Figure 2-4: Picture, Slice, Coding tree Unit (CTU), Coding Unit (CU) [7]



Example of CTU, partitioning and processing order when size of CTU is equal 64 × 64 and minimum CU size is equal to 8 × 8 (a) CTU partitioning (b) Corresponding coding tree structure.

Figure 2-5: CTU partitioning and corresponding coding tree structure [8]

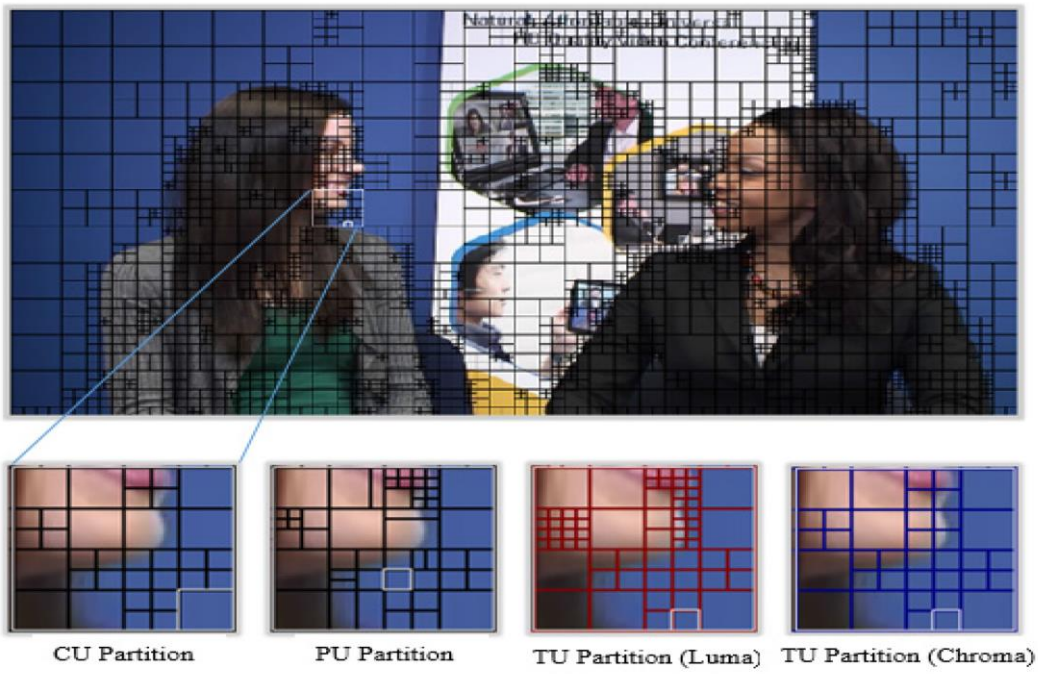


Figure 2-6: Partitioning of a frame in Kirsten and Sara Test Sequence [23] [28]

2.3.2 Prediction Schemes

Prediction in HEVC can be intra prediction or inter prediction. The decision whether to apply intra or inter prediction is made at the CU level. A sequence of CTBs is called a slice. CUs in intra mode are predicted from reconstructed neighboring samples within the same slice. In I slice, only intra prediction is enabled for the CUs. In P and B slices, CUs may be in both intra or inter prediction mode.

Intra prediction: In this prediction method, blocks are predicted using the neighboring pixels reconstructed from the same frame, exploring the spatial redundancy. Each PU is predicted from neighbouring image data in the same picture, using DC prediction (an average value for the PU), planar prediction (fitting a plane surface to the PU) or directional prediction (extrapolating from neighbouring data). HEVC has 35 intra prediction modes including a DC, planar and 33 angular prediction modes. The prediction

modes 2-18 are the horizontal prediction modes, 19-34 are the vertical prediction modes. Intra-coded CUs may only use the partition modes 2Nx2N or NxN, so intra PUs are always square.

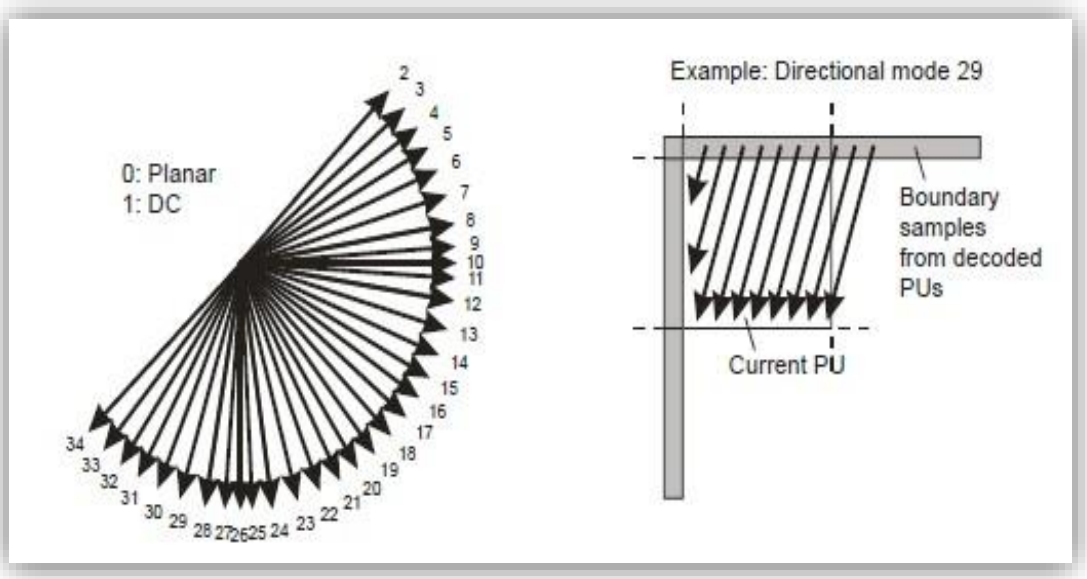
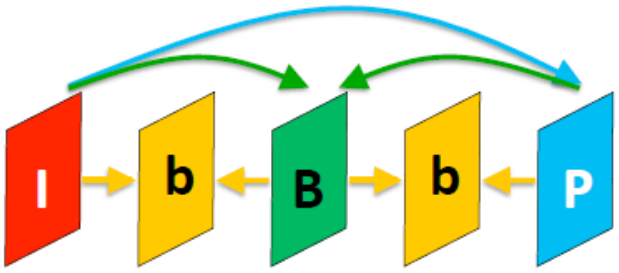


Figure 2-7: Modes and angular intra prediction directions in HEVC [2]

Inter prediction: This prediction method is performed at the prediction block (PB) level. It uses temporal redundancy between the adjacent frames in order to predict the current block of frame. Inter prediction is called the motion compensated prediction, since the shifted areas of the reference pictures are used for prediction of the current PB. Each PU is predicted from image data in one or two reference pictures (before or after the current picture in display order), using motion compensated prediction. For inter prediction, the partitions of CB into PB can be symmetric or asymmetric.

Motion compensated prediction can be performed using one or two reference pictures as the prediction source. The number of available prediction sources depends on the slice type to which the PB belongs. For P slices, uni-prediction can be used, which needs only a single prediction reference. For B slices, one or two prediction sources can

be applied and it uses two reference picture lists. Bi-prediction uses reference pictures from list 0 and list 1. Here uni-prediction or bi-prediction can be employed.



- Intra Picture (I)
  - Picture is coded without reference to other pictures
- Inter picture (P, B, b)
  - Uni-directionally predicted (P) Picture
    - Picture is predicted from one prior coded picture
  - Bi-directionally predicted (B, b) Picture
    - Picture is coded from two prior coded pictures

Figure 2-8: Picture Coding Types [7]

2.3.3 Motion Estimation and Motion Compensation [1]

Motion estimation (ME) refers to the process of determining motion vectors and representing the transition of objects in successive video frames. Motion estimation finds its applications in two main areas: reduction of temporal redundancy in video coders and representation of true motion of objects in real-time video applications. Motion estimation process in HEVC consumes more than 50% coding complexity or time to encode with equal perceptual quality.

Motion estimation creates a model of the current frame based on available data in one or more previously encoded frames (‘reference frames’). These reference frames may be ‘past’ frames or ‘future’ frames .The design goals for a motion estimation algorithm are to model the current frame as accurately as possible (since this gives better compression performance) whilst maintaining acceptable computational complexity. Figure 2-9 represents the concept of Motion estimation.

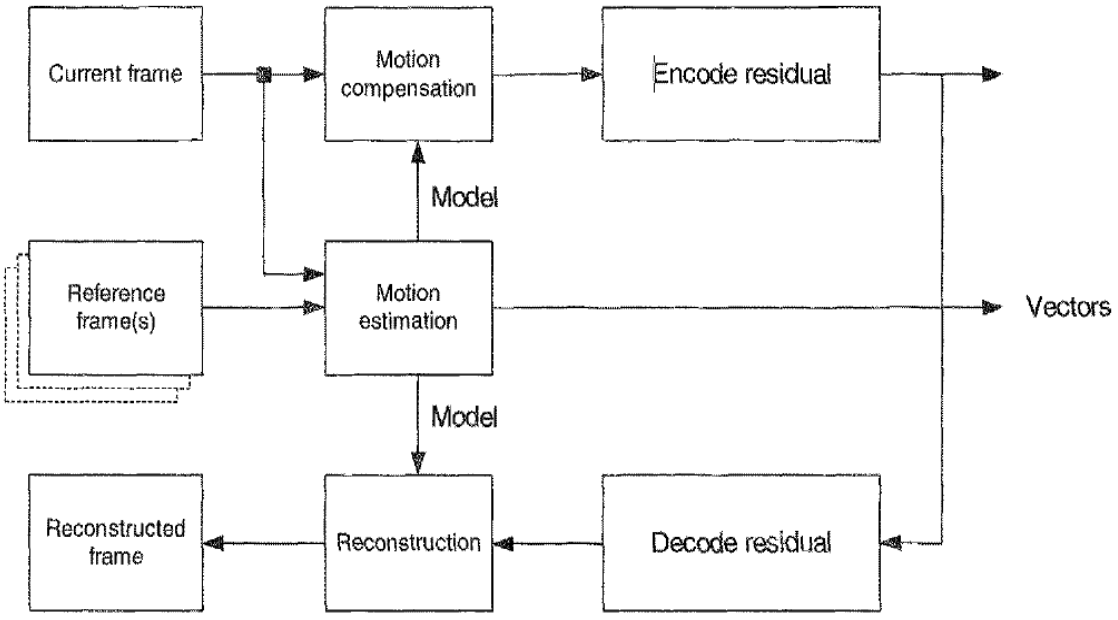


Figure 2-9: Motion estimation and Motion compensation block diagram [1]

Block based motion estimation (BBME) [63] is the most widely used motion estimation method for video coding. The underlying supposition behind motion estimation is that the patterns corresponding to objects and background in a frame of video sequence move within the frame to frame The idea behind block matching is to divide the current frame into a matrix of “macro blocks” that are then compared with corresponding block and its adjacent neighbors’ in the previous frame to create a vector that stipulates the movement of a macro block from one location to another in the previous frame. This movement calculated for all the macro blocks comprising a frame, constitutes the motion estimated in the current frame. The search area for a good macro block match is constrained up to “p” pixels on all fours sides of the corresponding macro block in previous frame. This “p” is called as the search parameter. Figure 11 represents the block based motion estimation process. Larger motions require a larger p and the larger the search parameter the more computationally expensive the process of motion estimation process. Usually the macro block is taken as a square of side 16 pixels, and the search parameter p is 7 pixels.

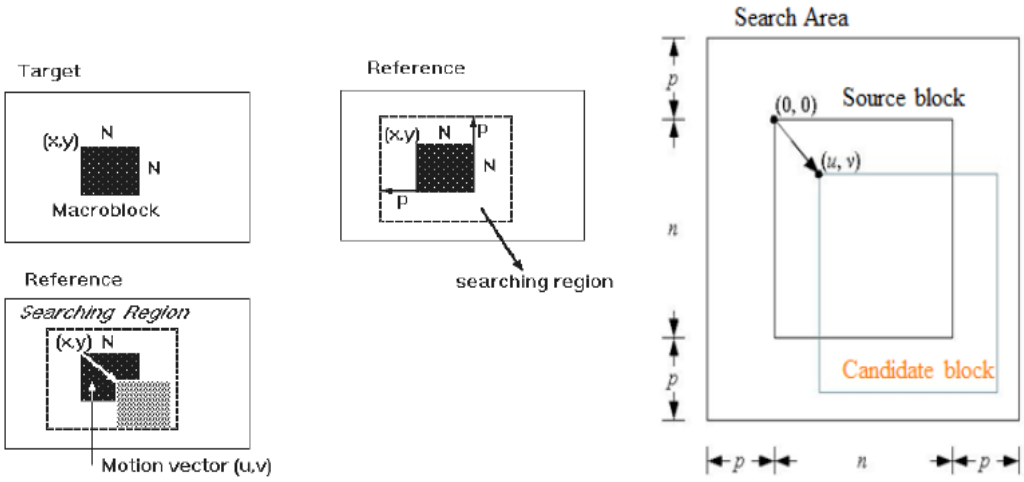


Figure 2-10 : Block based motion estimation process [29]

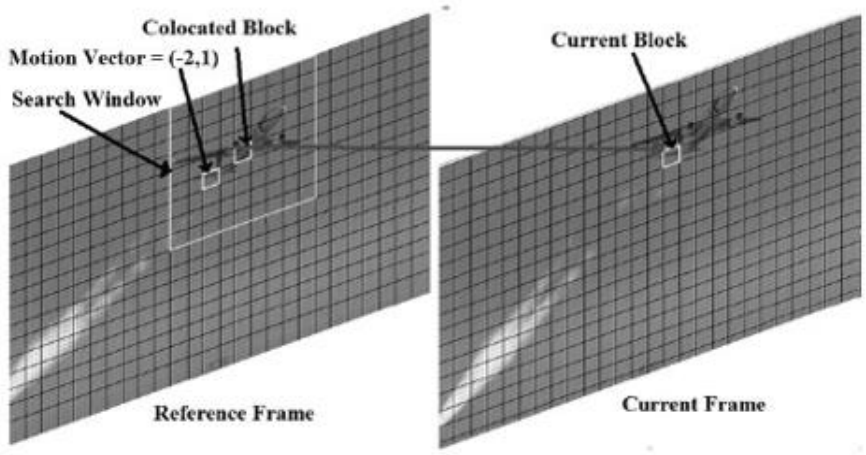


Figure 2-11: Illustration of motion estimation process [29]

The applicable motion vectors for motion compensation can be derived in two ways - motion vector predictor and motion vector difference. The motion vector predictor is selected from two candidates derived from the spatial and temporal neighborhood of the current block. This predictor selection method is called advanced motion vector prediction. Furthermore, the motion information such as motion vector and reference index can be derived by selection from a configurable set of candidates, without encoding a motion vector difference. The derivation of motion vectors from a candidate set is called merge mode [6].

When the motion vector does not have an integer value, fractional sample interpolation is used to generate the prediction samples for non-integer sampling positions. HEVC supports motion vectors with units of one quarter of the distance between luma samples and for chroma samples it is dependent on the chroma sampling format. For 4:2:0 sampling format (for every four luma samples, there will be one chroma sample obtained by horizontal and vertical sampling), one eighth of the distance between the chroma samples

is used. Fractional interpolation for luma samples uses separable application of an 8-tap filter for the half sample positions and a 7-tap filter for the quarter sample positions. For chroma components, 4-tap filter is used and the fractional accuracy is one eighth for the 4:2:0 chroma format. In Figure 2-12, the available luma samples at integer sample locations are labeled with upper-case letters, whereas the other positions labeled with lower-case letters represent samples at non-integer sample locations, which need to be generated by interpolation [2].

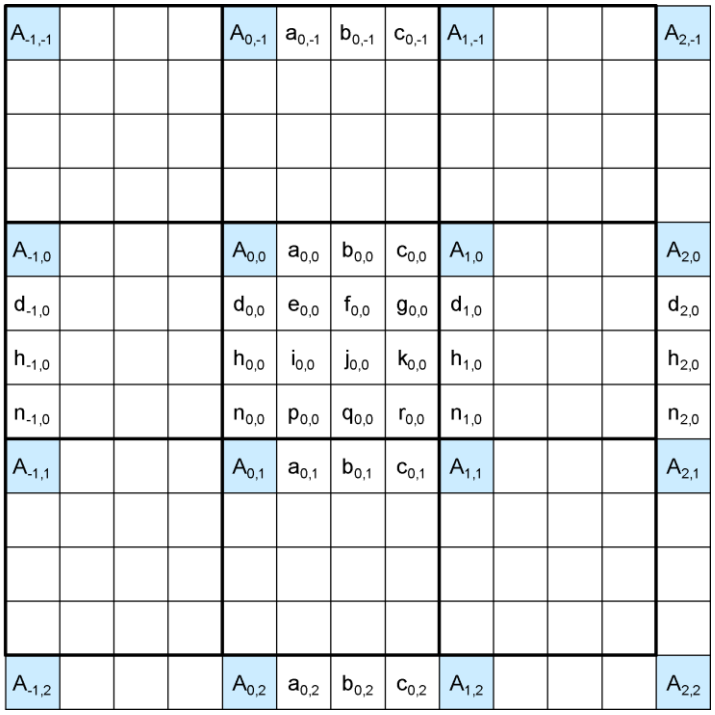


Figure 2-12: Integer and fractional sample positions for luma interpolation [2]

2.3.4 Transforms

The transform process in HEVC is specified using fixed point integer operations with output and intermediate values not exceeding 16-bit word length. Any residual data remaining after prediction is transformed using a block transform based on the Discrete Cosine Transform (DCT) or 4x4 intra Y Discrete Sine Transform (DST).

One or more block transforms of size 32x32, 16x16, 8x8 and 4x4 are applied to residual data in each CU. The transforms are separable, where two dimensional transforms are obtained by applying one-dimensional transforms in both the horizontal and vertical directions. Each transform block (TB) is divided into 4x4 sub-blocks (coefficient groups). The processing starts with the last significant coefficient and proceeds to the DC coefficient in the reverse scanning order. The higher precision and larger sizes of the transforms are one of the main reasons HEVC performs better than H.264/MPEG-4 AVC [6][8][23].

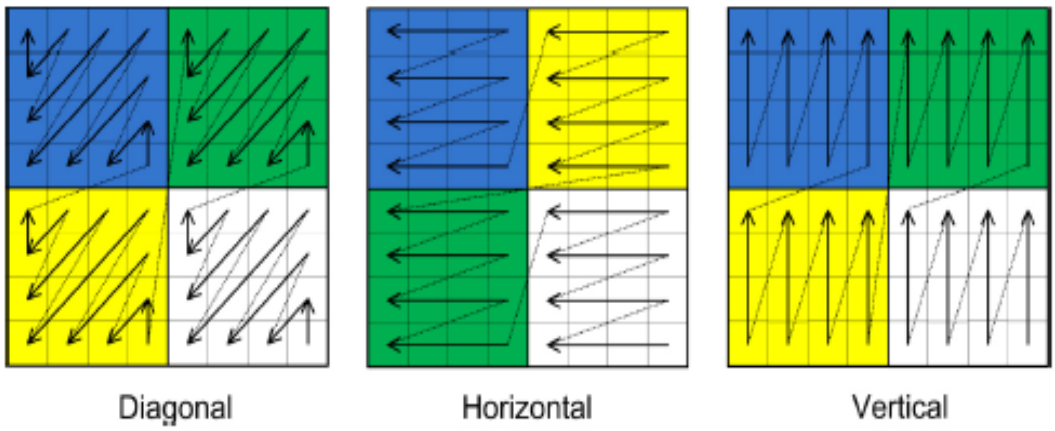


Figure 2-13: Coefficient scanning methods in HEVC [23] [30]

2.3.5 Quantization [23]

The transformed data is quantized [7]. For quantization, HEVC uses a scalar quantization such as uniform-reconstruction quantization (URQ) scheme controlled by a quantization parameter (QP). Quantization is the main source of loss of information in a lossy video compression such as HEVC. The range of QP values is defined from 0 to 51.

2.3.6 Entropy Coding [23]

A coded HEVC bit stream consists of quantized transform coefficients, prediction information such as prediction modes and motion vectors, partitioning information and

other header data. All of these elements are encoded using Context Adaptive Binary Arithmetic Coding (CABAC) [9].

CABAC is a method of arithmetic coding in which the probability models are updated based on the previous coding statistics. CABAC provides good compression performance through selecting probability models for each syntax element according to element's context, adapting probability estimates based on local statistics and using arithmetic coding.

Coding the data symbols involves the following steps [8]:

(1) Binarization - maps the syntax elements into binary symbols (0 or 1)

(2) Context modeling - estimates the probability of the bins

(3) Arithmetic coding - compresses bins to bits based on the estimated probability

2.3.7 In-loop filtering [2] [23]

HEVC uses two consecutive in-loop filters such as the de-blocking and the Sample Adaptive Offset (SAO) filters which are shown in Figure 2-1. A de-blocking filter similar to the one used in H.264/MPEG-4 AVC is operated within the inter-picture prediction loop. However, the design is simplified in regard to its decision-making and filtering processes, and is made more friendly to parallel processing[2]. It is applied to prediction block and transform block edges in order reduce the amount of visible block structures, which results from the block based nature of the coding scheme. This filter operates on the block edges with adaptive filter strength and adaptive filter length [6].

The Sample Adaptive Offset (SAO) is a sample based filtering operation which is operated on a CTU basis. The filter operates on the samples in the slice and does not only consider block edges. SAO is applied for the output of de-blocking filter and it reduces the ringing artifacts.

The DCT in HEVC works well on flat areas, but fails on areas with noise, contours and other peculiarities of the signal. It is efficient for large size of blocks but it is less efficient

for smaller sized blocks. The artifacts are more observable when the transform size increases. De-blocking can reduce artifacts on TB boundaries, while artifacts inside a TB can be reduced only by SAO. Hence, SAO can be applied when large transform sizes (32x32) are used [8].

2.4 Parallel Tools [2]

HEVC gives flexibility to use various in-built high level parallelization tools. This section describes the parallelization tools that can be used in HEVC such as slice, tile and wave-front parallel processing. [2] [23].

2.4.1 Slices

A slice is a partition of picture that can be decoded independently from other slices in the same picture. It can be the entire region of the picture or a portion of the picture. The minimum block structure that has to be present in a slice is a CTU. The main purpose of slice is to provide error resilience, parallel processing capability and resynchronization in the event of data losses.

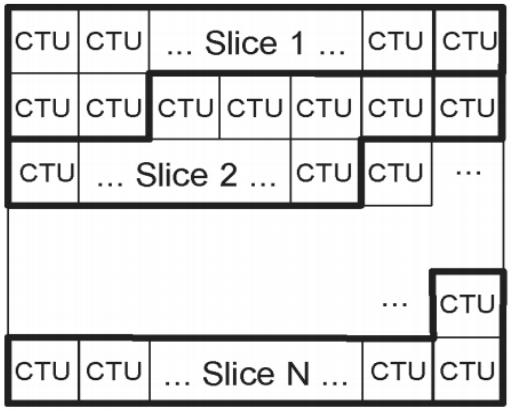


Figure 2-14 : Segmentation of Slice in HEVC [2]

2.4.2 Tiles

Tile is a feature in HEVC that divides the picture into rectangular shaped groups of CTUs separated by vertical and/or horizontal boundaries. Tiles break prediction and entropy coding dependencies across the tile boundaries, making them independent of each other. As the number of tiles increases, the coding efficiency reduces due to breaking of dependencies across boundaries and also resets the CABAC context at the beginning of each tile. Its main purpose is for parallel processing. Tiles can also be used for random access to local regions in video pictures.

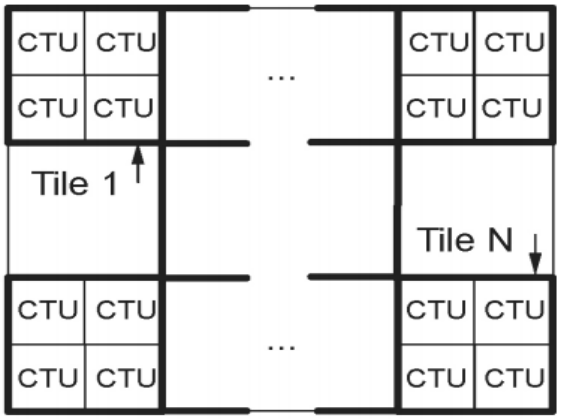


Figure 2-15: Subdivision of a picture into tiles [2]

2.4.3 Wavefront Parallel Processing (WPP)

Wave-Front Parallel Processing (WPP) is a parallelization technique, when used in HEVC it splits the picture into rows of CTUs, where each CTU can be processed by a different thread. The total number of threads that can be used when WPP is enabled depends on the number of CTU rows available, which in turn depends on ratio of picture height in luma samples and the luma CTB size. When WPP is used, each CTU row is processed relative to its previous row using a delay of two consecutive CTUs [2] [23].

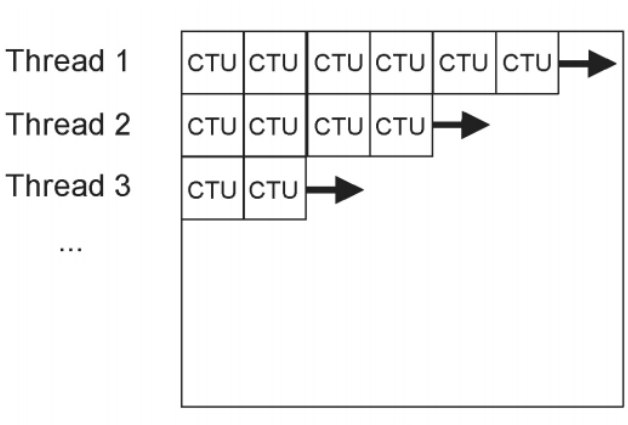


Figure 2-16: Illustration of Wavefront Parallel Processing [2]

2.5 Profiles, Levels and Tiers [2] [3] [31]

A profile defines the set of coding tools which can be used to encode a video sequence into a bit-stream. An encoder for a profile may choose which coding tools to use as long as it generates a conforming bit-stream while a decoder for a profile must support all coding tools that can be used in that profile. Version 1 of the HEVC standard defines three profiles: Main, Main 10, and Main Still picture. Minimizing the number of profiles provides a maximum amount of interoperability between devices, and is further justified by the fact that traditionally separate services, such as broadcast, mobile, streaming, are converging to the point where most devices should become usable to support all of them.

- *Main profile:* The Main profile allows for a bit depth of 8-bits per sample with 4:2:0 chroma sampling, which is the most common type of video used with consumer devices.
- *Main 10 profile:* The Main 10 profile allows for a bit depth of 8-bits to 10-bits per sample with 4:2:0 chroma sampling. HEVC decoders that conform to the Main 10 profile must be capable of decoding bit streams made with the following profiles: Main and Main 10. A higher bit depth allows for a greater number of colors. 8-bits

per sample allows for 256 shades per primary color while 10-bits per sample allows for 1024 shades per primary color. A higher bit depth allows for a smoother transition of color which resolves the problem known as color banding. The Main 10 profile allows for improved video quality since it can support video with a higher bit depth than what is supported by the Main profile. Additionally, in the Main 10 profile 8-bit video can be coded with a higher bit depth of 10-bits, which allows improved coding efficiency compared to the Main profile.

- *Main Still Picture profile:* The Main Still Picture profile allows for a single still picture to be encoded with the same constraints as the Main profile. As a subset of the Main profile the Main Still Picture profile allows for a bit depth of 8-bits per sample with 4:2:0 chroma sampling.

The levels indicate restrictions on parameters which determine decoding and buffering capabilities. These include the maximum picture size, the coded and decoded picture buffer sizes, the number of slice segments and tiles in a picture, as well as the maximum sample rate and maximum bitrate. It further sets a requirement for the minimum compression ratio which must be met by a bit-stream. There are 13 levels defined in HEVC standard.

Level	Max Luma Picture Size (samples)	Max Luma Sample Rate (samples/s)	Main Tier Max Bit Rate (1000 bits/s)	High Tier Max Bit Rate (1000 bits/s)	Min Comp. Ratio
1	36 864	552 960	128	–	2
2	122 880	3 686 400	1500	–	2
2.1	245 760	7 372 800	3000	–	2
3	552 960	16 588 800	6000	–	2
3.1	983 040	33 177 600	10 000	–	2
4	2 228 224	66 846 720	12 000	30 000	4
4.1	2 228 224	133 693 440	20 000	50 000	4
5	8 912 896	267 386 880	25 000	100 000	6
5.1	8 912 896	534 773 760	40 000	160 000	8
5.2	8 912 896	1 069 547 520	60 000	240 000	8
6	35 651 584	1 069 547 520	60 000	240 000	8
6.1	35 651 584	2 139 095 040	120 000	480 000	8
6.2	35 651 584	4 278 190 080	240 000	800 000	6

Figure 2-17: 13 levels in HEVC

The concept of tiers enables the differentiation between different application types which require different available bitrate ranges. Correspondingly, the maximum bitrate and the maximum CPB size differ between tiers. Two tiers of levels are specified in HEVC. The Main tier targets consumer applications, the High tier is designed for professional applications.

2.6 High-Level Syntax Architecture [2] [3]

An HEVC bit-stream consists of a sequence of data units called network abstraction layer (NAL) units. There are two classes of NAL units in HEVC - video coding layer (VCL) NAL units and non-VCL NAL units. Each VCL NAL unit carries one slice segment of coded picture data while the non-VCL NAL units contain control information that typically relates to multiple coded pictures. There are 64 different NAL unit types. One coded picture, together with the non-VCL NAL units that are associated with the coded picture, is called an HEVC access unit.

The syntax elements that describe the structure of the bit-stream or provide information that applies to multiple pictures or to multiple coded block regions within a picture, such as the parameter sets, reference picture management syntax, and Supplemental

Enhancement Information (SEI) messages, are known as the “high level syntax” part of HEVC [3][6].

HEVC specifies three parameter sets. These are the video, sequence, and picture parameter set (VPS, SPS, and PPS). VPS collectively provides information on different layers and sub-layers of the coded video sequence. SPS contains information which applies to all slices of a video sequence.PPS conveys information which could change from picture to picture such as QP, tiles etc. Slice header conveys information that can change from slice to slice such as Picture Order Count (POC), slice type, prediction weights, de-blocking parameters and tiles entry points.

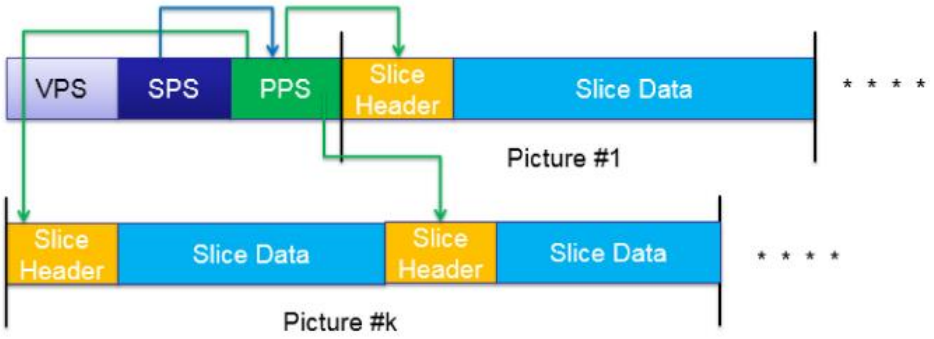


Figure 2-18: High Level Syntax of HEVC [8]

2.7 Summary

Chapter 2 describes overview of HEVC while focusing on encoder and decoder workflow, different features of HEVC, Parallel tools, different profiles, levels and tiers, High-Level Syntax Architecture. Chapter 3 will describe High Dynamic Range and Wide Colour Gamut, Definitions, Encoding and Decoding chain process, Colour Transformation, discuss various methods for HDR video compression and propose a scheme that will be used in this research.

Chapter 3

HIGH DYNAMIC RANGE & WIDE COLOR GAMUT

3.1 Introduction [32]

Currently, TVs providing Standard Dynamic Range (SDR) typically support brightness in the range of the order of 0.1 to 100 nits [62]. However, that range is significantly smaller than the range encountered in real life. For example, a light bulb can have much more than 10,000 nits, surfaces lit in the sunlight can have brightness upwards of 100s of thousands of nits, while the night sky can be as low as 0.005 nits (or lower).One of the key aspects of Ultra High Definition TVs (UHDTVs) is to provide the user a sense of "being there" and "realness". This requires creating, capturing and displaying the content that has much higher peak brightness and much larger contrast values than today's TVs. Simply increasing the resolution is not significant enough to attain this goal.

In addition, this also requires providing colors that are significantly richer than the ones provided by today's standard, e.g. BT.709 [36]. Thus, the new content will not only have several orders of magnitude larger brightness and contrast, but also significantly wider color gamut (for example, BT.2020 [37] or even wider than that in the future).The content may consist of computer-generated animation and camera captured video or images. It may be distributed to the consumer electronically (i.e. via a network connection) or physically (on optical media, flash memory, magnetic storage, etc.).

As both the dynamic ranges and the volumes of the color gamut associated with various capturing and displaying devices are expected to increase significantly, it is not clear at this stage if existing MPEG video coding standards are able to efficiently support the needs of future TV as well as other content distribution environments. The need of future high quality content distribution systems, including future 4k UHDTV, that support content with much higher dynamic ranges and wider color gamut will be discussed.

3.2 Definitions

3.2.1. Higher dynamic Range (HDR) [32]

Overall, the dynamic range of a scene can be described as the ratio of the maximum light intensity to the minimum light intensity. In digital cameras, the most commonly used unit for measuring dynamic range is in terms of f-stops, which describes total light range by powers of 2. Here each stop corresponds to a doubling of light intensity. The current ad hoc use of the term f-stop, refers to the following dynamic ranges:

10 f-stops = a difference of 2^{10} = 1024: 1 contrast ratio.

14 f-stops = a difference of 2^{14} = 16,384: 1 contrast ratio.

16 f-stops = a difference of 2^{16} = 65,536: 1 contrast ratio. It is normally regarded as 100,000:1; approximately what the eye can see in a scene with no adaptation.

20 f-stops = a difference of 2^{20} = 1,048,576: 1 contrast ratio. It is normally regarded as 1,000,000:1; approximately what the eye can see in a scene with minimal (no noticeable) adaptation. In the ad hoc categorization of the dynamic ranges, the following definitions are typical:

Standard Dynamic Range (SDR) is ≤ 10 f-stops

Enhanced Dynamic Range (EDR) is > 10 f-stops and ≤ 16 f-stops

High Dynamic Range is (HDR) > 16 f-stops

HDR is often defined as any signal or device that has a dynamic range greater than Standard Dynamic Range (SDR) and, thus, better approximating the capabilities of human vision. Notice that this does not correspond to a defined maximum and minimum luminance. For example, in the MPEG requirements for HDR and WCG content distribution and storage [32], HDR signals are characterized as having more than 16 f-stops of range. SDR, as used in current HDTV standards and video distribution environments, is defined as a luminance range of approximately 0.1 to a few hundred nits, or 10-11 f- stops, and an

“Intermediate” or “Enhanced” dynamic range (EDR) signal is considered to have a dynamic range of between 10 and 16 f- stops. Notice that SDR is significantly smaller than the range encountered in real life and far from the capabilities of human vision system. Again though, in practice, the definition of HDR is often any dynamic range that is greater than SDR.

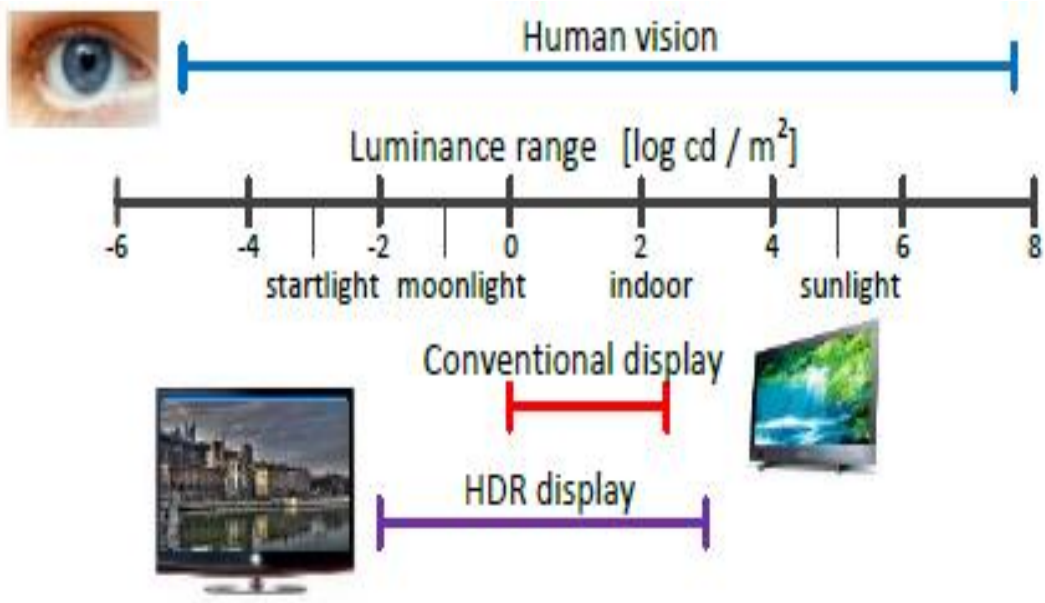


Figure 3-1: Dynamic Range of human vision compared to display capabilities [34]

3.2.2. Wide Color Gamut (WCG) [34]

Wide color gamut is the second dimension for providing a more realistic viewing experience, and it expresses an extension of the range of colors that are rendered or sensed. This range is often defined as a color gamut, and Figure 3-2 illustrates the color gamut used in HDTV standards in the xy chromacity diagram.

Historically, the defined color gamut for TV delivered content in Standard Definition (SD) is defined in ITU-R BT.601 and High Definition (HD) is defined in ITU-R BT.709 [36]. This includes all content viewed by the consumer on a TV, such as cable, broadcast and Blu-ray.

While the color gamut for BT.709 [36] may look fairly impressive, it was originally designed to work on Cathode Ray Tube (CRT) monitors. This type of monitor has long been replaced by new technologies, which are capable of displaying a greater number of colors. Examples of this technology include, LED, OLED and quantum dot technologies. With the proliferation of new display technologies and Ultra High Definition Television (UHDTV), the industry recognized the need to include colors beyond those available in BT.709 [36]. Color gamut larger than BT.709 [36] is referred as Wide Color Gamut (WCG). Examples of wide color gamut include (but, are not limited to) ITU-R BT.2020 [37] and Digital Cinema Initiatives (DCI).

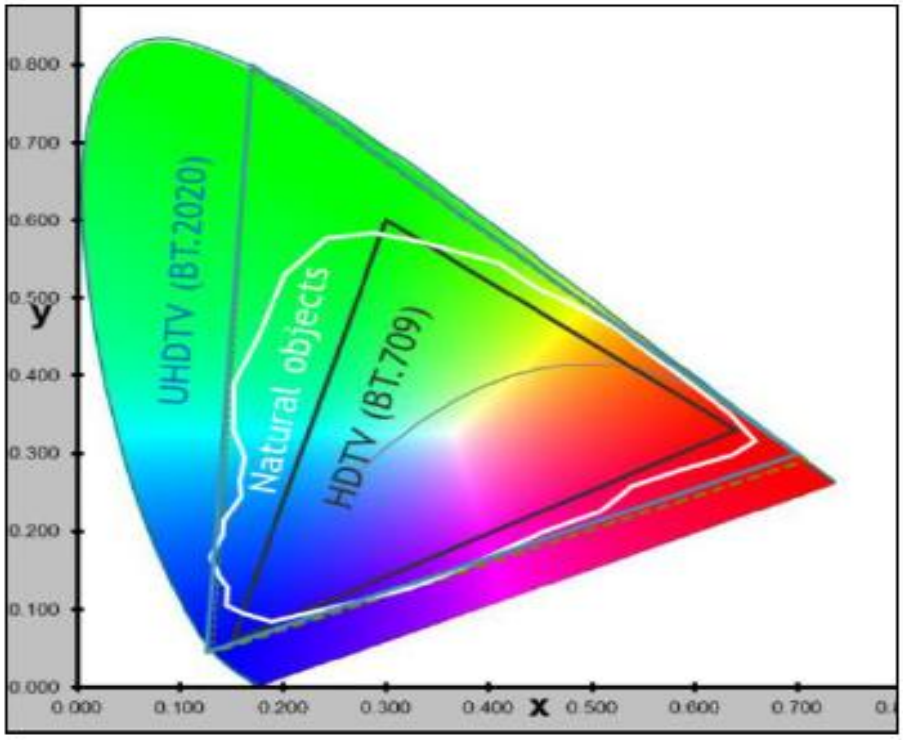


Figure 3-2: Pointer's gamut contour compared to different gamut [40]

This corresponds to the familiar triangle prescribed by the BT.709 color primaries [36]. UHDTV systems anticipate a wider color gamut, and this gamut is also shown in the figure and corresponds to the color primaries defined in BT.2020 [37]. To provide more

intuition about the meaning of the color gamuts, Figure 3-2 also depicts the spectrum locus curve that delimits the tongue- or horseshoe-shaped area. The spectrum locus is the boundary containing all monochromatic colors that can be seen by the human eye. Additionally, the figure contains Pointer's gamut that denotes all real surface colors that an average human visual system is able to perceive [38]. As illustrated by this figure, BT.2020 [37] covers a much wider area of the color space than BT.709 [36]. Specifically, in the xy chromacity diagram, BT.709 [36] covers 33.5% of the chromaticities, while BT.2020 [37] covers 63.3% of the chromaticities [38].

Finally, HDR and WCG are two independent attributes of the video content. A video signal may possess high dynamic range but still have BT.709 [36] color gamut or may have wider color gamut than BT.709 [36] but still have standard dynamic range.

3.2.3. Scene Referred and Display Referred pictures [32]

Scene referred pictures relate to the real luminance captured from the scene. It corresponds to how an image appeared when it was originally captured. Scene Referred values are linearly related to the amount of light in the depicted scene. In a Scene Referred pipeline the processed image is not directly viewable.

Display Referred corresponds to how an image gets rendered on a specific display. The pixel sample values in the captured scene or associated graded sequence may get modified to match the capabilities of the display. For example, if a scene is captured or represented in BT.2020 [37] color space with colors out of the display gamut then at least the out-of-gamut colors get modified to match the display capabilities. Similarly, the luminance/luma values in the captured or graded scene may get modified to match the dynamic range and the peak luminance of the display.

3.3. Video Level Considerations and Requirements

3.3.1. Higher Dynamic Range [32]

The dynamic ranges of the content and the displays should be decoupled. The achievable and desired brightness and the dynamic ranges of various displays may be significantly different than those on the capturing and creating ends. For example, a content creation system may be able to create or capture content with contrast of 1,000,000:1 but it may be neither desirable nor feasible to have displays with that range. So, there may be a need to do some display dependent mapping of the content's dynamic range. That mapping may be done at the encoding end or the receiving end. This may also be a function of the distribution mechanism, e.g. point-to-point communication or broadcasting. Therefore, it is important that the standard be largely feature proof in terms of maximum brightness and contrast range and it should be developed with flexibility in their values.

3.3.2. Content, Input Types and Bit Depths [32]

All three types of content are supported: Computer generated (animation), Camera captured video and images. The standard shall support integer (8, 10, 12 and 16 bits) and half-floating point (IEEE 754) [56] input video data formats. One or several compression internal image integer formats can be defined. As the standard is likely to operate internally using fixed point arithmetic, mechanisms should be provided that would allow an encoder to map floating point formats to the appropriate integer formats that the encoder considers to be the most efficient. A mechanism to indicate the mapping used to create the input integer values provided to an encoder should be provided. Also the mechanism should be provided that would allow a receiver to map the decoded video format to the one needed for display.

3.3.3. Transfer Function [34]

Input video to the encoder may or may not have a coding Transfer Function (TF) applied to it. This applies to both integer and floating point representations. Multiple coding TFs should also be supported. Mechanism to map the input coding TF (linear or non-linear) to another function, which may be more efficient from a coding efficiency point of view or appropriate from display point of view, should also be supported. The receiver may apply appropriate inverse coding TF for the signal to be displayed. Luma and Chroma may have different TF s.

Transfer Function establishes the video signal dynamic range by specifying a mapping between an integer sample level between 0 and $2^{BitDepth}$ -1 to a display primary intensity in nits. Most transfer functions are designed to compand a linear signal requiring a large number of uniform intensity steps into a more compact, non-linear signal with fewer, more visually uniform steps. For example, the inverse gamma transfer function in BT.709 expands the non-linear 8-bit signal to approximately 11-bits of a linear light in the range 0.1 to 100 nits.

The new Perceptual Quantizer (PQ) transfer function [39] is more adapted to HDR than the usual gamma. PQ is designed toward the human contrast sensitivity model developed by Barten. The PQ transfer function (officially defined in the SMPTE specification ST 2084 [40], is illustrated in Figure 3-3 . The figure relates the log10 intensity levels (vertical axis), in nits, to code levels of integer video samples (horizontal axis). For comparison, the traditional 8-bit SDR (green) curve is plotted next to the 10-bit PQ (blue) curve. Figure 3-3 conveys a key tradeoff achieved with the two extra bits that 10-bit HDR provides over 8-bit SDR: the SDR intensity range 0.1 to 100 nits [53] , within the overall HDR intensity range 0.00005 to 10000 nits, is effectively assigned one of those extra two bits.

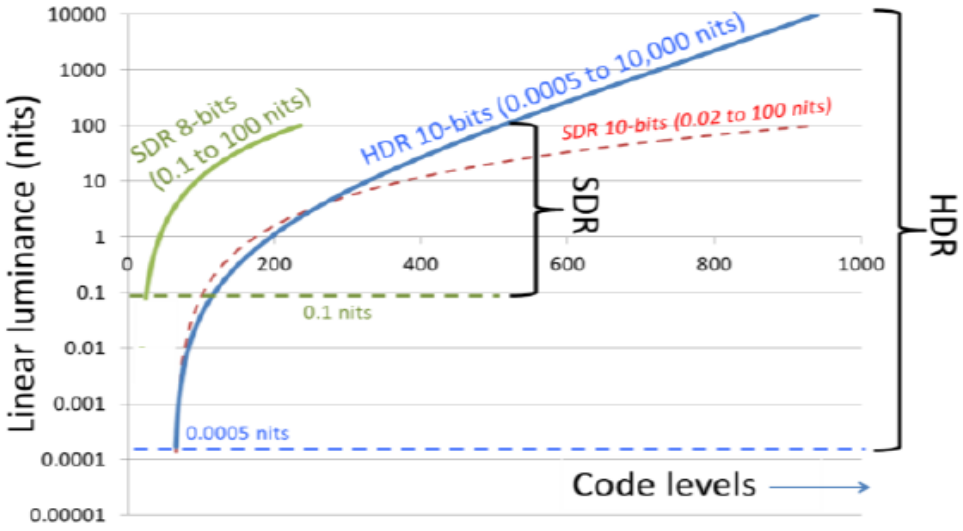


Figure 3-3: Mapping Linear light values for SDR and HDR representations [34]

3.3.4. Color Sampling and Formats

The standard should support 4:2:0, 4:2:2 and 4:4:4 chroma formats are shown in Figure 1-2. Chroma sub-sampling and the domain in which it gets performed may have a visual quality and coding efficiency impact and should be studied.

3.3.5. Color Spaces [52]

The standard should support multiple color spaces. Higher Dynamic Range (EDR and HDR) can be combined with any of the color spaces, for example, BT. 709 [36], BT. 2020 [37] or XYZ or others. The standard should allow signaling of meta-data to assist in the conversion of the bit-stream color space to a display color space. Impact of compression noise and interaction of the compression noise with those mapping schemes should be studied. More generally, the impact of potential interactions between the transfer functions, color space conversions, chroma sub/up-sampling and compression should be studied.

3.4. Encoding and Decoding Chain Workflow

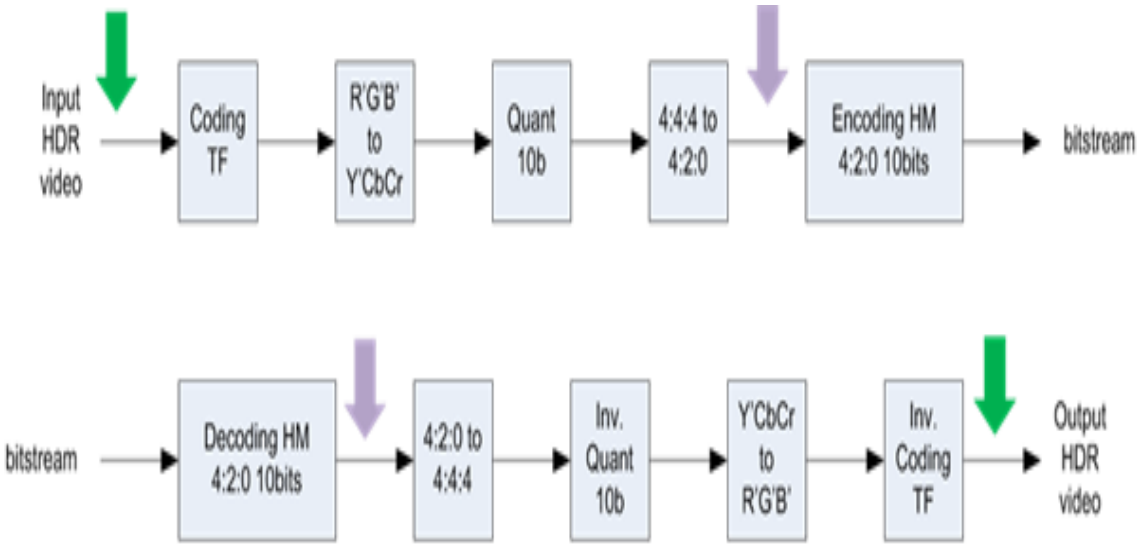


Figure 3-4: End to end coding and decoding chain [52]

3.4.1. Encoding Process [52]

If the input is in a half float 4:4:4 RGB linear-light format (e.g. OpenEXR), the bit streams shall be generated using the coding / decoding chain illustrated in Figure 3-4.

The conversion to 4:2:0 10 bits $Y'C_bC_r$ is obtained with the HDRConvert tool using one of the configuration files provided in the directory 'bin\CfE_cfgFiles' of the HDRTools package, namely *HDRConvertEXR709ToYC$_b$C$_r$420.cfg* (or) *EXR2020ToYC$_b$C$_r$420.cfg*, depending on the content container primaries (BT.709 [36] or BT.2020 [37]).

It consists of the following steps:

- Map using the PQ transfer function (PQ-TF) from RGB (float) to R'G'B' (float)
- Convert from R'G'B' (float) to $Y'C_bC_r$
- Quantize from $Y'C_bC_r$ (float) into $D_{Y'}D_{Cb}D_{Cr}$ (10bit)
- Downsample both chroma components from 4:4:4 $D_{Y'}D_{Cb}D_{Cr}$ (10bit) to 4:2:0 $D_{Y'}D_{Cb}D_{Cr}$ (10bit)

3.4.2. Decoding Process [52]

The reverse conversion is obtained with the HDRConvert tool using one of the configuration files provided in the directory 'bin\CfE_cfgFiles' of the HDRTools package, namely *HDRConvertYC$_b$C$_r$420ToEXR709.cfg* (or) *YC$_b$C$_r$420ToEXR2020.cfg*, depending on the content container primaries (BT.709 or BT.2020).

The process consists of the following steps:

- Upsample both chroma components from 4:2:0 $D_{Y'}D_{Cb}D_{Cr}$ (10bit) to 4:4:4 $D_{Y'}D_{Cb}D_{Cr}$ (10bit)
- Inverse quantize from $D_{Y'}D_{Cb}D_{Cr}$ (10bit) into Y'CbCr (float)
- Convert from Y'C$_b$C$_r$ (float) to R'G'B' (float)
- Inverse map using the inverse PQ-TF from R'G'B' (float) to RGB (float)

3.5. Color Transformation Process [52]

3.5.1. Conversion from RGB to R'G'B'

R' = PQ_TF (max (0, min(R/10000, 1)))

G' = PQ_TF (max (0, min (G/10000, 1)))

B' = PQ_TF (max (0, min (B/10000, 1)))

with $PQ\_TF(L) = \left(\frac{c_1 + c_2 L^{m_1}}{1 + c_3 L^{m_1}}\right)^{m_2}$

m_1 = 0.1593017578125

m_2 = 78.84375

$c_1 = c_3 - c_2 + 1$ =0.8359375

c_2 =18.8515625

c_3 =18.6875

3.5.2. *R'G'B' with BT.709 to Y'C_bC_r / R'G'B' with BT.2020 to Y'C_bC_r*

The ITU-R BT.709 standard specifies the following conversion process from R'G'B' to Y'CbCr (non-constant luminance representation):

Y' = 0.2126 * R' + 0.7152 * G' + 0.0722 * B'

C_b = - 0.114572 * R' - 0.385428 * G' + 0.500000 * B'

C_r = 0.500000 * R' - 0.454153 * G' - 0.045847 * B'

The ITU-R BT.2020 standard specifies the following conversion process from R'G'B' to Y'CbCr (non-constant luminance representation):

Y' = 0.262700 * R' + 0.678000 * G' + 0.059300 * B'

C_b = -0.139630 * R' - 0.360370 * G' + 0.500000 * B'

C_r = 0.500000 * R' - 0.459786 * G' - 0.040214 * B'

3.5.3. *Quantization from Y'C_bC_r into $D_YD_{Cb}D_{Cr}$*

This process quantizes the input Y'C_bC_r signal into a signal of bit-depth BitDepthY for the Y component and BitDepthC for the chroma components (C_b, C_r).

$$D_{Y'} = \mathrm{Clip1}_Y\left(\mathrm{Round}\left(\left(1 \ll (\mathrm{BitDepth}_Y - 8)\right) * (219 * Y' + 16)\right)\right)$$

$$D_{Cb} = \mathrm{Clip1}_C\left(\mathrm{Round}\left(\left(1 \ll (\mathrm{BitDepth}_C - 8)\right) * (224 * Cb + 128)\right)\right)$$

$$D_{Cr} = \mathrm{Clip1}_C\left(\mathrm{Round}\left(\left(1 \ll (\mathrm{BitDepth}_C - 8)\right) * (224 * Cr + 128)\right)\right)$$

with

Round(x) = Sign(x) * Floor (Abs(x) + 0.5)

Sign (x) = -1 if x < 0, 0 if x=0, 1 if x > 0

Floor(x) the largest integer less than or equal to x

Abs(x) = x if x>=0, -x if x<0

$\mathrm{Clip1}_Y$(x) = Clip3 (0, (1 << BitDepthY) − 1, x)

$Clip1_C(x) = Clip3\ (0, (1 << BitDepthC) - 1, x)$

Clip3(x, y, z) = x if z<x, y if z>y, z otherwise

3.5.4. Chroma downsampling from 4:4:4 to 4:2:0

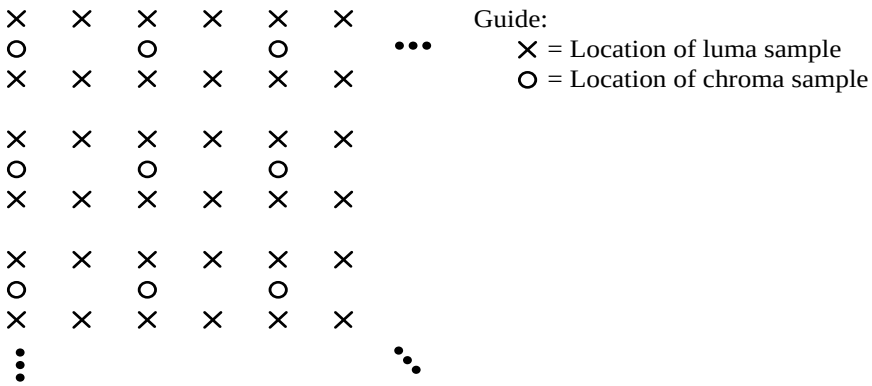


Figure 3-5 : Chroma samples alignment

Table 3-1: Chroma samples alignment

Phase k	Coef's c1[k] 4:4:4 → 4:2:2 (phase=0)	Coef's c2[k] 4:2:2 → 4:2:0 (phase=0.5)
-1	1	0
0	6	4
1	1	4

- Define shift = 6 and offset = 32.
- Let H and W be the input picture height and width respectively in chroma samples. For i = 0..H-1, j = 0..W/2-1, the intermediate samples f[i][j] are derived from the input samples s[i][j] as follows:

$$f[i][j] = \sum_{k=-1}^{1} c1[k] * s[i][\mathrm{Clip3}(0, W-1, 2*j+k)]$$

with Clip3(x, y, z) = x if z<x, y if z>y, z otherwise

- For i = 0..H/2-1, j = 0..W/2-1, the output samples r[i][j] are derived from the intermediate samples f[i][j] as follows:

$$r[i][j] = (\sum_{k=-1}^{1} c2[k] * f[\text{Clip3}(0, H-1, 2*i+k)][j] + offset) \gg shift$$

3.5.5. *Chroma Upsampling from 4:2:0 to 4:4:4 (Y'C_bC_r domain)*

The upsampling filter used is the same for both horizontal and vertical processes. First, vertical filtering is applied on the 4:2:0 picture, then horizontal filtering.

Filter coefficients values are as follows:

Table 3-2: Upsampling Filter Coefficient's values

Phase	-2	-1	0	1
Coef c[k]	-4	36	36	-4
Coef d0[k]	-2	16	54	-4
Coef d1[k]	-4	54	16	-2

- Define shift1 = 6, offset1 = 32, shift2 = 12, offset2 = 2048.
- Let H and W be the input picture height and width in chroma samples. For i = 0..H-1, j = 0..W-1, the intermediate samples f[i][j] are derived from the input samples s[i][j] as follows:

$$f[2*i][j] = \sum_{k=-2}^{1} d0[k] * s[\text{Clip3}(0, H-1, i+k)][j]$$

$$f[2*i+1][j] = \sum_{k=-2}^{1} d1[k] * s[\text{Clip3}(0, H-1, i+k+1)][j]$$

- For i = 0..2*H-1, j = 0..W-1, the output samples r[i][j] are derived from the intermediate samples f[i][j] as follows:

$$r[i][2*j] = (f[i][j] + offset1) \gg shift1$$

$$r[i][2*j+1] = \left(\sum_{k=-2}^{1} c[k] * f[i][\text{Clip3}(0, W-1, j+k+1)] + offset2\right) \gg shift2$$

3.5.6. *Inverse Quantization from $D_{Y'}D_{Cb}D_{Cr\ to}$ $Y'C_bC_r$ domain*

This process dequantizes the input signal represented on BitDepthY bits for the Y component and BitDepthC bits for the chroma components (C_b, C_r) into a (float) signal $Y'C_bC_r$.

$$Y' = \mathrm{Clip}_{Y'}\left(\left(\frac{D'_Y}{\left(1 \ll (\mathrm{BitDepth}_Y - 8)\right)} - 16\right)/219\right)$$

$$Cb = \mathrm{Clip}_C\left(\left(\frac{D'_{Cb}}{\left(1 \ll (\mathrm{BitDepth}_C - 8)\right)} - 128\right)/224\right)$$

$$Cr = \mathrm{Clip}_C\left(\left(\frac{D'_{Cr}}{\left(1 \ll (\mathrm{BitDepth}_C - 8)\right)} - 128\right)/224\right)$$

with $\mathrm{Clip}_{Y'}(x) = \mathrm{Clip3}(0, 1.0, x)$

$\mathrm{Clip}_C(x) = \mathrm{Clip3}(-0.5, 0.5, x)$

Clip3(x, y, z) = x if z<x, y if z>y, z otherwise

3.5.7. *Colour Transformation from $Y'C_bC_r$ to RGB*

$Y'C_bC_r$ to R'G'B' with BT.709 primaries

R' = clipRGB (Y' + 1.57480 * C_r)

G' = clipRGB (Y' – 0.18733 * C_b – 0.46813 * C_r)

B' = clipRGB (Y' + 1.85563 * C_b)

Y'CbCr to R'G'B' with BT.2020 primaries

R' = clipRGB (Y' + 1.47460 * Cr)

G' = clipRGB (Y' – 0.16455 * C_b – 0.57135 * C_r)

B' = clipRGB (Y' + 1.88140 * C_b)

with clipRGB(x) = Clip3(0, 1, x)

Clip3(x, y, z) = x if z<x, y if z>y, z otherwise

3.5.8. Conversion from R'G'B' to RGB

R = 10000*inversePQ_TF (R')

G = 10000*inversePQ_TF (G')

B = 10000*inversePQ_TF (B')

with

$$\text{inversePQ\_TF}(N) = \left(\frac{\max\left[\left(N^{1/m_2} - c_1\right), 0\right]}{c_2 - c_3 N^{1/m_2}}\right)^{1/m_1}$$

$$m_1 = \frac{2610}{4096} \times \frac{1}{4} = 0.1593017578125$$

$$m_2 = \frac{2523}{4096} \times 128 = 78.84375$$

$$c_1 = c_3 - c_2 + 1 = \frac{3424}{4096} = 0.8359375$$

$$c_2 = \frac{2413}{4096} \times 32 = 18.8515625$$

$$c_3 = \frac{2392}{4096} \times 32 = 18.6875$$

3.6. Responses for HDR video compression [47] [48] [49]

3.6.1. Philips Response [47]

Philips proposed a method for efficiently compressing HDR video and it provides backward-compatible SDR video. Philips proposes a parameter based single layer HDR+SDR solution transmitting either an HDR or a SDR signal with parameters that enable the reconstruction of both the HDR and SDR signals. In the case an HDR signal with parameters is transmitted, the Y"u"v" color space and the Philips HDR EOTF can optionally be used for maximum image quality. The Philips submission for the CfE is based on transmitting SDR (YC_bC_r) with parameters in order to derive the original HDR signal at the decoder which is depicted clearly in the Figure 3-6.

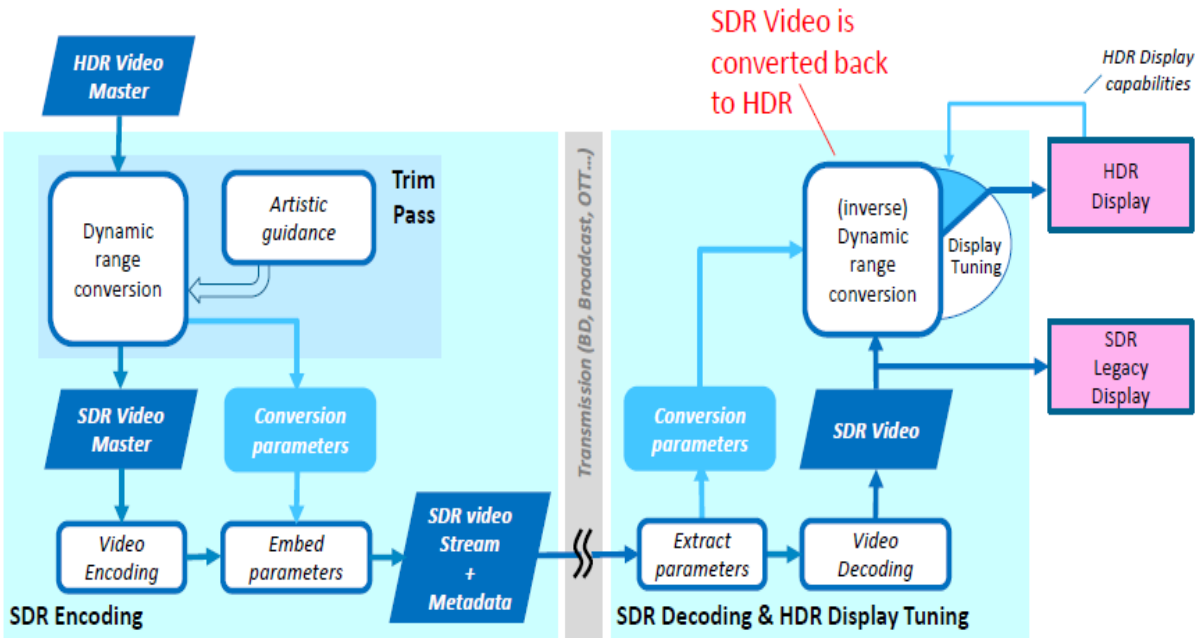


Figure 3-6 : Philips parameter based single layer HDR solution [47]

The HDR Video Master enters the processing chain in the left top corner of the figure. Grading (Dynamic Range conversion) results in an SDR Video Master. After MPEG Video (SDR) Encoding the Conversion parameters from the grading process are added as SEI messages to the MPEG video bitstream. This results in an SDR Video Stream + Metadata for transmission.

A standard SDR video decoder reproduces the received compressed content to SDR, as it would any other standard SDR content. A new HDR decoder will discover the metadata embedded in the content and be capable of performing the inverse dynamic range conversion of the SDR Video back to the original HDR Video. A display may then perform its own special operations to adapt the recreated HDR to its own display levels.

3.6.2. Technicolor Response [48]

Technicolor response is based on a single layer codec design and can provide SDR backward compatibility. Also pre-processing step is performed prior to the encoding. The resulting SDR video can be compressed, then decoded using legacy decoders (eg: HEVC Main 10) and directly rendered on SDR displays. The dynamic metadata is used to reconstruct the HDR signal from decoded SDR video, using a post-processing which is functional inverse of the pre-processing. Both HDR quality and artistic intent are preserved. Pre- and post-processing are applied independently per picture, do not involve any inter-sample dependency, and are codec independent.

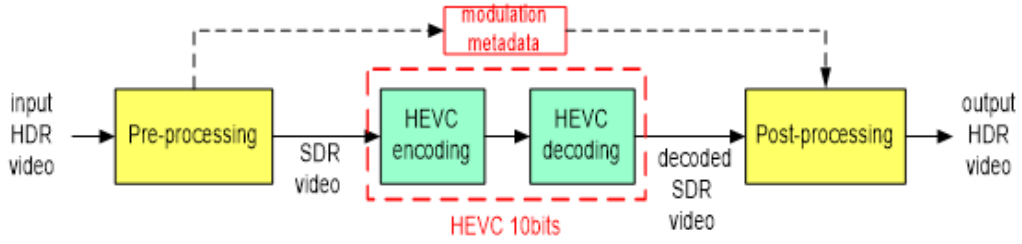


Figure 3-7: Schematic block diagram of Technicolor solution [48]

The input HDR video is first pre-processed to generate an SDR version and side metadata. This pre-processing applies for each picture and is similar to signal demodulation. Each HDR picture is split into one modulation value B_a (meta data)and one SDR picture. The SDR video is coded using for instance HEVC Main 10 profile, with side modulation values metadata. The decoded video is SDR and can be used by SDR displays. For HDR displays, a post-processing is applied to each reconstructed SDR picture. The post-processing modulates each SDR reconstructed picture with its related modulation values metadata to reconstruct the HDR picture version. The post-processing is the functional inverse of the pre-processing step, which guarantees artistic intent preservation of the HDR content.

3.6.3. *FastVDO Response [49]*

FastVDO proposed an efficient dual stream approach for HDR video coding. FastVDO has studied splitting the high bit-depth (HBD) or floating point video signal into two streams: a smoothed luminance (SL) signal, and an associated base signal (B). High Dynamic Range (HDR) and Wide Colour Gamut (WCG) video content is first converted into a FastVDO-developed integer color space YFbFr, designed to be closely aligned to the YCbCr color space of BT.709, that separates luminance and chrominance components. From the Y component, a gray scale smoothed luminance signal (SL) is generated, and then used to generate a low bitdepth Base signal (B). Both signals are then converted to YFbFr 4:2:0 signals, and coded using the Main10 High Efficiency Video Coding (HEVC) standard. This approach provides backward compatibility with the existing HEVC Main 10 Profile which can produce a usable SDR stream.

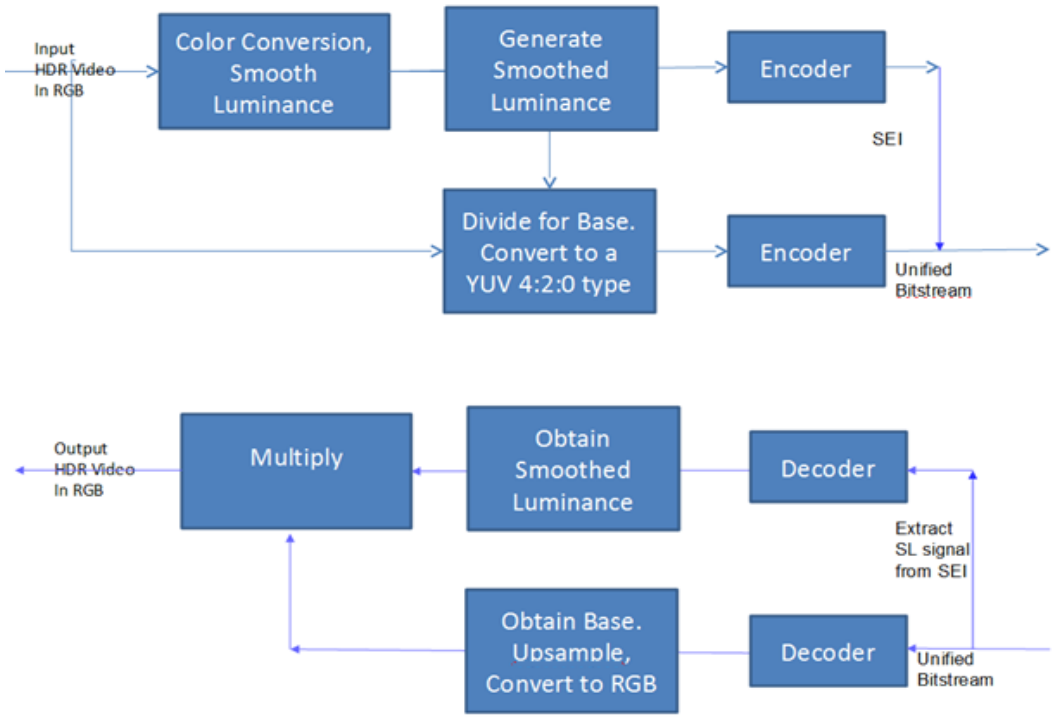


Figure 3-8: Flow diagram of FastVDO SEI-based method [49]

3.7. Proposed Scheme and Implementation

The scheme in Figure 3-8 is a two stream approach where the smoothed Luminance (Y) is encoded as a separate channel. Although the visual quality and objective metrics are good with this approach, an increase in bitrate can be observed as the full resolution 'Y' has to be encoded. The proposed scheme in Figure 3-9 is a variant of the above mentioned scheme in which only one channel needs to be encoded and the smoothened Luminance (Y) value could be transmitted as an Supplemental Enhancement Information (SEI).The visual quality and objective metrics are improved with the proposed scheme compared to the anchor but slightly less than the FastVDO response [49]. Also the bitrate got increased slightly compared to the anchor but there is a reduction in bitrate compared to the scheme in Figure 3-8. But the main advantage is that only one channel needs to be coded and so existing decoder implementations can be used with very little changes

In the proposed scheme a Luma Y component is obtained using YC_bC_r from the input HDR video. Once after obtaining Luma component it is smoothened to get the low frequency information for which Gaussian smoothing is employed. After this the arithmetic mean is calculated which is used to convert the HDR video to SDR video and this is obtained by the divide block which divides the original HDR video with the arithmetic mean generated. Now the SDR video is given as the input to the pre-processing block and HEVC encoding and decoding is performed and given as the input for the post-processing block and the output of this is multiplied with the arithmetic mean generated to obtain the output HDR video. After implementing this approach it is observed that there is good amount of increase in PSNR (tPSNR-X, tPSNR-Y, tPSNR-Z, tPSNR-XYZ, tOSNR-XYZ, PSNR_DE100, PSNR_L100) with a slight increase in bitrate over the anchor. There was an overall gain in the average BD-bit rate by 2.4% to 8.3% taking different metrics into consideration which can be seen in Table 4-10.

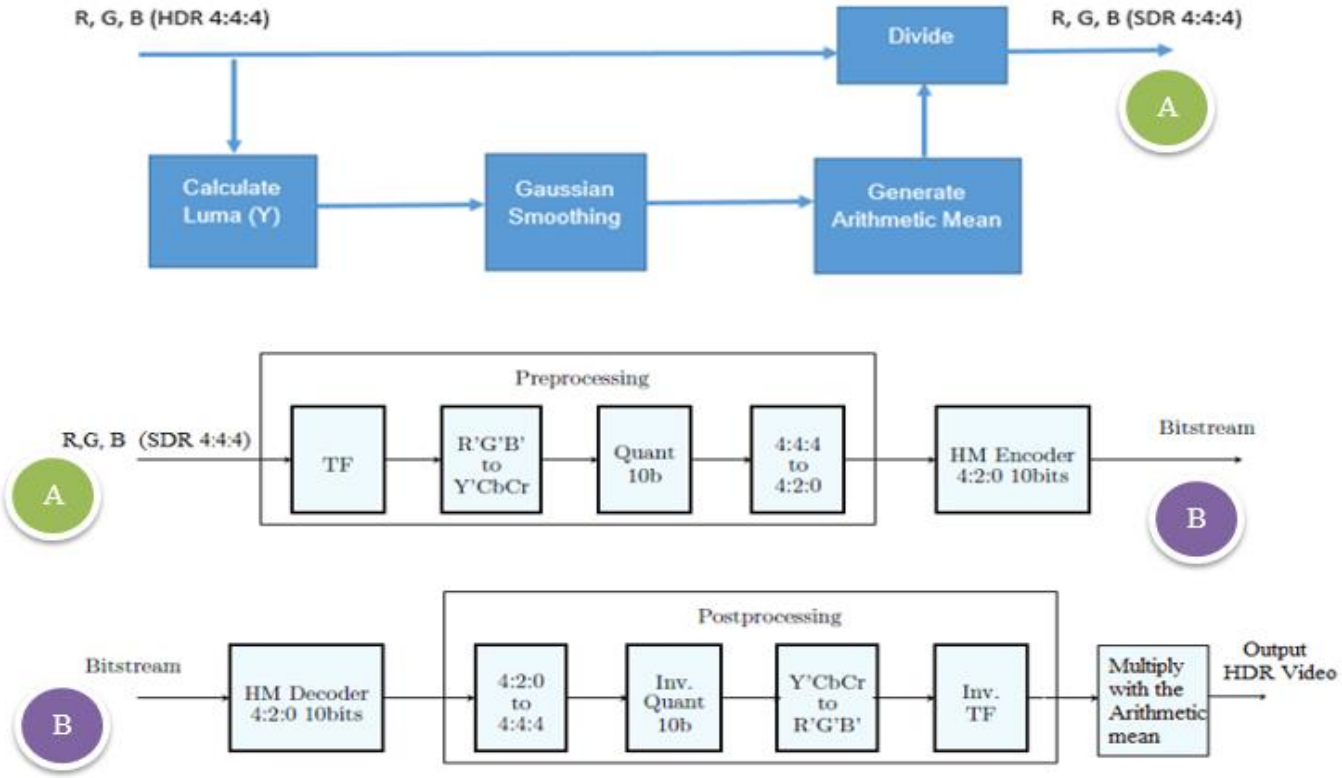


Figure 3-9: Implementation of the proposed scheme

3.8. Summary

Chapter 3 describes the concepts of High Dynamic Range (HDR) and Wide Colour Gamut (WCG), encoding and decoding workflow process, gives an overview about the responses for HDR video compression and discusses the proposed scheme and its implementation. Chapter 4 provides the results of the proposed scheme. The conclusions and insight into future work are given in Chapter 5.

Chapter 4

RESULTS

Different test sequences are used for simulation purpose. One GOP (8-picture) for each test sequence is taken and metrics related to PSNR (Peak Signal to Noise Ratio) in dB and bitrate in kilobits per second (kbps) are measured and represented in tabular format for each case. The metrics related to PSNR are plotted against bitrate for anchor and the proposed scheme.

Table 4-1 Test Sequences

Name of the Sequence	***fps***	***Frames***
FireEater2Clip4000r1_1920x1080p_25_hf_709_ct2020_444_xxx.exr	25	1 GOP
Market3Clip4000r2_1920x1080p_50_hf_709_ct2020_444_xxx.exr	50	1 GOP
BalloonFestival_1920x1080p_24_hf_709_ct2020_444_xxx.exr	24	1 GOP
Warm_1920x1080p_24_12_P3_ct2020_444_xxx.exr	24	1 GOP

4.1 Quality Metrics for Anchor and Proposed scheme

Table 4-2 FireEater2Clip.exr sequence anchor quality metrics

QP	Bit rate (kbps)	tPSNR-X	tPSNR-Y	tPSNR-Z	tPSNR-XYZ	tOSNR-XYZ	PSNR_DE100	PSNR_L100
20	1510.475	45.616	47.461	45.512	46.107	43.123	43.662	47.349
23	980.375	44.494	46.371	44.694	45.107	42.537	43.085	46.591
26	631.375	43.139	45.016	43.782	43.908	41.698	42.449	45.776
29	408.45	41.832	43.636	42.653	42.641	40.745	41.815	44.992

Table 4-3 FireEater2Clip.exr sequence proposed scheme quality metrics

QP	Bit rate (kbps)	tPSNR-X	tPSNR-Y	tPSNR-Z	tPSNR-XYZ	tOSNR-XYZ	PSNR_DE100	PSNR_L100
20	1719.775	45.99	47.78	45.908	46.479	43.548	43.875	47.495
23	1073.525	44.942	46.733	45.129	45.528	42.911	43.33	46.733
26	683.275	43.614	45.379	44.154	44.318	42.088	42.704	45.916
29	441.225	42.311	44.025	43.17	43.108	41.262	42.093	45.147

Table 4-4 Market3Clip.exr sequence anchor quality metrics

QP	Bit rate (kbps)	tPSNR-X	tPSNR-Y	tPSNR-Z	tPSNR-XYZ	tOSNR-XYZ	PSNR_DE100	PSNR_L100
21	8646.624	40.77	41.296	38.011	39.773	37.131	33.185	35.13
28	3518.616	36.445	36.717	34.284	35.672	33.361	31.753	33.176
31	2363.28	34.696	34.89	32.837	34.037	31.92	31.175	32.414
33	1776.816	33.542	33.698	31.765	32.909	30.875	30.8	31.92

Table 4-5 Market3Clip.exr sequence proposed scheme quality metrics

QP	Bit rate (kbps)	tPSNR-X	tPSNR-Y	tPSNR-Z	tPSNR-XYZ	tOSNR-XYZ	PSNR_DE100	PSNR_L100
21	8720.648	40.874	41.39	38.138	39.885	37.23	33.215	35.156
28	3596.616	36.516	36.805	34.361	35.75	33.422	31.772	33.201
31	2411.928	34.785	34.974	32.918	34.121	32.004	31.268	32.434
33	1821.624	33.635	33.795	31.841	32.996	30.934	30.91	31.948

Table 4-6 WarmNight.exr sequence anchor quality metrics

QP	Bit rate (kbps)	tPSNR-X	tPSNR-Y	tPSNR-Z	tPSNR-XYZ	tOSNR-XYZ	PSNR_DE100	PSNR_L100
21	4351.776	42.617	42.879	36.389	39.504	37.406	35.76	40.85
25	2561.712	40.909	41.143	35.575	38.383	36.27	35.351	39.979
29	1525.464	38.912	39.154	34.528	36.969	34.913	34.827	38.988
33	907.344	36.837	36.993	33.365	35.384	33.492	34.343	37.931

Table 4-7 WarmNight.exr sequence proposed scheme quality metrics

QP	Bit rate (kbps)	tPSNR-X	tPSNR-Y	tPSNR-Z	tPSNR-XYZ	tOSNR-XYZ	PSNR_DE100	PSNR_L100
21	4428.208	43.374	43.762	36.747	39.999	37.94	35.893	41.17
25	2644.784	41.649	41.947	35.988	38.921	36.885	35.499	40.307
29	1644.104	39.827	40.084	35.063	37.659	35.666	35.052	39.365
33	1018.216	37.829	38.002	33.99	36.18	34.289	34.603	38.321

Table 4-8 Balloon Festival.exr sequence anchor quality metrics

QP	Bit rate (kbps)	tPSNR-X	tPSNR-Y	tPSNR-Z	tPSNR-XYZ	tOSNR-XYZ	PSNR_DE100	PSNR_L100
22	5245.8	41.485	43.207	38.156	40.425	40.299	35.842	38.06
26	3156.6	39.04	40.569	36.108	38.164	38.098	34.723	36.848
29	2189.088	37.248	38.634	34.565	36.473	36.37	33.902	35.949
31	1729.992	36.144	37.349	33.737	35.472	35.57	33.534	35.36

Table 4-9 Balloon Festival.exr sequence proposed scheme quality metrics

QP	Bit rate (kbps)	tPSNR-X	tPSNR-Y	tPSNR-Z	tPSNR-XYZ	tOSNR-XYZ	PSNR_DE100	PSNR_L100
22	5329.368	41.678	43.388	38.394	40.641	40.474	35.976	38.11
26	3215.568	39.221	40.743	36.284	38.34	38.28	34.927	36.894
29	2269.152	37.4	38.786	34.733	36.634	36.554	34.102	36.004
31	1799.208	36.306	37.531	33.921	35.649	35.743	33.711	35.394

4.2 tPSNR-X (dB), tPSNR-Y (dB), tPSNR-Z (dB) vs Bit rate (kbps)

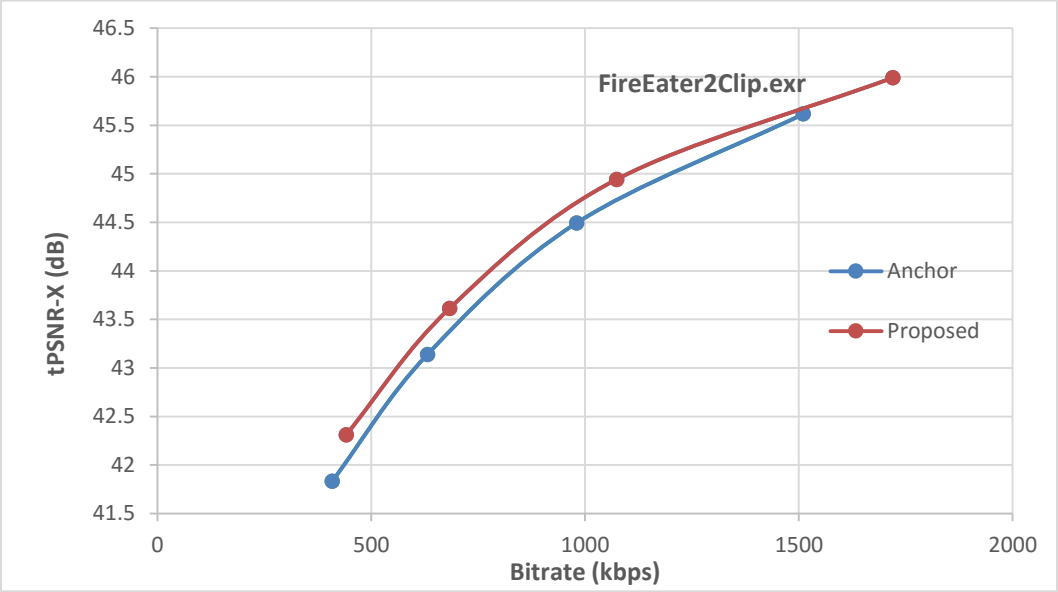


Figure 4-1: tPSNR-X (dB) vs Bit rate (kbps) for FireEater2Clip.exr

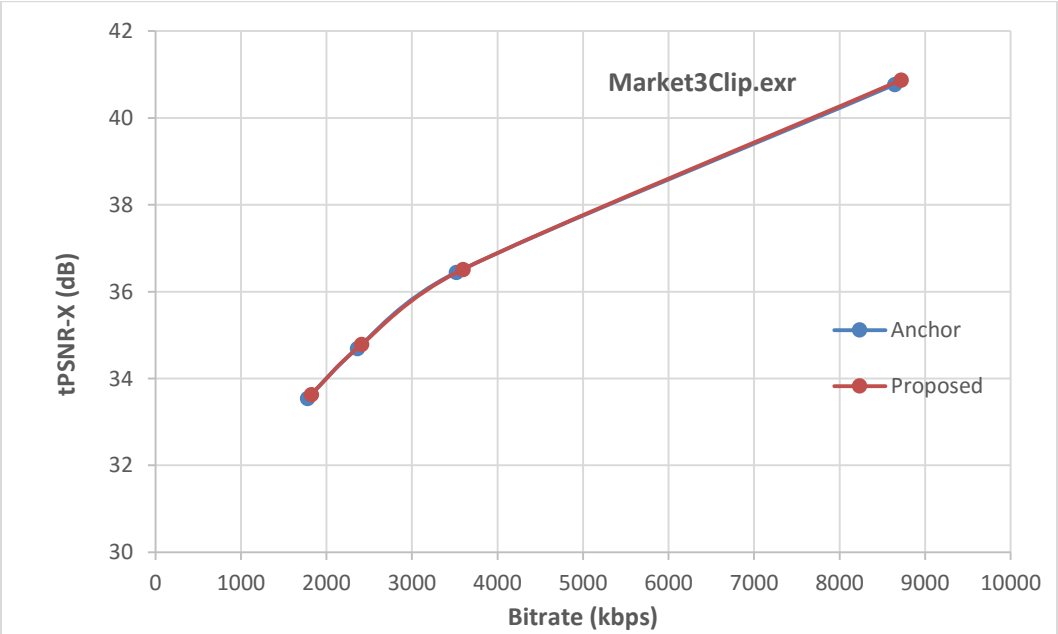


Figure 4-2: tPSNR-X (dB) vs Bit rate (kbps) for Market3Clip.exr

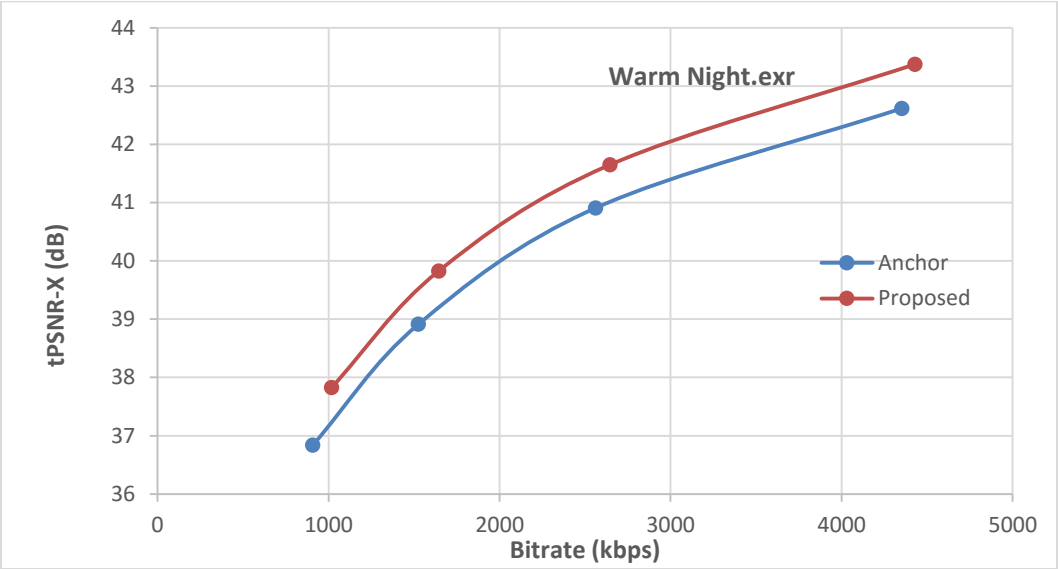


Figure 4-3: tPSNR-X (dB) vs Bit rate (kbps) for Warm Night.exr

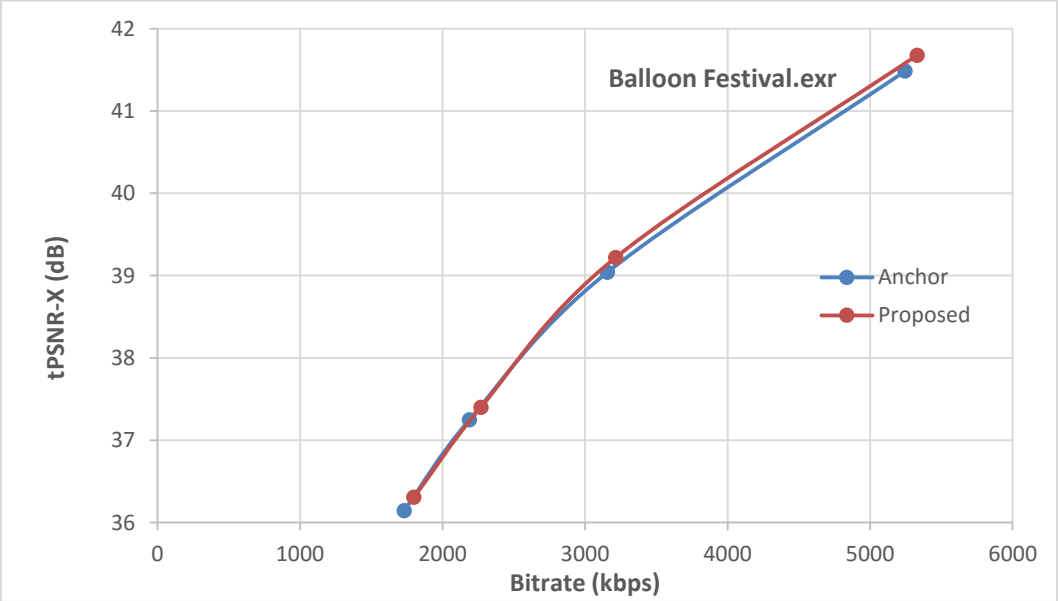


Figure 4-4: tPSNR-X (dB) vs Bit rate (kbps) for Balloon Festival.exr

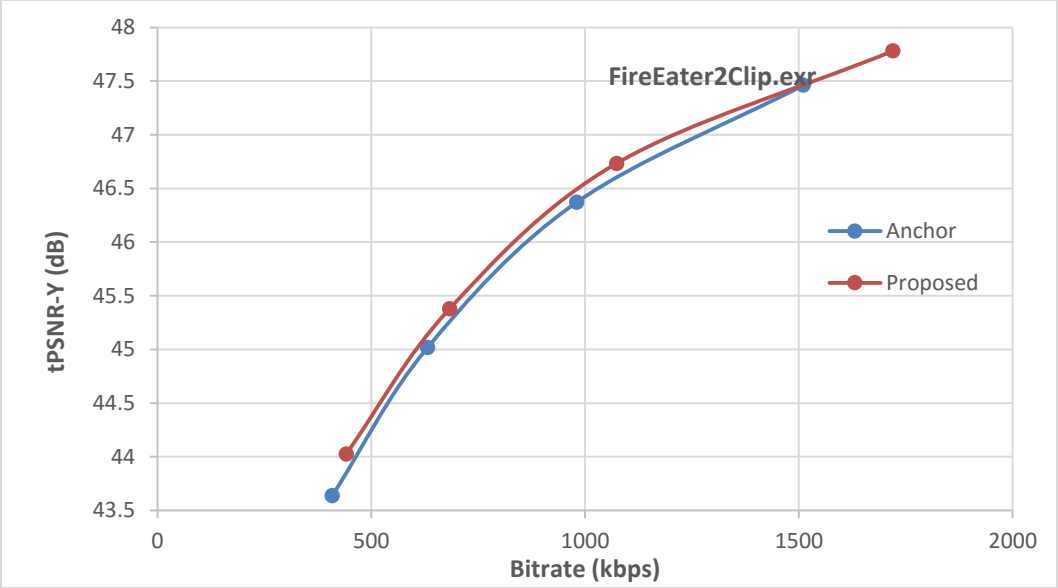


Figure 4-5: tPSNR-Y (dB) vs Bit rate (kbps) for FireEater2Clip.exr

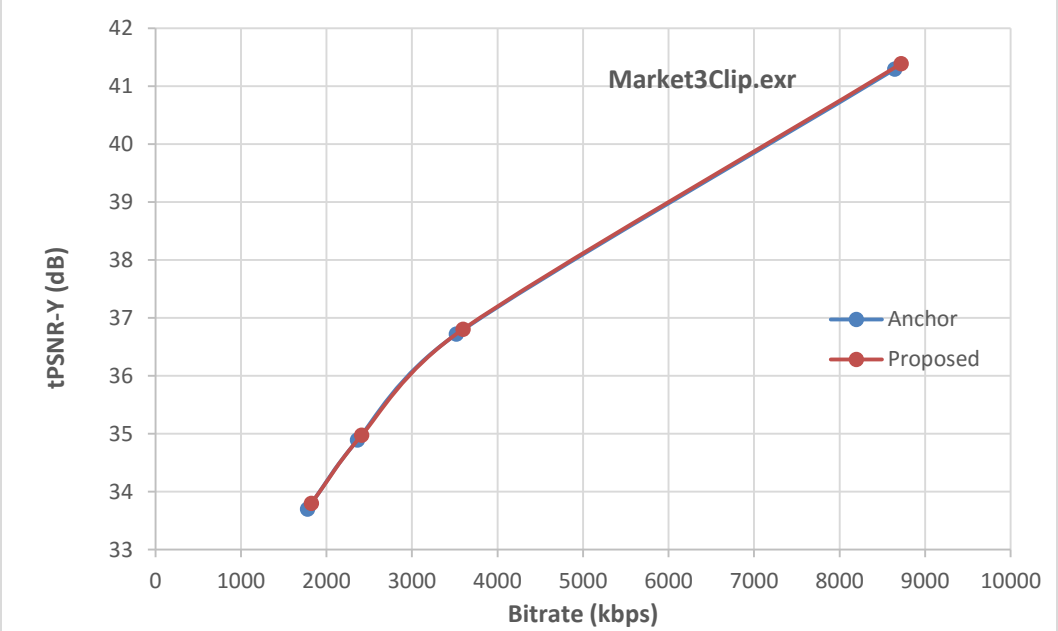


Figure 4-6: tPSNR-Y (dB) vs Bit rate (kbps) for Market3Clip.exr

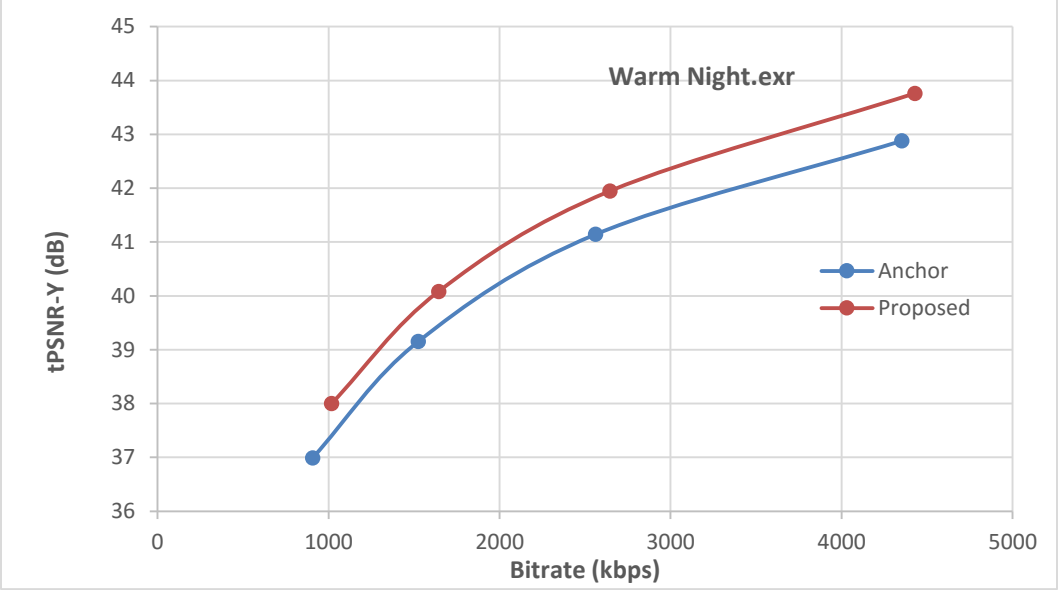


Figure 4-7: tPSNR-Y (dB) vs Bit rate (kbps) for WarmNight.exr

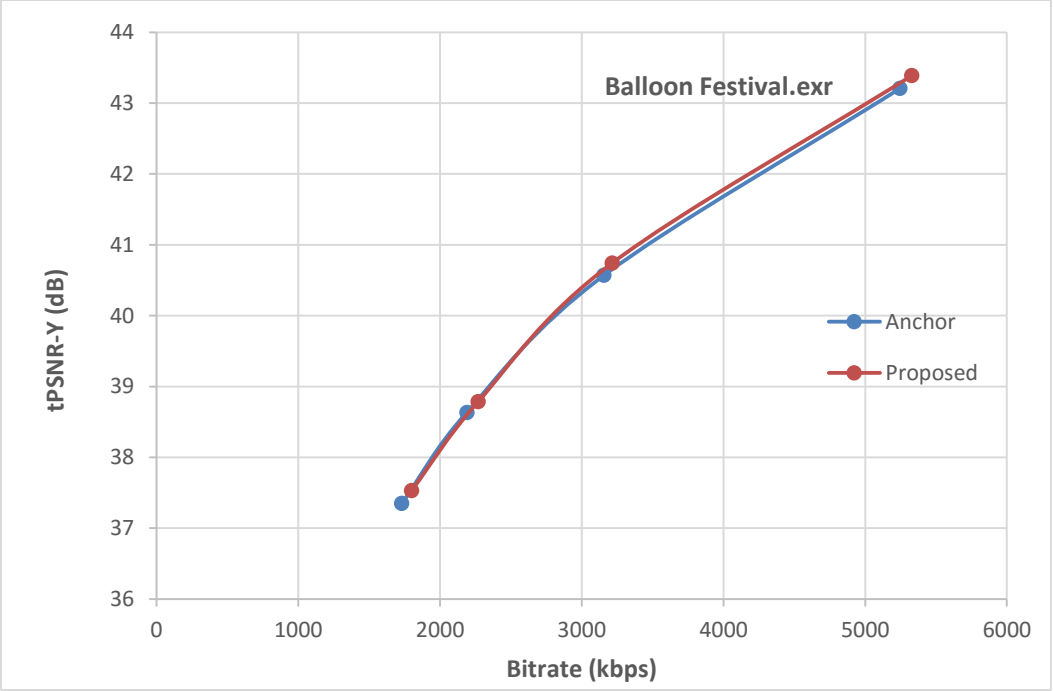


Figure 4-8: tPSNR-Y (dB) vs Bit rate (kbps) for Balloon Festival.exr

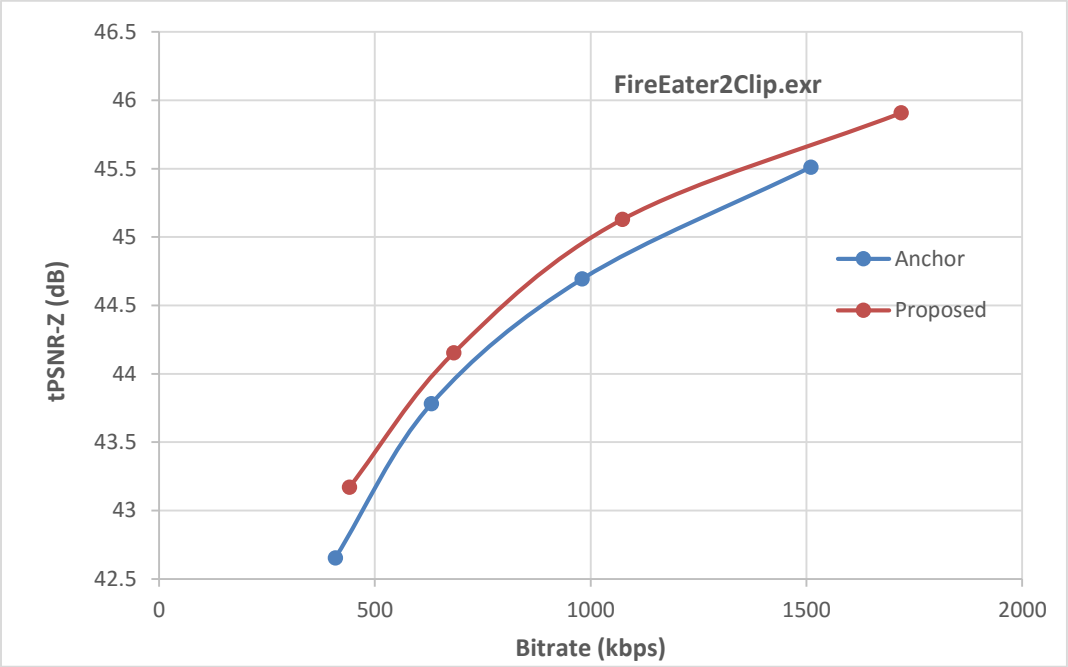


Figure 4-9: tPSNR-Z (dB) vs Bit rate (kbps) for FireEater2Clip.exr

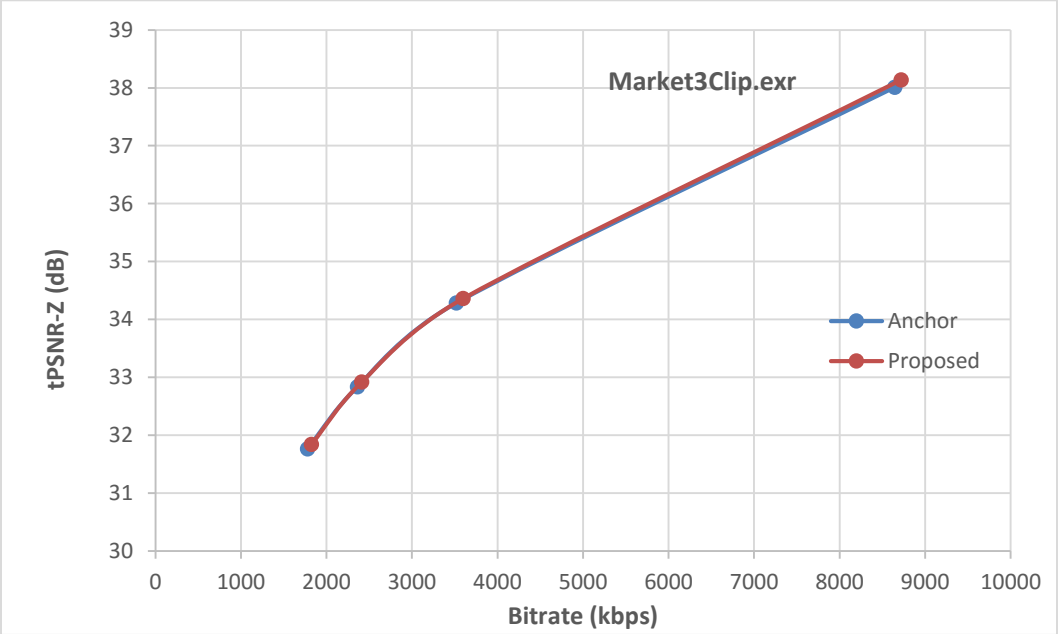


Figure 4-10: tPSNR-Z (dB) vs Bit rate (kbps) for Market3Clip.exr

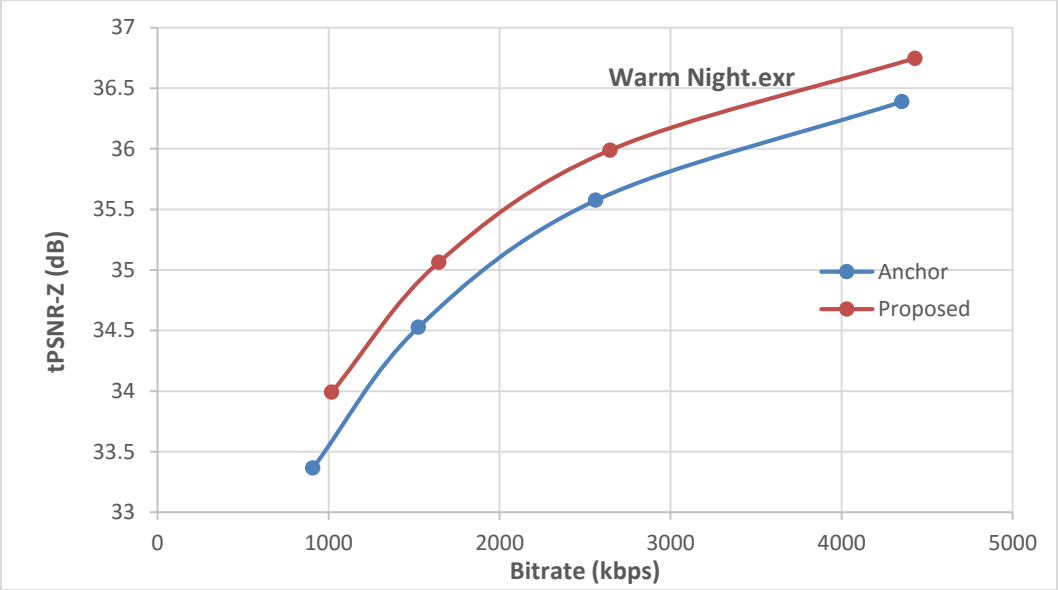


Figure 4-11: tPSNR-Z (dB) vs Bit rate (kbps) for Warm Night.exr

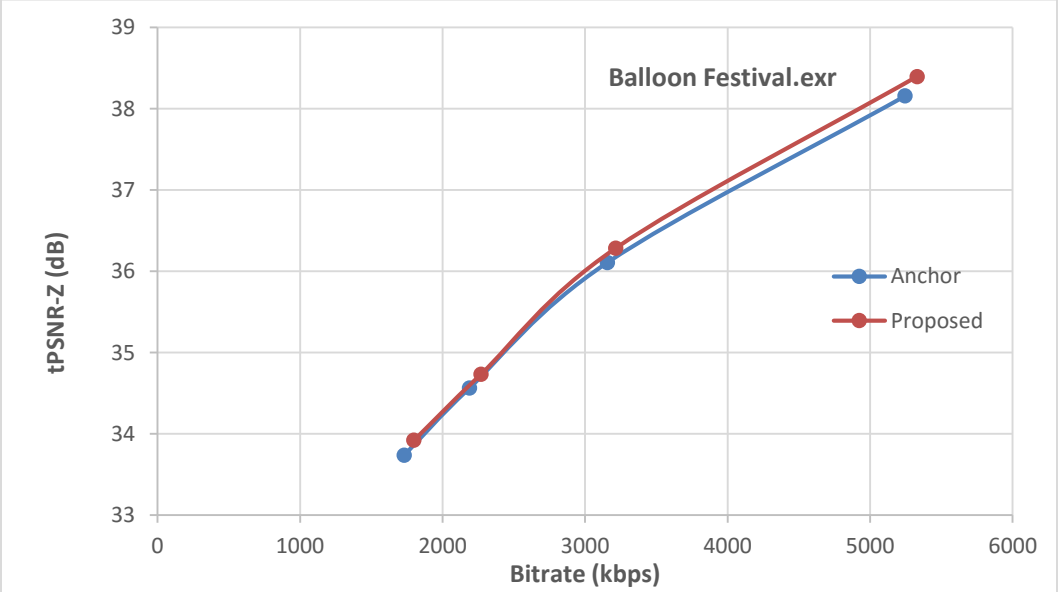


Figure 4-12: tPSNR-Z (dB) vs Bit rate (kbps) for Balloon Festival.exr

4.3 tPSNR-XYZ (dB) vs Bit rate (kbps)

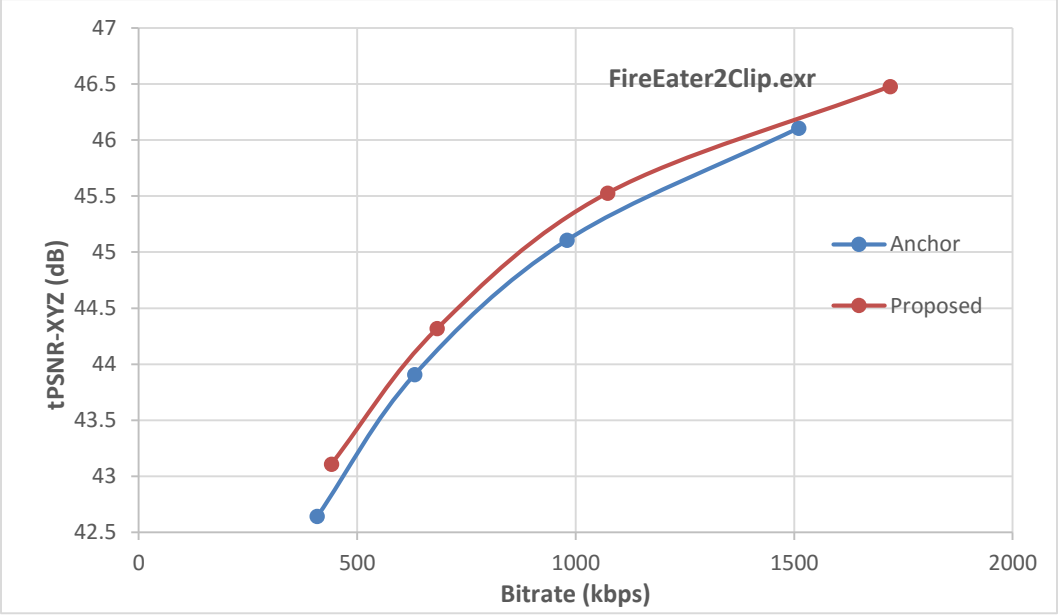


Figure 4-13: tPSNR-XYZ (dB) vs Bit rate (kbps) for FireEater2Clip.exr

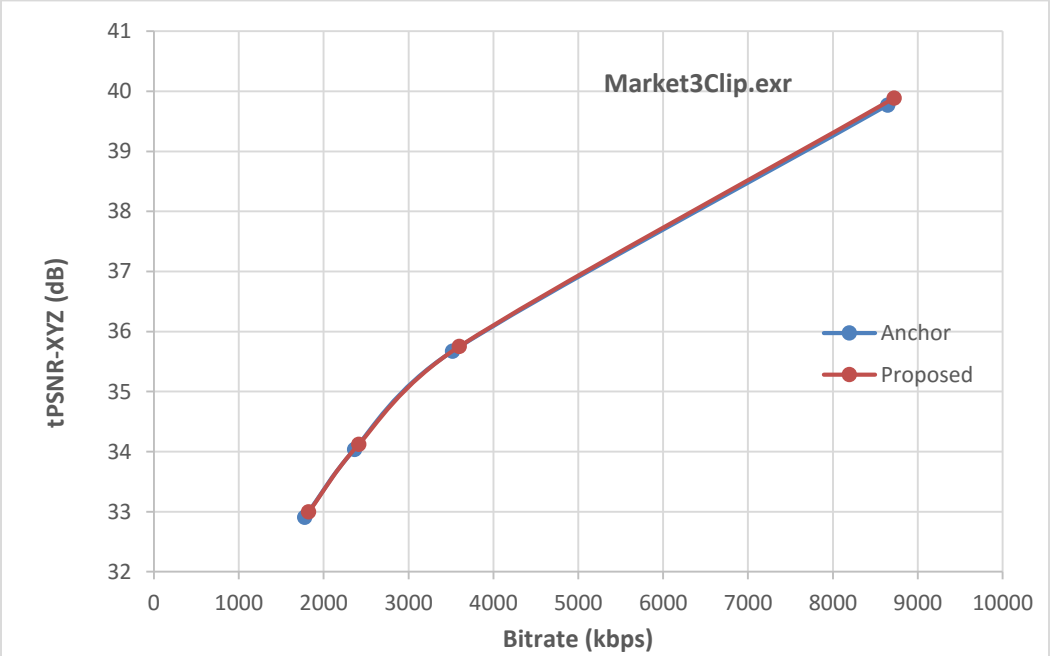


Figure 4-14: tPSNR-XYZ (dB) vs Bit rate (kbps) for Market3Clip.exr

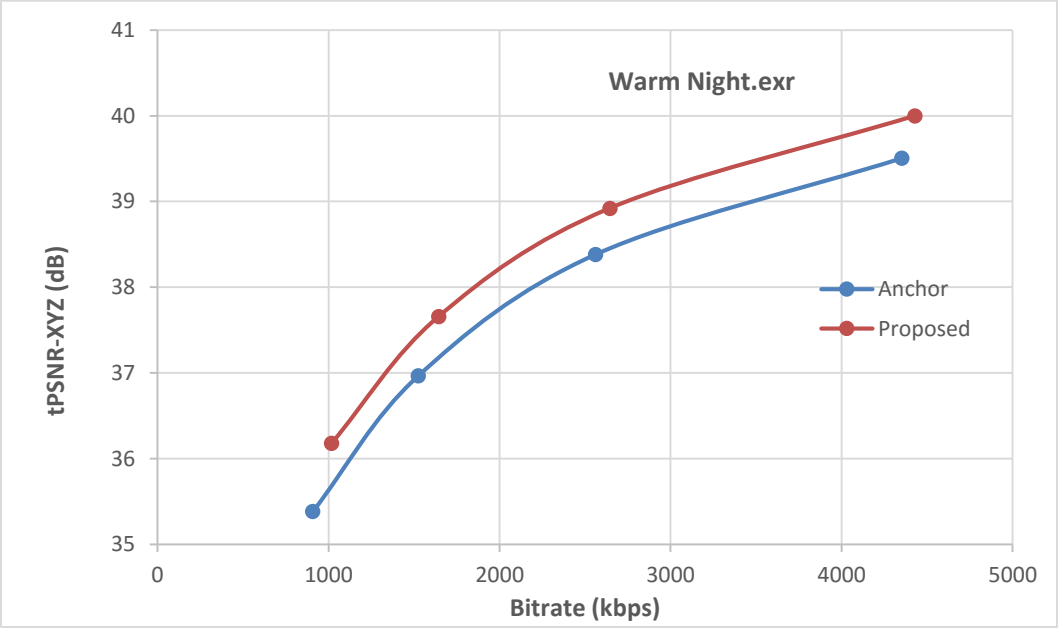


Figure 4-15: tPSNR-XYZ (dB) vs Bit rate (kbps) for Warm Night.exr

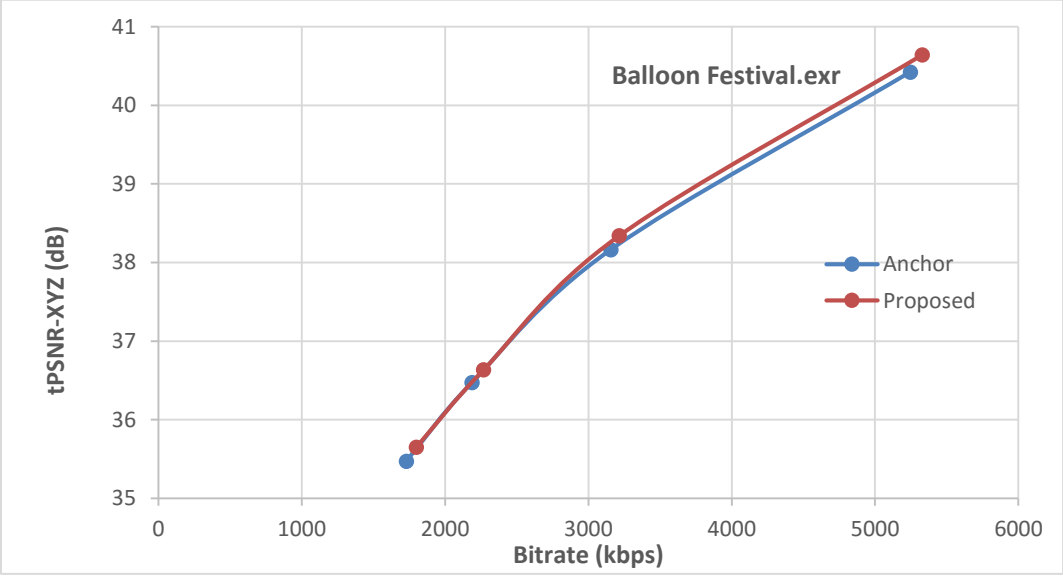


Figure 4-16: tPSNR-XYZ (dB) vs Bit rate (kbps) for Balloon Festival.exr

4.4 tOSNR-XYZ (dB) vs Bit rate (kbps)

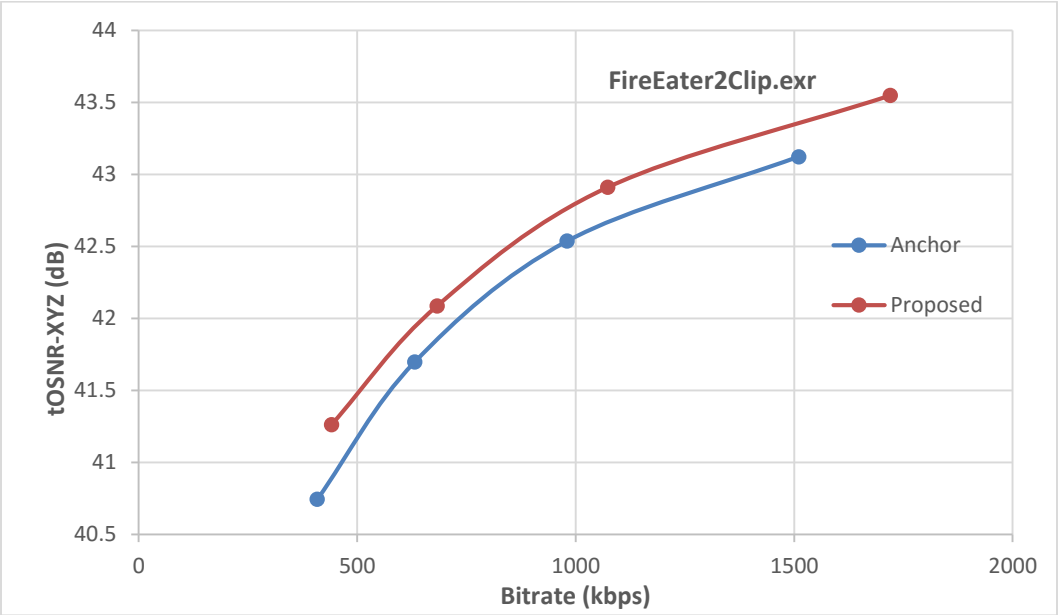


Figure 4-17: tOSNR-XYZ (dB) vs Bit rate (kbps) for FireEater2Clip.exr

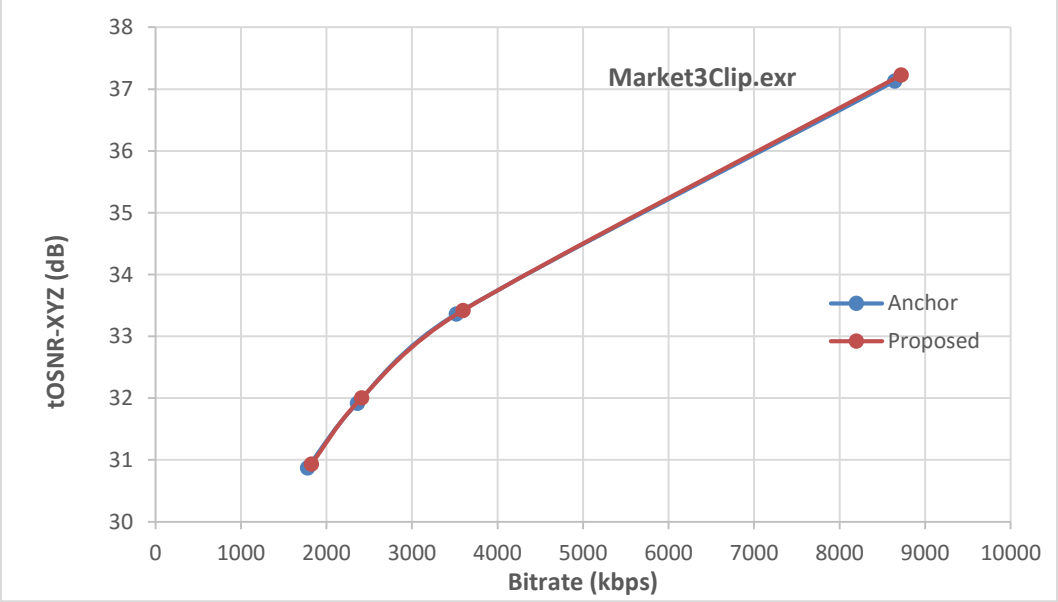


Figure 4-18: tOSNR-XYZ (dB) vs Bit rate (kbps) for Market3Clip.exr

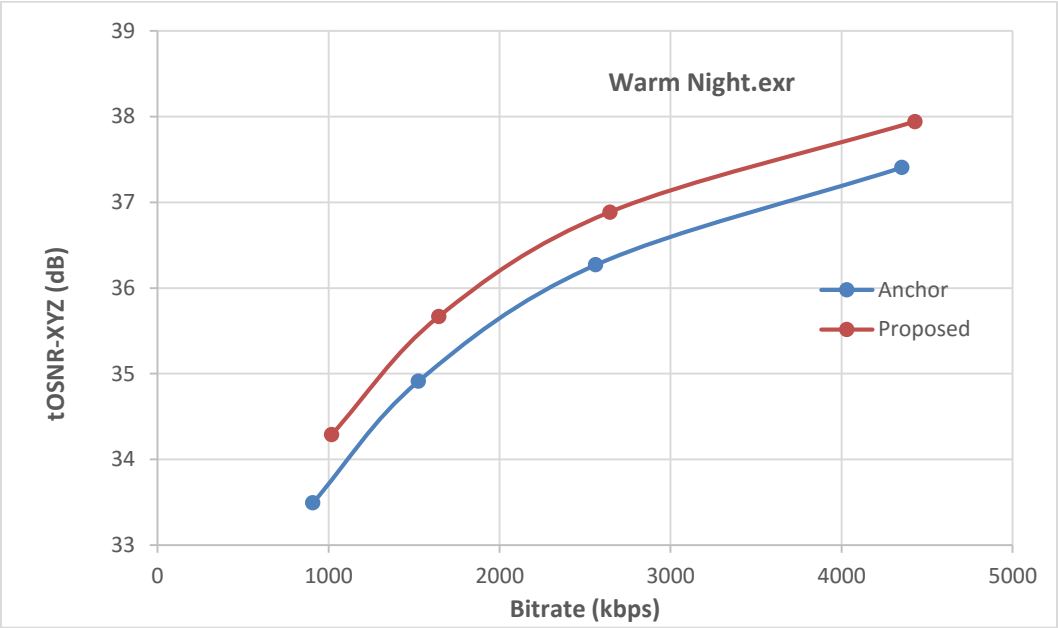


Figure 4-19: tOSNR-XYZ (dB) vs Bit rate (kbps) for Warm Night.exr

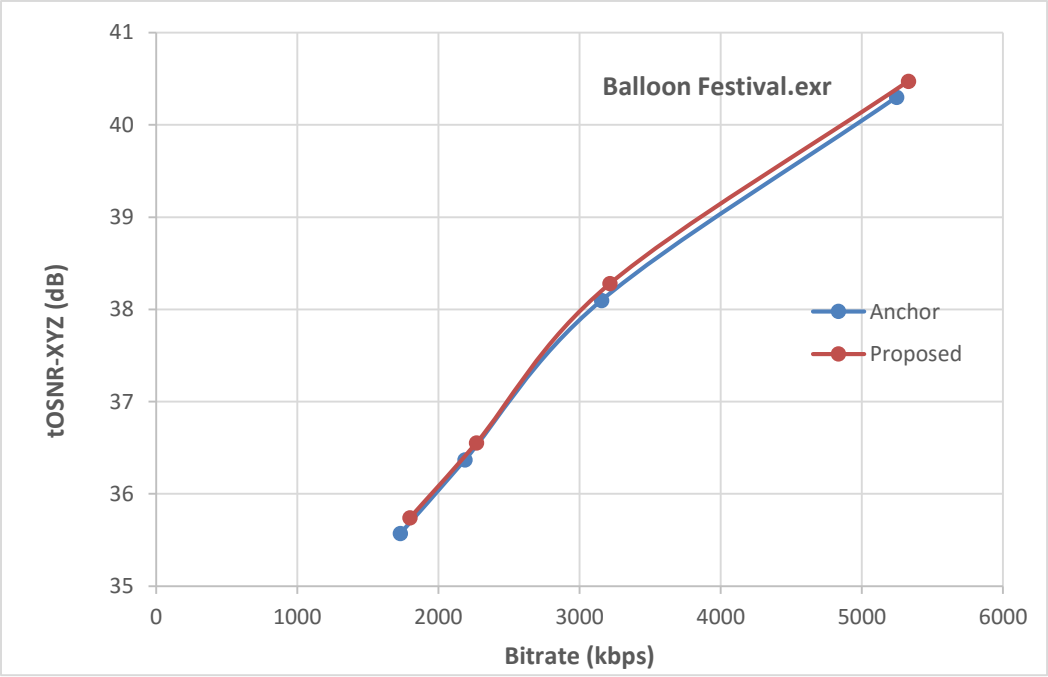


Figure 4-20: tOSNR-XYZ (dB) vs Bit rate (kbps) for Balloon Festival.exr

4.5 PSNR_DE100 (dB) vs Bit rate (kbps)

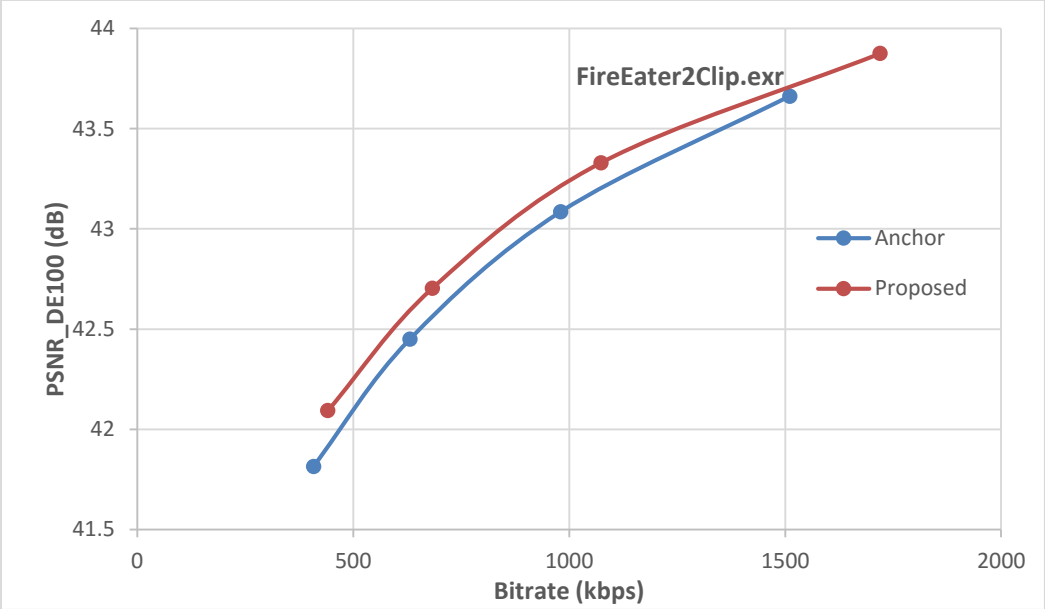


Figure 4-21:PSNR_DE100 (dB) Bit rate (kbps) for Balloon Festival.exr

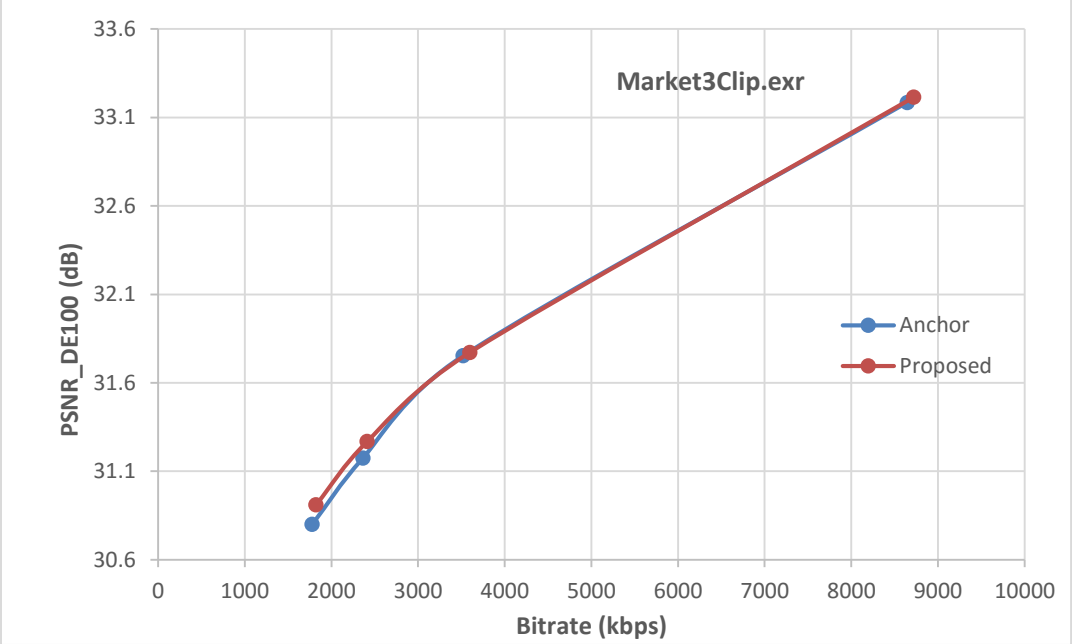


Figure 4-22: PSNR_DE100 (dB) Bit rate (kbps) for Balloon Festival.exr

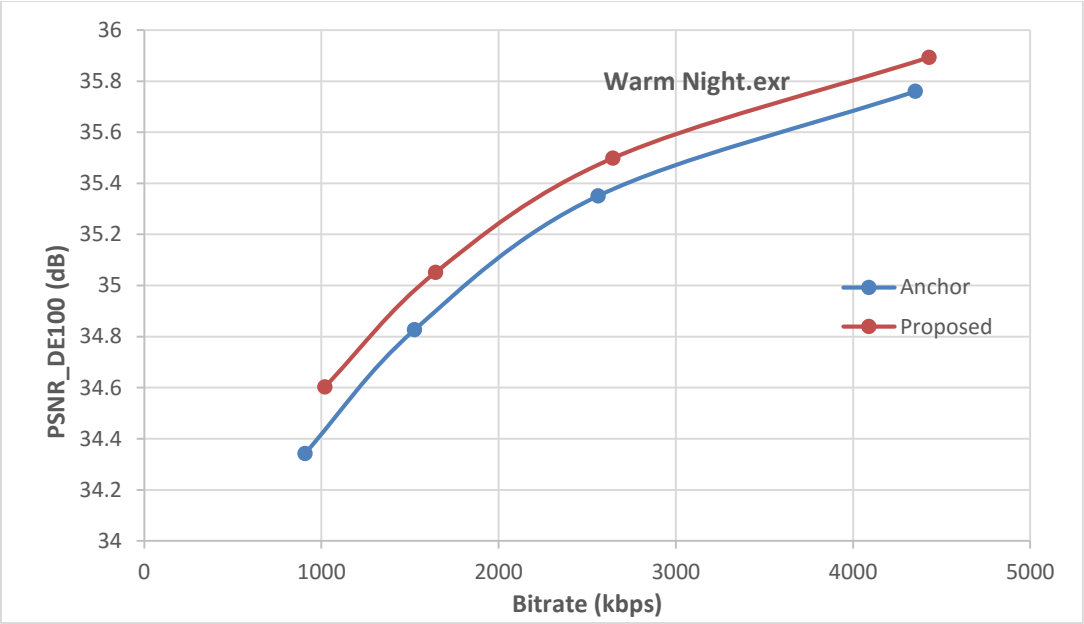


Figure 4-23: PSNR_DE100 (dB) Bit rate (kbps) for Balloon Festival.exr

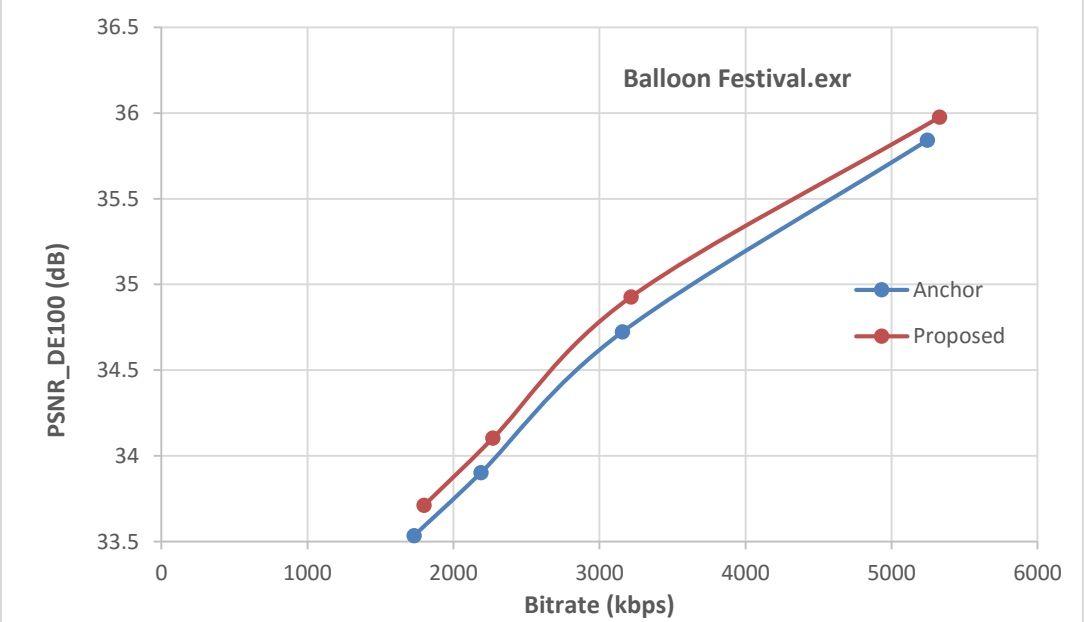


Figure 4-24: PSNR_DE100 (dB) Bit rate (kbps) for Balloon Festival.exr

4.6 PSNR_L100 (dB) vs Bit rate (kbps)

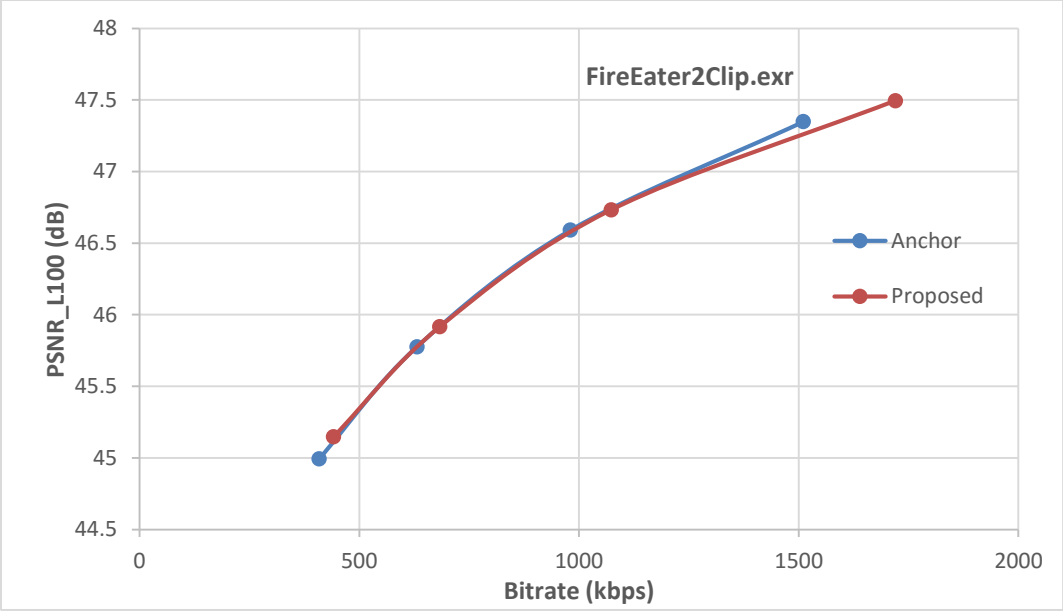


Figure 4-25: PSNR_L100 (dB) Bit rate (kbps) for FireEater2Clip.exr

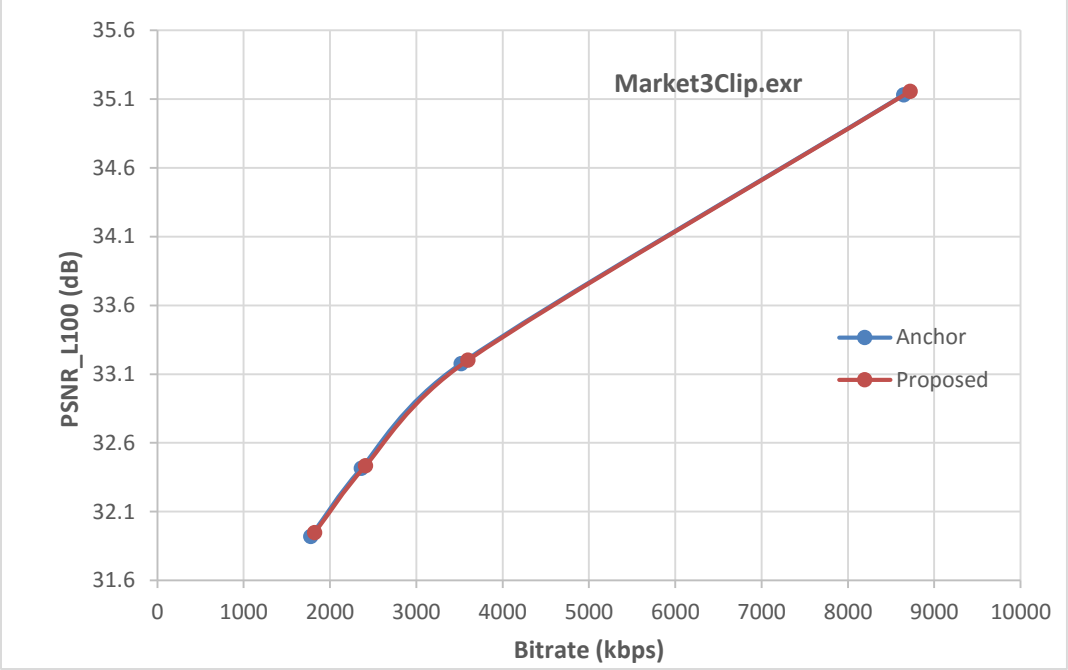


Figure 4-26: PSNR_L100 (dB) Bit rate (kbps) for Market3Clip.exr

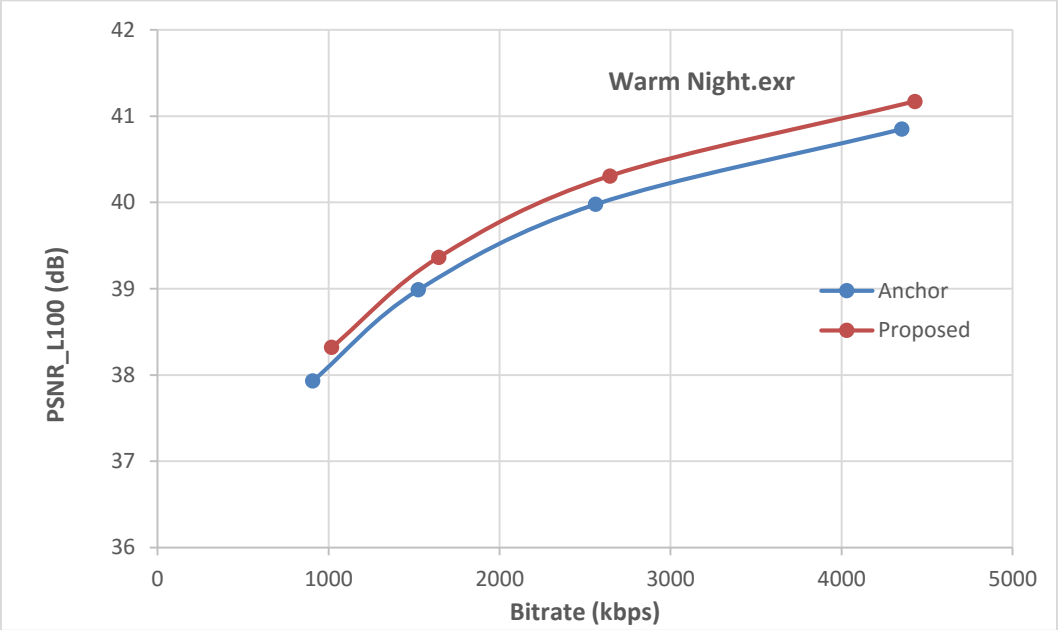


Figure 4-27: PSNR_L100 (dB) Bit rate (kbps) for Warm Night.exr

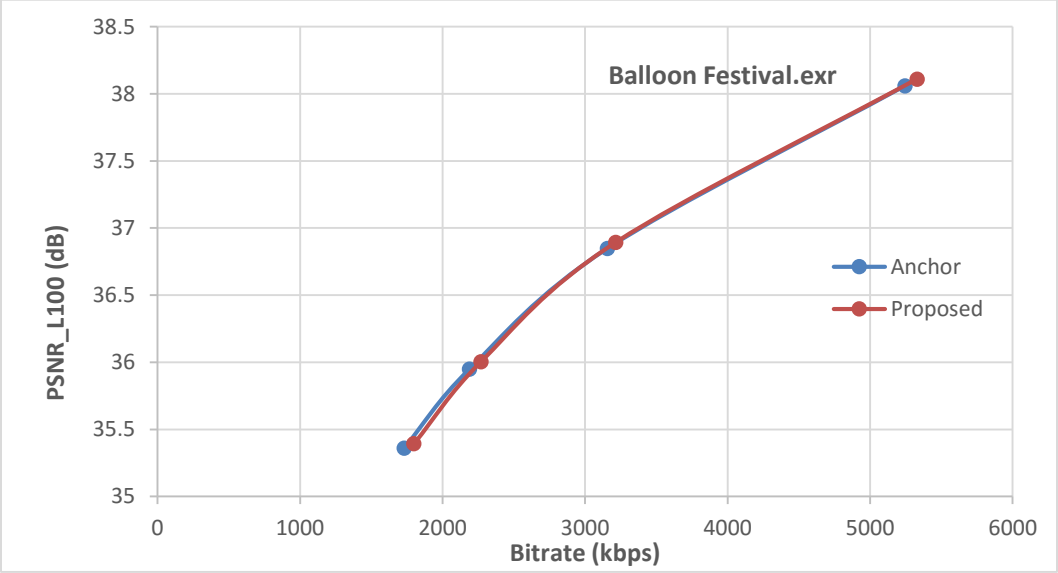


Figure 4-28: PSNR_L100 (dB) Bit rate (kbps) for Balloon Festival.exr

Table 4-10 BD-Bitrate computation using piece-wise cubic interpolation

Test Sequence	X	Y	Z	tPSNR-XYZ	tOSNR-XYZ	DE100	PSNRL100
FireEater2Clip	-7.1%	-3.8%	-10.3%	-7.1%	-12.7%	-9.0%	0.8%
Market3Clip	0.1%	0%	-0.3%	-0.2%	-0.4%	-0.2%	1.0%
Warm Night	-15.3%	-15.7%	-17.3%	-16.2%	-19.2%	-13.9%	-11.9%
Balloon Festival	-1.2%	-0.8%	-2.2%	-1.5%	-1.8%	-6.5%	0.5%
Overall	**-5.8%**	**-5.0%**	**-7.4%**	**-6.2%**	**-8.3%**	**-7.3%**	**-2.4%**

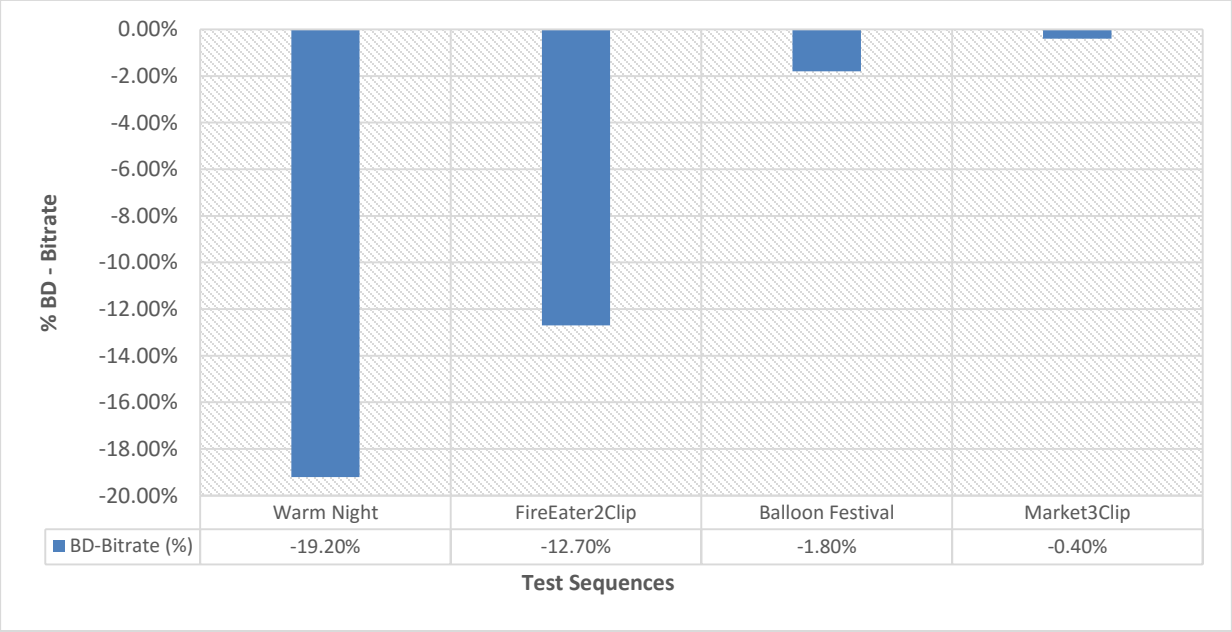


Figure 4-29: BD - Bitrate of all the test sequences with tOSNR-XYZ

Bjontegaard Delta bitrate (BD-bitrate) determines the quality of the encoded video sequence. Ideally BD-bitrate should decrease for a good quality video [50]. Table 4-10 illustrates the BD- bitrate computation for all the parameters using piece-wise cubic interpolation. From the Figure 4-29 it can be seen that the BD-bitrate has decreased by 0.5% to 20% for different test sequences which implies that the quality of the encoded video sequence using the proposed scheme has not degraded compared to the video encoded with the unaltered reference software.

Bjontegaard Delta PSNR (BD-PSNR) provides a good evaluation of the rate distortion (R-D) performance based on the R-D curve fitting [50]. Ideally, BD-PSNR should increase. It can be seen from Figure 4-30 that BD-PSNR has increased by 0.05dB to 0.5dB for different test sequences which also implies the qualities of the videos are not degraded when encoded using the proposed scheme.

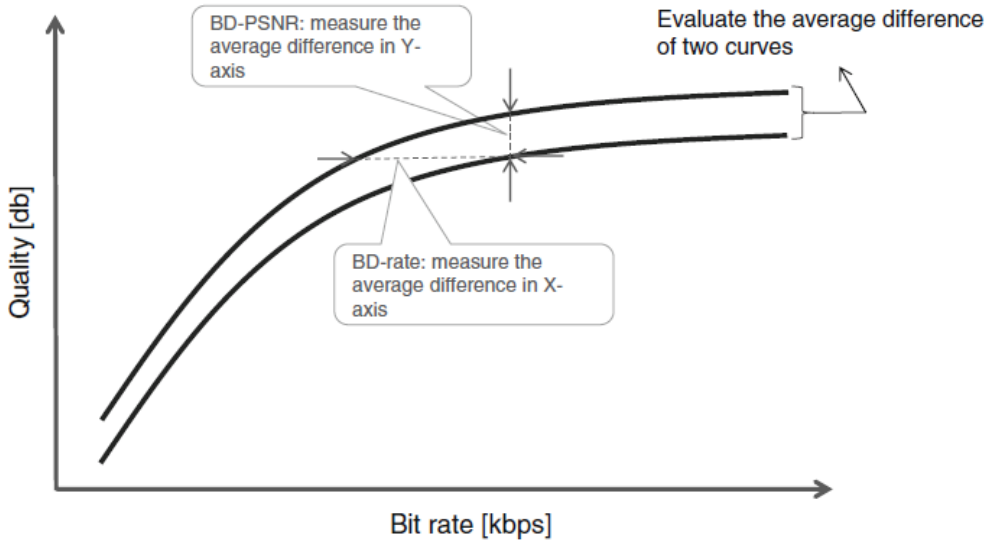


Figure 4-30: An example of R-D curve [3]

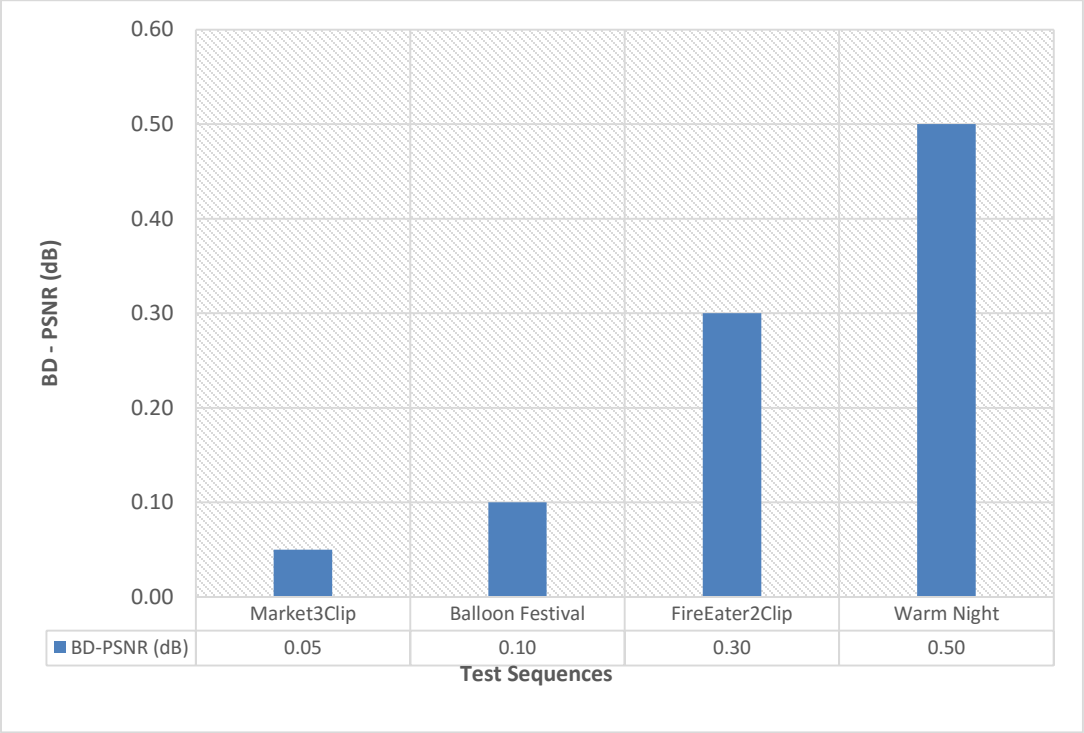


Figure 4-31: BD-PSNR of all the test sequences

4.4 Summary

In chapter 4, results and graphs for different test sequences and quantization parameters have been plotted, which compare the anchor with the proposed scheme. Chapter 5 will discuss about various conclusions that can be drawn from the results obtained in chapter 4 and it will also discuss about topics that can be explored as part of future work in the same direction.

Chapter 5

CONCLUSION AND FUTURE WORK

The objective of the thesis is to discuss different responses for HDR video compression and implement the proposed scheme that would make it possible to improve the visual quality which is the main element of concentration by adhering to the Objective metrics as seen in chapter 4.

It is observed that there is good amount of increase in PSNR (tPSNR-X, tPSNR-Y, tPSNR-Z, tPSNR-XYZ, tOSNR-XYZ, PSNR_DE100, PSNR_L100) with a slight increase in bitrate. There was an overall gain in the average BD-bitrate by 2.4% to 8.3% taking different metrics into consideration. Encoding time is almost the same as anchor. Also to achieve the same PSNR we require less BD-bitrate which is evident from the results in chapter 4. BD-bitrate has decreased by 0.5% to 20% and BD-PSNR has increased by 0.05dB to 0.5dB for different test sequences when the proposed scheme is compared with the anchor.

Further, Investigation of single layer and dual layer approaches for HDR video coding with SDR backward compatibility can be done. Efficient techniques for mapping HDR content to SDR monitors as well as HDR monitors at the same time can be explored. Also, experiments with multiple transfer functions can be performed and impact of peak luminance on the bit-depth can be studied. Sub sampling and its impact on coding performance and video quality is a good area to work [32]. Also implementing efficient parallel processing techniques during encoding can be explored.

APPENDIX A

Test Sequences

A 1: FireEater2Clip.exr

A 2: Market3Clip.exr

A 3: Balloon Festival.exr

A 4: Warm Night.exr

APPENDIX B

Test Conditions

Software

The code revision of reference software for HEVC encoder and decoder i.e., HM used for this research is HM 16.7 [21]

The code revision of reference software for HDR i.e., HDRTools used for this research is HDRCRM_1.1-dev [41]

All the work was done on a system with the following configuration:

- Operating System: Windows 10 Home Edition
- Processor: Intel(R) Core(TM) i5-3537U @ 2.00GHz 2.50GHz
- RAM: 8.00 GB
- System type: 64-bit Operating System, x64-based processor

File Exchange Formats & Coding Conditions [52]

Common test conditions (CTC) are desirable to conduct experiments in a well-defined environment and ease the comparison of the outcome of experiments. This section defines the CTC for HDR/WCG Video Coding Experiments, test sequences, QP values, and encoder configuration files to be used.

The filenames are specified as follows:

Name_Resolution_Fps_Format_ContentPrimaries_ContainerPrimaries_ChromaFormat_xxx.yyy

- Name: sequence name
- Resolution: picture size (e.g. 1920x1080p)
- Fps: frame rate in frames per second

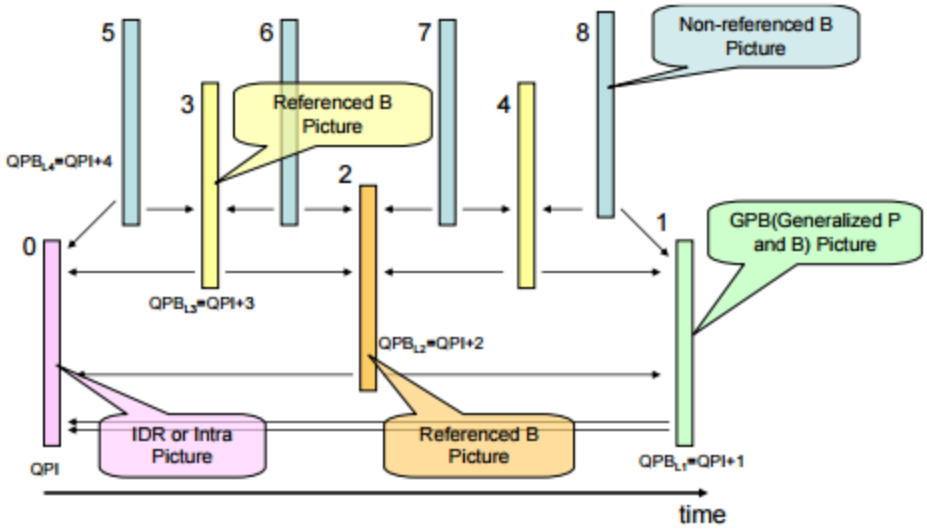
- Format: format of the samples (e.g. ff for 32-bit floating point, hf for half-float 16-bit floating point, 10 for 10-bit integer)
- Content primaries: colour primaries of the colour volume of the content, e.g. ITU-R Recommendations BT.709 and BT.2020, SMPTE ST 428-1:2006 P3 with D65 white point (P3D65)
- Container primaries: color primaries of the container (when different from the content primaries), e.g. ITU-R Recommendations BT.709 and BT.2020, SMPTE ST 428-1:2006 P3 with D65 white point (P3D65)
- Chroma format: e.g. 4:2:0, 4:2:2, or 4:4:4 (when applicable; for instance TIFF format involves 4:4:4 interleaved)
- xxx: frame number (when applicable)
- yyy: exr, tiff or yuv

Name of Sequence	**fps**	**Frames**
FireEater2Clip4000r1_1920x1080p_25_hf_709_ct2020_444_xxx.exr	25	1 GOP
Market3Clip4000r2_1920x1080p_50_hf_709_ct2020_444_xxx.exr	50	1 GOP
BalloonFestival_1920x1080p_24_hf_709_ct2020_444_xxx.exr	24	1 GOP
Warm_1920x1080p_24_12_P3_ct2020_444_xxx.exr	24	1 GOP

OpenEXR [55] is an open source image format used for special effects rendering and compositing. The format is a general purpose wrapper for the 16 bit half-precision floating-point data type, half. The Half format, or binary16, is specified in the IEEE 754-2008 standard [56]. OpenEXR also supports other formats such as both floating point and integer 32 bit formats [55]. Using the Half data type the format will have 16 bits per channel, or 48 bits per pixel. The OpenEXR format is able to cover the entire visible gamut and a range of about 10.7.orders of magnitude with a relative precision of 0.1%. Based on the fact that the human eye can see no more than 4 orders of magnitude simultaneously, OpenEXR makes for a good candidate for archival image storage.

TIFF (Tagged Image File Format) [57] is a widely supported and flexible image format. It provides support for a wide range of image formats wrapped into one file. The format allows the user to specify the type of image (CMYK, YC_bC_r, etc), compression methods, and also to specify the usage of any of the extensions provided for TIFF.

The Random Access (RA) coding constraint conditions shall be used for testing the sequences. A structural delay of processing units not larger than 8-picture "groups of pictures (GOPs)" (e.g., dyadic hierarchical B usage with 4 levels), and random access intervals of 1.1 seconds or less will be used (Intra random access picture every 24, 24, 32, 48 and 64 pictures for 24 fps, 25 fps, 30 fps, 50 fps and 60 fps sequences, respectively).



The HDR anchors are based on the HEVC Main 10 profile, which implies encodings using 4:2:0 subsampling and 10 bits per channel. The Y'C_bC_r colour space shall be used.

The encoding is performed with HM16.7[21], configured in 4:2:0 with the following settings:

- The macro REXt__HIGH_BIT_DEPTH_SUPPORT is set to 1, which results in
- FULL_NBIT set to 1
- REXt__HIGH_PRECISION_FORWARD_TRANSFORM set to 1

Quality Metrics [54]

To measure the performance of a video codec you can either perform subjective measurements or objective measurements. Subjective measurements will have human observers watching and rating the quality of the video. Objective measurements on the other hand you will have mathematical models and set of parameters designed to

approximate the results when doing subjective measuring. The output video should be served to the user and therefore subjective measurements are of more value. But they are usually very costly and time-consuming to gather. Therefore objective measurements are commonly used as a preliminary quality measurement.

Objective measurements can be done using a number of methods. This thesis will focus mainly on the tPSNR, PSNR_DE100, PSNR_L100 measurements introduced for the MPEG CfE, which both are variations of the PSNR (Peak Signal-to-Noise Ratio) measure. The reason for using these measures is that the non-linear behavior of the human visual system makes PSNR an ill-fitted measurement when it comes to image compression [58].

The metrics needed for HDR/WCG compression are still under investigation. Till now options for individual luminance and color metrics were considered in every CfE. The metrics needed to evaluate the compression scenario are yet to be finalized [54] [58]. The metrics which were identified and are in use are mentioned below. These metrics are taken into account to get the results in chapter 4.

PSNR: Peak signal-to-noise ratio (PSNR) is an expression for the ratio between the maximum possible value (power) of a signal and the power of distorting noise that affects the quality of its representation. Because many signals have a very wide dynamic range (ratio between the largest and smallest possible values of a changeable quantity), the PSNR is usually expressed in terms of the logarithmic decibel scale. PSNR is most commonly used to measure the quality of reconstruction of lossy compression codecs. PSNR is defined via Mean Square Error (MSE). Given a noise-free m x n monochrome image I and its noisy approximation K, MSE is defined as:

$$MSE = \frac{1}{m\,n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2$$

The PSNR is defined as,

$$\begin{aligned} PSNR &= 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \\ &= 20 \cdot \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right) \\ &= 20 \cdot \log_{10} (MAX_I) - 10 \cdot \log_{10} (MSE) \end{aligned}$$

Here, MAX_I is the maximum possible pixel value of the image.

tPSNR: When calculating tPSNR measurement [43] transfer function PQ_TF is taken into account. To calculate the measurement there are a number of steps applied for each sample of the two contents to compare.

- Each sample needs to be normalized to support a luminance of 10,000 cd/m^2, this is done by dividing the values by 10,000.
- The samples are then converted to XYZ, using the below equations depending on the color space of the samples (BT.709[36], BT.2020 [37],…)

  The equations to convert RGB to XYZ using BT.709 [36] are:

  X = 0.412391 * R + 0.357584 * G + 0.180481 * B

  Y = 0.212639 * R + 0.715169 * G + 0.072192 * B

  Z= 0.019331 * R + 0.119195 * G+ 0.950532 * B

  The equations to convert RGB to XYZ using BT.2020 [37] are:

  X = 0.636958 * R + 0.144617 * G + 0.168881 * B

  Y = 0.262700* R + 0.677998 * G + 0.059302 * B

  Z= 0.000000* R + 0.028073 * G+ 1.060985 * B
- Apply the transfer functions for each sample:

X' =PQ_TF(X), Y' = PQ_TF(Y), Z'=PQ_TF (Z)

- Four sums of Mean square error (MSE) values are computed between the two contents, MSE_x, MSE_y, MSE_z, and MSE_{xyz}.

  where $MSE_{xyz} = (MSE_x + MSE_y + MSE_z)/3$

- Finally, the PSNR values are computed for each MSE as:

$$tPSNR = 10 \log_{10} (1024^2/MSE)$$

Here $MAX_I = 1024$ when having an input with 10 bits per color channel.

Depending on this we calculate the tPSNR-X, tPSNR-Y, and tPSNR-XYZ.

tPSNR-Y: A luminance specific metric based on MSE calculations in nonlinear domain [54].

PSNR_DE100: The DeltaE is a color specific metric used with the CfE accounts for both lightness and color errors and hence responds to both luminance and color differences. Among the CfE metrics it showed good ability to respond to color distortions [54].

PSNR_L100: A luminance specific metric based on a human vision modelling the degree of visibility of differences. The visual model only accounts for luminance distortions [54].

APPENDIX C

List of Acronyms

AMVP – Advance Motion Vector Prediction

AVC - Advanced Video Coding

CABAC – Context Adaptive Binary Arithmetic Coding

CAVLC – Context Adaptive Variable Length Coding

CB – Coding Block

CTB – Coding Tree Block

CTU – Coding Tree Unit

CU – Coding Unit

DCT – Discrete Cosine Transform

DPB – Decoded Picture Block

DST – Discrete Sine Transform

EBU – European Broadcasting Union

EDR– Extended Dynamic Range

FPS – Frames per Second

GOP – Group of Pictures

HD – High Definition

HEVC – High Efficiency Video Coding

HDR – High Dynamic Range

IBC – Intra Block Copy

IDCT – Inverse Discrete Cosine Transform

ISO – International Standards Organization

ITU-T– International Telecommunication Union

JCT-VC - Joint Collaborative Team on Video Coding

JVT - Joint video team

Kbps – kilobits per second

MB - Macroblock

MC – Motion Compensation

MPEG – Moving Pictures Experts Group

NAL – Network Abstraction Layer

PB – Prediction Block

PQ – Perceptual Quantization

PSNR – Peak-Signal-to-Noise-Ratio

PU – Prediction Unit

QP – Quantization Parameter

RExt –Range Extensions

SAO – Sample Adaptive Offset

SDR – Standard Dynamic Range

SEI - Supplemental Enhancement Information

TB – Transform Block

TIFF – Tagged Image File Format

TU – Transform Unit

UHDTV – Ultra High Definition Television

URQ – Uniform Reconstruction Quantization

VCEG – Video Coding Experts Group

VLC – Variable Length Coding

VPS – Video Parameter Set

VPQM – Video Processing and Quality Metrics

VUI – Video Usability Information

WCG – Wide Colour Gamut

WPP – Wavefront Parallel Processing

REFERENCES

[1] I.E.G. Richardson, “Video Codec Design: Developing Image and Video Compression Systems”, Wiley, 2002.

[2] G. J. Sullivan et al, “Overview of the High Efficiency Video Coding (HEVC) Standard”, IEEE Trans. on Circuits and Systems for Video Technology, Vol. 22, No. 12, pp. 1649-1668, Dec. 2012.

[3] V. Sze, M. Budagavi and G. J. Sullivan, “High Efficiency Video Coding (HEVC) – Algorithms and Architectures”, Springer, 2014.

[4] G. J. Sullivan et al “Standardized Extensions of High Efficiency Video Coding (HEVC)”, IEEE Journal of selected topics in Signal Processing, vol. 7, pp.1001-1016, Dec. 2013.

[5] JVT Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264-ISO/IEC 14496-10 AVC), Mar. 2003, JVT-G050 [Online] Available: http://ip.hhi.de/imagecom_G1/assets/pdfs/JVT-G050.pdf

[6] M. Wien, “High efficiency video coding: Tools and specification”, Springer, 2015.

[7] V. Sze and M. Budagavi, “Design and Implementation of Next Generation Video Coding Systems (H.265/HEVC Tutorial)”, IEEE International Symposium on Circuits and Systems (ISCAS), Melbourne, Australia, June 2014 [Online] Available: http://www.rle.mit.edu/eems/publications/tutorials/

[8] Tutorial on HEVC by S. Riabtsev, “Detailed Overview of HEVC/H.265”. [Online] Available: https://app.box.com/s/rxxxzr1a1lnh7709yvih

[9] K.R. Rao, D.N. Kim and J.J. Hwang, “Video Coding Standards: AVS China, H.264/MPEG-4 Part 10, HEVC, VP6, DIRAC and VC-1”, Springer, 2014.

[10] K. R. Rao and J. J. Hwang, “Techniques and standards for image/video/audio coding”, Prentice-Hall, 1996.

[11] Tutorial on HEVC by D. Grois, et al, “HEVC/H.265 Video Coding Standard including the Range Extensions, Scalable Extensions, and Multiview Extensions”, IEEE International Conference on Image Processing (ICIP), Quebec city, Canada, 2015. [Online] Available:https://datacloud.hhi.fraunhofer.de/owncloud/public.php?service=files&t=8edc97d26d46d4458a9c1a17964bf881 (Password: a2FazmgNK)

[12] J. R. Ohm et al, “Comparison of the Coding Efficiency of Video Coding Standards—Including High Efficiency Video Coding (HEVC)”, IEEE Trans. on Circuits and Systems for Video Technology, Vol. 22, No. 12, pp. 1669-1684, Dec. 2012.

[13] 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.

[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] B. Bross et al, “High Efficiency Video Coding (HEVC) Text Specification Draft 10”, Document JCTVC- L1003, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Mar. 2013.[Online]. Available: http://phenix.it-sudparis.eu/jct/doc_end_user/current_document.php?id=7243

[16] C.Fogg, “Suggested Figures for the HEVC specification”, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC) document JCTVC-J0292rl, July 2012.

[17] D. Grois et al, “Performance Comparison of H.265/ MPEG-HEVC, VP9,and H.264/MPEG-AVC Encoders”, IEEE Picture Coding Symposium (PCS), pp.394-397, Dec.2013.

[18] HEVC tutorial by I.E.G. Richardson: http://www.vcodex.com/h265.html

[19] H.264 tutorial by I.E.G. Richardson: https://www.vcodex.com/h264.html

[20] HEVC white paper – Elemental Technologies https://www.elementaltechnologies.com/resources/white-papers/hevc-h265-demystified-primer

[21] Access to HM 16.7 Reference Software: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.7+SCM-5.4/

[22] Access to HM 16.7 Software Manual: http://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.0/doc/software-manual.pdf

[23] Access the MPL website: http://www.uta.edu/faculty/krrao/dip/Courses/EE5359/

[24] How to use HM Software: https://codesequoia.wordpress.com/2012/11/04/make-your-first-hevc-stream/ http://kohtoomaung.blogspot.com/p/blog-page_10.html

[25] Cisco White paper on “Visual Networking Index: Forecast and Methodology, 2014–2019”,http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.pdf

[26] HEVC walkthrough by vcodex : https://vimeo.com/65819014

[27] Relax its only HEVC: http://www.worldbroadcastingunions.org/wbuarea/library/docs/isog/presentations/2012B/2.4%20Bross%20HHI.pdf

[28] HEVC Analyzer: http://www.elecard.com/en/products/professional/analysis/hevc-analyzer.html

[29] M. J. Jakubowski and G. Pastuszak, “Block-based motion estimation algorithms – a survey,” Opto-Electronic Review, Vol. 21, pp 86-102, March2013.

[30] J. Sole et al, "Transform Coefficient Coding in HEVC", IEEE Trans. on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1765-1777, Dec. 2012.

[31] HEVC reference [Online]
Available: https://en.wikipedia.org/wiki/High_Efficiency_Video_Coding

[32] A. Luthra, E. Francois, W. Husak, ‘‘Requirements and Explorations for HDR/WCG Content Distribution and Storage,’’ ISO/IEC JTC1/SC29/WG11 MPEG2015/N15084, 111th MPEG meeting, Geneva, Feb. 2015

[33] EBU, “User requirements for video monitors in television production,” Tech 3320, Version 3.0, Geneva, Oct. 2014.

[34] E. Francois et al “High Dynamic Range and Wide Color Gamut Video Coding in HEVC: Status and Potential Future Enhancements”, IEEE Trans. Circuits and Systems for Video Technology, Vol. 26, No. 1, pp.63-75, Jan 2016.

[35] T. Smith, J. Guild, “The C.I.E. colorimetric standards and their use,” Transactions of the Optical Society vol. 33, no. 3, pp 73–134, Jan.1931.

[36] “Recommendation ITU-R BT.709-5, Parameter values for the HDTV standards for production and international programme exchange,” http://www.itu.int/rec/R-REC-BT.709 , ITU-T, April 2002.

[37] “Recommendation ITU-R BT.2020-1, Parameter values for ultra-high definition television systems for production and international programme exchange,” http://www.itu.int/rec/R-REC-BT.2020 , ITU-T, June 2014.

[38] K.Jansen, “The Pointer’s Gamut – The coverage of real surface colors by RGB color spaces and wide gamut displays”, TFT Central, Feb. 2014.

[39] S. Miller, M. Nezamabadi, S. Daly, ‘‘Perceptual signal coding for more efficient usage of bit codes,’’ SMPTE Motion Imaging Journal, vol. 122, no. 4, pp 52-59, May-June 2013.

[40] “SMPTE standard ST 2084, High Dynamic Range Electro-Optical Transfer Function of Mastering Reference Displays,” SMPTE, 2014.

[41] Access to HDR Tools Software: http://wg11.sc29.org/svn/repos/Explorations/XYZ/HDRTools/tags/0.10

[42] R. Mantiuk et al, “High dynamic range image and video compression Fidelity matching human visual performance,” IEEE ICIP 2007, vol. 1, pp 9-12, San Antonio, Sept. 2007.

[43] A. Luthra, E. Francois, W. Husak, ‘‘Call for Evidence (CfE) for HDR and WCG Video Coding,’’ ISO/IEC JTC1/SC29/WG11 MPEG2015/N15083, 111th MPEG meeting, Geneva, Feb. 2015.

[44] ‘‘Test Results of Call for Evidence (CfE) for HDR and WCG Video Coding,’’ ISO/IEC JTC1/SC29/WG11 MPEG2015/N15350, 112th MPEG meeting, Warsaw, June 2015.

[45] S. Lasserre et al, “High Dynamic Range video coding,” JCTVC-P0159, 16th JCT-VC meeting, San Jose, Jan. 2014.

[46] E. Reinhard et al, “High dynamic range video chains,” International Broadcasting Convention (IBC), Amsterdam, Sept. 2014.

[47] R. Goris et al, “Philips response to CfE for HDR and WCG,” ISO/IEC JTC1/SC29/WG11MPEG2015/M36266, Warsaw, Poland, July 2015.

[48] S.Lasserre et al, “Technicolor’s response to CfE for HDR and WCG- Single layer HDR video coding with SDR backward compatibility,” ISO/IEC JTC1/SC29/WG11 MPEG 2015/M36263, Warsaw, Poland, June 2015.

[49] W.Dai, M.Krishnan, P.Topiwala, “ An Efficient Dual-stream Approach for HDR video coding,” ISO/IEC JTC1/SC29/WG11 MPEG2015/M36251, Warsaw, Poland, June 2015.

[50] G. Bjontegaard, “Calculation of Average PSNR Differences between RD Curves”, document VCEGM33, ITU-T SG 16/Q 6, Austin, TX, Apr. 2001.

[51] G. Sullivan, " Color format down-conversion for test sequence generation," N6295, 67th MPEG meeting, Waikoloa, HI, December 2003.

[52] E. Francois, P. Yin, J. Sole, “ Common Test Conditions for HDR/WCG Video Coding Experiments,” ISO/IEC JTC1/SC29/WG11 MPEG2014/N15793, Geneva, Oct. 2015.

[53] “Recommendation ITU-R BT.1886 Reference electro-optical transfer function for flat panel displays used in HDTV studio production,” http://www.itu.int/rec/R-REC-BT.1886 , ITU-T, March 2011.

[54] V.Baroncini et al, “CE3: Report on the on objective metrics,” ISO/IEC JTC1/SC29/WG11 MPEG2015/m37365, Geneva, Oct. 2015.

[55] OpenEXR. http://www.openexr.com/. Accessed: February 2015.

[56] IEEE Standard for Floating-Point Arithmetic. *IEEE Std 754-2008*, pp 1-70, Aug. 2008.

[57] TIFF Revision 6.0. http://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf. Accessed: February 2015.

[58] R. Martin et al, " Subjective and Objective evaluation of HDR video compression," 9th International Workshop on Video Processing and Quality metrics for Consumer Electronics (VPQM), number EPFL-CONF-203874, 2015.

[59] High Dynamic Range Now Available on Amazon Instant Available for Prime members.[Online] http://phx.corporate-ir.net/phoenix.zhtml?c=176060&p=irol-newsArticle&ID=2062190 . Accessed: July 2015.

[60] W. Dai, M. Krishnan, and P.Topiwala, "Chroma sampling and modulation techniques in high dynamic range video coding," SPIE Applications of Digital Image Processing XXXVIII, Vol. 9599, pp 95990D1-14, Sept. 2015.

[61] Gaussian smoothing: http://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm

[62] F. Xiao et al, "High Dynamic Range Imaging of Natural Scenes," in Proc. 10th Color Imaging Conference, pp 337-442, Scottsdale, Nov. 2002.

[63] M. J. Jakubowski and G. Pastuszak, "Block-based motion estimation algorithms – a survey," Opto-Electronic Review, Vol. 21, pp 86-102, March2013.

[64] P. Topiwala, W. Dai, M. Krishnan, "Improvements to HDR10", IEEE Conference, 4-6 July, 2016, Santorini, GR.

[65] Exercise on Running video codec: http://www.cs.tut.fi/~vc/Exercises/Exercise%201.pdf

BIOGRAPHICAL INFORMATION

Srikanth received his Bachelor's Degree in Electronics and Communication Engineering from Koneru Lakshmaiah University, Vijayawada, India in 2013. He worked as an Engineer for two years with Honeywell.Inc, Bangalore, India.

With a passion for Signal Processing applications and Wireless Communications, he pursued his Master's degree in Electrical Engineering from the University of Texas at Arlington from August 2014 to May 2016. During this period, he worked with Dr. K.R. Rao and was actively involved in research with the Multimedia Processing Lab. His research interests include Motion Estimation and HDR video compression.

He was awarded with Research In Motion (RIM) Endowed Scholarship (2015 & 2016) and Electrical Engineering Scholarship (2014 & 2015) during his course of study at UT Arlington. He is presently working as a Fall/Spring Graduate Research Intern at FastVDO LLC, Florida.