MULTI-STAGE PREDICTION SCHEME FOR SCREEN CONTENT BASED ON HEVC

by

NAGASHREE MUNDGEMANE NAGARAJ

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

December 2015

Copyright © by Nagashree Mundgemane Nagaraj 2015

All Rights Reserved

Acknowledgements

Foremost, I would like to express my sincere gratitude to my advisor Dr. K. R. Rao for the continuous support of my thesis study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

I would like to thank my thesis committee members Dr. Howard Russell and Dr. Kambiz Alavi for their valuable time and taking interest to review my thesis work.

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

November 20, 2015

Abstract

MULTI-STAGE PREDICTION SCHEME FOR SCREEN CONTENT BASED ON HEVC

Nagashree Mundgemane Nagaraj, MS

The University of Texas at Arlington, 2015

Supervising Professor: K. R. Rao

Screen content is a computer generated content which contains text, graphics and animations and the coding of such content is different from coding a photographic content. This thesis emphasizes on coding the screen content keeping the present video coding standard High Efficiency Video Coding (HEVC) as the base. An extension to HEVC to support screen content coding (SCC) is currently under development by the Joint Collaborative Team on Video Coding (JCT- VC) and the main goal of the extension is to provide improved compression performance of text, graphics and animation in addition to photographic content.

This thesis focuses on a particular coding technique called palette mode to efficiently improve the coding abilities. The palette based coding takes the advantage of screen content having distinct limited number of colors and based on the colors and the structure of the content, the content is divided into palette table (color table) and palette index map. The palette index map represents the structure of the content and aims to exploit the correlation among the structural components. After coding the neighboring indices, it is observed that there are repeated patterns that are present in the structure and they may provide potential coding efficiency if repeated patterns are identified and coded using already coded patterns. This gives rise to two-stage prediction and coding schemes for screen content focusing to achieve bit-rate savings. If the coding of neighboring indices takes the first stage of coding, the second stage is of coding the non-local correlation of the repeated patterns. The proposed algorithm for the second stage coding combines the intra block copy full frame search mode for identifying the repeated pattern and palette mode

to code the matched pattern. The proposed algorithm was implemented on standard test sequences and it provides average bitrate savings of 2.5% for all intra and 1.86% for random access profiles for lossless coding.

TABLE OF CONTENTS

List of Illustrations

List of Tables

Chapter 1

INTRODUCTION

1.1 Evolution of Video Compression

High Definition (HD) and Ultra-High Definition (U- HD) videos have set a new trend in extraordinary visual quality and also increased the demand for compressing the huge data of extremely high definition videos. An uncompressed video that is recorded from a video camera occupies large amounts of storage space. High bit rates that result from various types of digital videos make their transmission through their intended channels very difficult. These types of digital videos would require higher bandwidth for their transmission and larger storage space than that is available from conventional storage drivers such as CD-ROM, flash drives, etc. and hence delivering consumer quality video on a compact disc becomes impossible. The high volumes of digital videos data have to be processed retaining the original quality of videos by exploiting the data correlation to reduce redundancy and the limitations of Human Visual System (HVS) to remove irrelevant data. In recent years, major works and research have been done on storage, transmission, processor technology and reduction of amount of data that needs to be stored and transmitted [1]. This reduction of bandwidth has enabled real-time video communication and broadcast of video content.

The evolution of video coding standards started with the growth of International Telecommunication Union (ITU-T) and International Standard Organization/International Electrotechnical Commission (ISO/IEC) standards [1]. Several video coding standards such as H.261 [54] and H.263 [55] developed by ITU- T and the ISO/IEC gave rise to MPEG-1 [56] and MPEG-4 Visual [57]. The joint venture of these two organizations produced H.262/MPEG-2 Video [2] and H.264/MPEG-4 Advanced Video Coding (AVC) [3] standards. The High Efficiency Video Coding (HEVC) is the recent major breakthrough in video coding standards. HEVC is a joint project of ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Moving Picture Experts Group (MPEG) standardization organizations in a partnership called as Joint Collaborative Team- Video Coding

(JCT- VC) [4]. HEVC mainly concentrates on issues with increased video resolution and increased use of parallel processing. Therefore, the state-of-the-art of HEVC is to achieve higher coding efficiency, ease of transport system integration and data loss resilience including applicability of parallel processing architectures [1]. Detailed structure of HEVC is discussed in Chapter 2.

The advancements in coding standards has enabled HDTV signals over satellite and terrestrial transmission systems, video content acquisition and editing systems, Internet and mobile network video, video on demand, video chat and conferencing, etc. Present standards aim for higher coding efficiency of HD and U- HD formats (e.g., 4Kx2K or 8Kx4K resolution [58]) and increase the ability to stream higher quality video to lower bitrate connections at lesser cost. Figure 1.1 shows the growth of video coding standards [52].

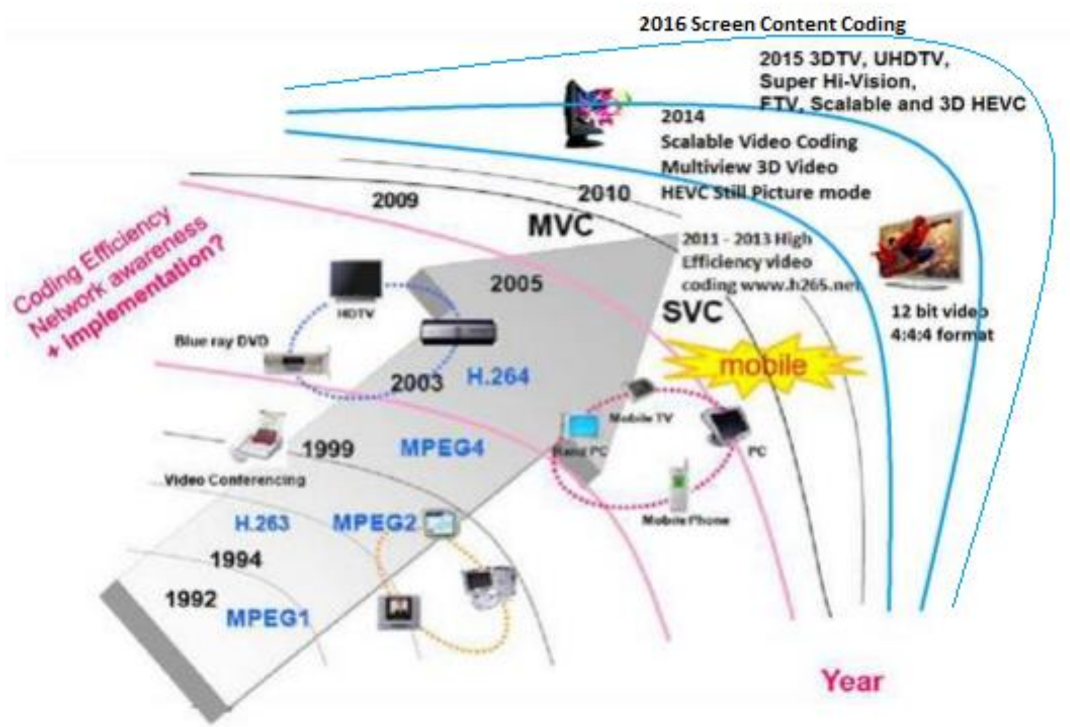


Figure 1.1. Growth and applications of video coding standards [52]

1.2 Scope

This thesis will focus on the implementation of the video coding standard HEVC on screen content at lesser bitrates [22]. The screen content here refers to videos generated by computer such as graphics, animation, text, etc. Compressing a video content in the modern day language is to reduce the information that has to be represented or transmitted in terms of bits and also not reducing the visual quality of the video being used [64].

The video devices that are being used in present day are required to display more than just camera captured images. The video devices are used for remote desktop sharing, demo recordings, automotive displays, tablets, wireless displays, etc. Therefore, compressing such contents becomes necessary for present coding standards to compress efficiently. The difficulties and required techniques to bring out the maximum efficiency of HEVC codec for screen content are discussed.

1.3 Objectives

The goal of this thesis lies on the design, implementation and analysis of a coding technique for screen content keeping HEVC structure as base. The objectives are split into several small objectives for better understanding. The objectives are:

- Examine the nature of screen content videos and understand the difference between a screen content video and a natural video.
- Understand how HEVC performs for screen content videos and learn the coding techniques.
- Learn about the noises and artifacts that may occur when coding for screen content using HEVC.
- Identify the components that cause noises and artifacts while coding screen content using HEVC.
- Study how such components can be modified or improved to get optimal results.

1.4 Thesis Structure

The thesis is organized into six chapters, including the first chapter that enumerates introduction to video coding standards and work flow of the thesis. The rest of the thesis is organized as follows:

- Chapter 2: The chapter contains the brief description of the video coding standard HEVC. The reader is introduced to basic principles, concepts, tools and technologies used in HEVC standard.
- Chapter 3: This chapter starts with differentiating screen content and camera captured content. Further in the chapter, reader is introduced to screen content coding and tools implemented in HEVC for screen content coding
- Chapter 4: The chapter describes the palette based coding and the design to improve the coding efficiency. Later in the chapter the implementation of the proposed scheme is understood.
- Chapter 5: The performance evaluation of the proposed scheme is reported. Firstly the used test conditions and sequences are defined. Later on the performance results obtained are presented and analyzed.
- Chapter 6: The conclusion is based on the work developed and provides suggested future work.

Chapter 2

HIGH EFFICIENCY VIDEO CODING

2.1Overview

The previous video coding standards MPEG-2 Video [2] and H.264/AVC [3] have been the fundamentals of HEVC [1] structure; however, HEVC implements many incremental improvements. The improvements are seen in flexible partitioning, from large to small partition sizes, flexibility in prediction modes and transform block sizes, more sophisticated interpolation and deblocking filters, sophisticated prediction and signaling of modes and motion vectors and features to support efficient parallel processing [59]. HEVC has the capacity to deliver better performances in storing or transmitting video more efficiently than earlier standards such as H.264/AVC [3]. It provides 50% better compression efficiency and supports resolutions up to 8192x4320 [1].

HEVC is based on block-based hybrid coding architecture. The architecture combines motion compensated prediction and transform coding with high-efficiency entropy coding. Figure 2.1 shows the HEVC encoder block diagram [1]. Quad-tree coding block partitioning structure is employed in HEVC which facilitates the use of large and multiple sizes of coding, prediction and transform blocks [6]. HEVC includes advanced intra prediction and coding, adaptive motion vector prediction and coding, a loop filter and an improved version of context-adaptive binary arithmetic coding (CABAC) [40] entropy coding and high level structures for parallel processing. Figure 2.2 shows the HEVC decoder block diagram [12].

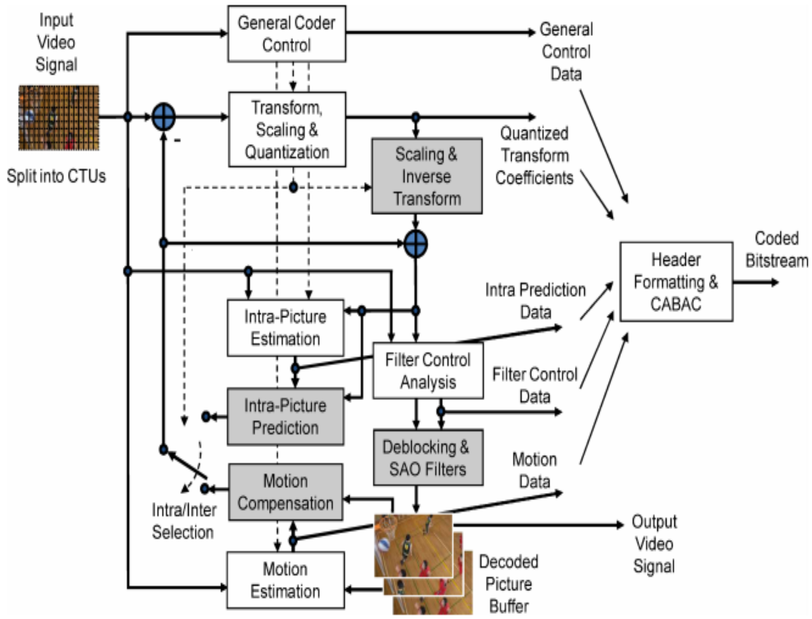


Figure 2.1. Typical HEVC encoder block diagram [1]

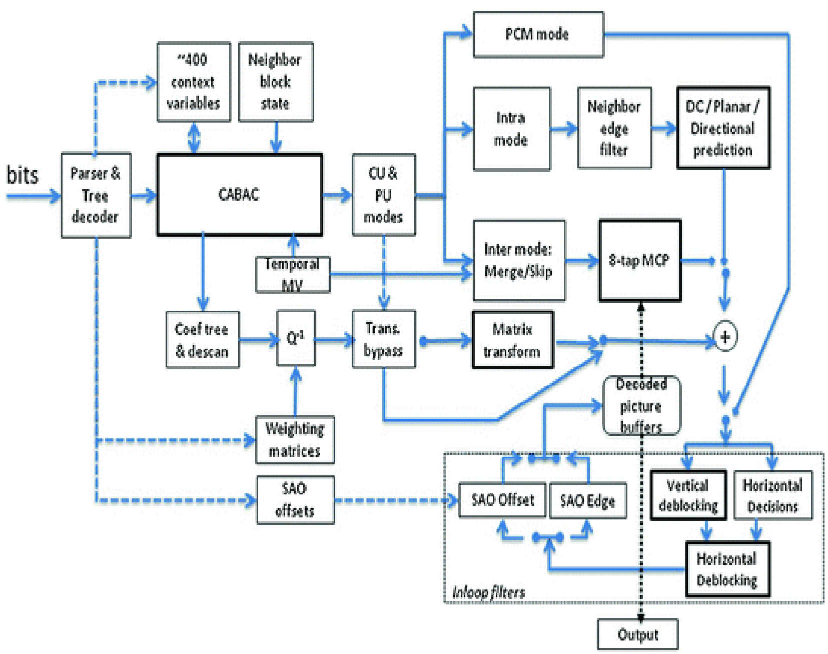


Figure 2.2. Standard decoder block diagram of HEVC [12]

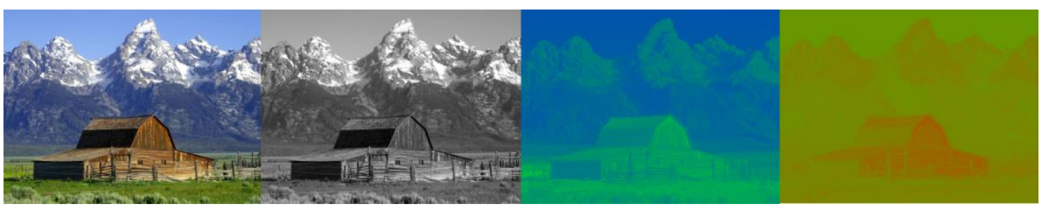
2.2 Color Coding

2.2.1 Color Space

HEVC supports RGB and YUV formats. RGB color model represents color data using red, green and blue components. YUV color model defines color data using a luminance component (Y) and two chrominance components (UV). The term YUV is often used as YCbCr although they are technically distinct. Figures 2.3 and 2.4 represent RGB and YUV for formats respectively, compared with original image [60].

Original Red (R) Green (G) Blue (B)

Figure 2.3. Image in RGB color format [60]



Original Luminance (Y) Chrominance (Cb/U) Chrominance (Cr/V)

Figure 2.4. Image in YUV color format [60]

2.2.2 Chroma Subsampling Types

In chroma subsampling, the chroma information has less information than luma information, due to the fact that human visual system's lower acuity for color differences than for luminance [61]. The common chroma subsampling types are as follows: 4:4:4 denotes no chroma subsampling. 4:2:2 denotes chroma subsampling by a factor of 2 horizontally. 4:2:0 denotes chroma subsampling by a factor of 2 both horizontally and vertically. 4:1:1 denotes chroma subsampling by a factor of 4 horizontally. Figure 2.5 shows the chroma subsampling with YUV color format [62].

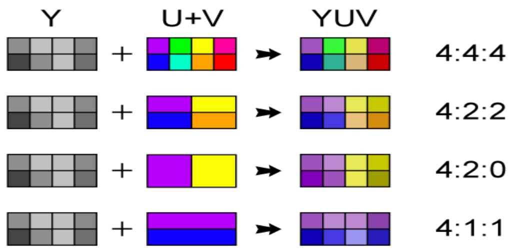


Figure 2.5. Chroma Subsampling [62]

HEVC version 1 supports 4:2:0 8-bit color formats, however, HEVC Range Extension (HEVC- RExt) is designed to support 4:2:2, 4:4:4 and sample bit depth beyond 10-bits per sample. Most of the screen content is captured in the 4:4:4 color format which is not supported by HEVC version 1 [63] [64].

The following subsections provide brief description of key elements that are incorporated in a HEVC encoder structure.

2.3 Picture partitioning

HEVC introduces larger block structures with flexible sub-partitioning mechanism. The basic block is known as the largest coding unit (LCU) also known as macroblock and each macroblock is split into smaller coding units (CUs). CUs are further split into small prediction units (PUs) and transform units (TUs). Figure 2.6 illustrates picture partitioning [20].

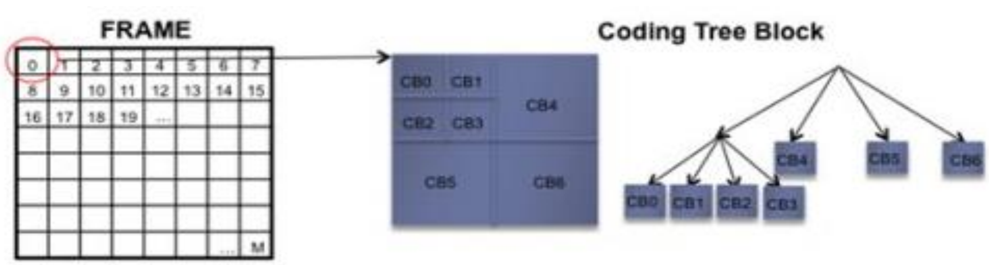


Figure 2.6. Illustration of picture partitioning [20]

2.3.1 Coding Tree Units and Coding Units

Each picture in HEVC is partitioned into coding tree units (CTUs) whose size varies from 16x16, 32x32 to 64x64. The CTU consists of a luma CTB and the corresponding chroma CTBs and syntax elements and the size of a luma CTB can be 16x16, 32x32, or 64x64 samples [1]. Using quadtree partitioning, a CTU can be split into square regions called Coding Units (CUs). The size of CU can vary from 64x64, 32x32, 16x16 to 8x8, depending on the picture content. Context-adaptive coding tree structure is thus included in HEVC to code the recursively quarter- size splitting of the CUs [7]. Figure 8 shows the splitting of a picture into slice, CTU and CUs [59].

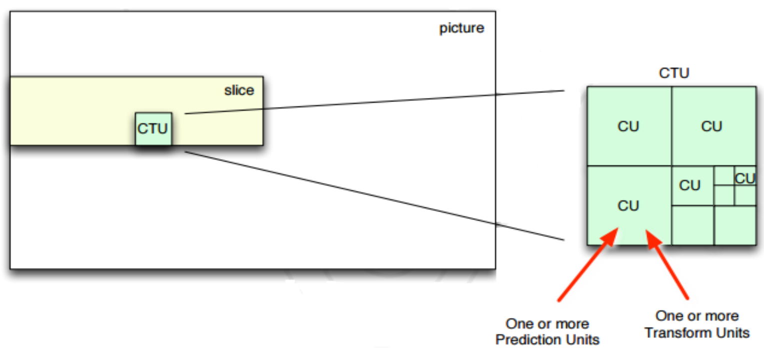


Figure 2.7. Partitioning of picture into slice, CTUs and CUs [59]

2.3.2 Prediction Units

The basis for prediction is prediction units which are formed by splitting each CU into smaller units according to a partition mode. PUs are used in both intra- and inter-prediction. Splitting of CUs (2Nx2N) into PUs can be done only once hence forming PUs of size varying from Nx2N or 2NxN or four PUs of NxN. As a result, PUs are symmetric or asymmetric. PUs can be as large as CUs, depending on the basic prediction-type. Figure 2.8 shows symmetric PUs. Figure 2.9 shows asymmetric PUs. Inter-prediction uses only asymmetric PUs.

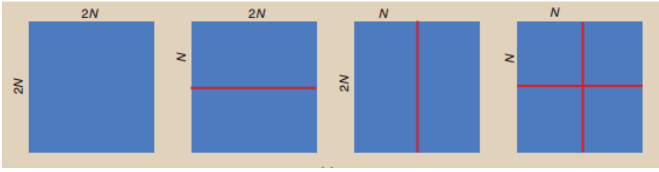


Figure 2.8. Symmetric PUs [8]

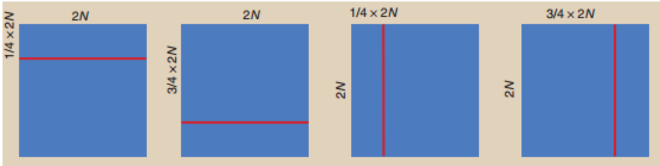


Figure 2.9. Asymmetric PUs [8]

2.3.3 Transform Units

Coding units are recursively divided into quadtree of transform units (TUs), as this is used for residual coding. TUs are also called as residual quadtree [1]. The TUs can be only square shaped and can be 32x32, 16x16, 8x8, 4x4 pixel block sizes. TUs contain coefficients for spatial block transform and quantization and every TU is associated with a transform block (TB) per luma color channel and two chroma color channels.

2.3.4 Slices and Tiles structures

Slices are structures that consist of sequence of CUs and slices in the same picture are independently decodable from one another [6]. The segment of a slice may have one or more slice segments beginning with an independent slice segment followed by subsequent dependent slice segments. Figure 2.10 shows CUs and slices on an image [9].

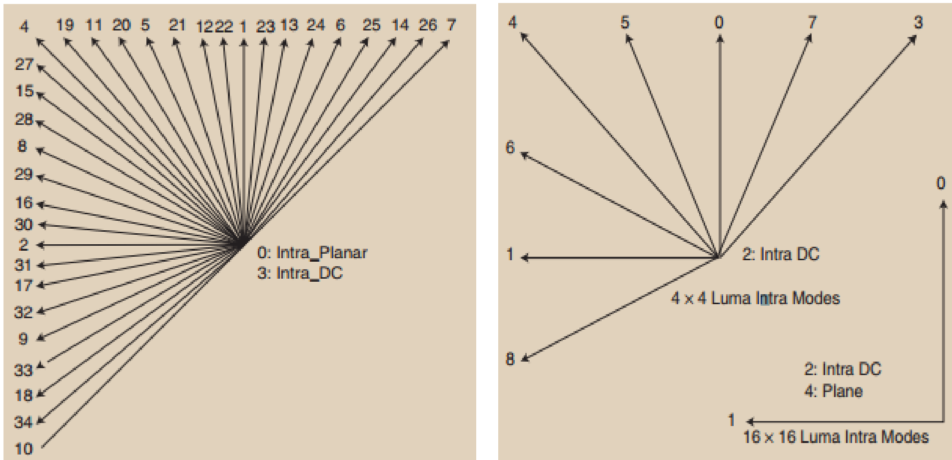
Tiles contain an integer number of CTUs and may contain CTUs present in more than one slice. Tiles are always rectangular and have specified boundaries that divide a picture into rectangular regions [6].

Figure 2.10. Picture showing CUs and Slices [9]

2.3.5 Intra prediction

Using spatial correlation within a picture, HEVC uses block-based intra-picture prediction. HEVC has 35 luma intra-prediction modes providing flexibility compared with nine modes in H.264/AVC. DC mode, planar mode and 33 directional modes are present in HEVC. Intra-prediction can be done at different block sizes, 4x4, 8x8, 16x16 and 32x32 [1]. Figures 2.11 (a) and 2.11 (b) show intra-prediction modes of HEVC and H.264/AVC respectively.

The number of supported modes varies based on PU size. Table 1 shows the different modes used for various PU sizes [13]. Intra mode coding is carried out by forming a 3-entry list of modes. The list is generated using left and above modes. If the desired mode is in the list, the index is sent, otherwise the mode is sent explicitly.



(a) (b)

Figure 2.11. Intra Prediction modes of HEVC and H.264/AVC. (a) HEVC intra-prediction mode [8] (b) H.264/AVC intra-prediction mode [8]

Table 2.1. Supported prediction modes for various PU sizes [13]

Luma intra-prediction modes supported for different PU sizes	
PU size	Intra-prediction modes
4x4	0-16, 34
8x8	0-34
16x16	0-34
32x32	0-34
64x64	0-2, 34

2.3.6 Inter prediction

Inter frame prediction takes the advantage from temporal redundancy between neighboring frames to achieve higher compression ratios. For a block of image samples, motion-compensated prediction is derived. Figure 2.12 shows a block of images with some correlation among the frames [14]. The correlation of motion data of a block with its neighboring blocks is predictively coded based on neighboring motion data. The predictive coding of motion vectors is improved in HEVC by introducing advanced motion vector prediction (AMVP) where the best predictor for each motion block is signaled to the decoder [15].

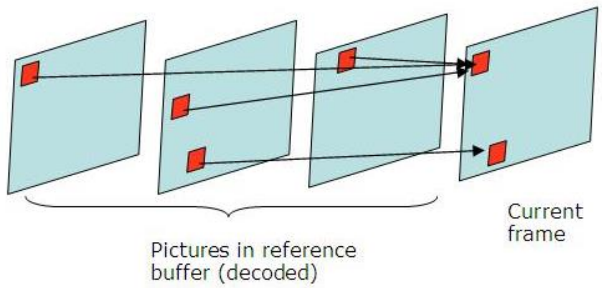


Figure 2.12. Correlation between a block of frames [14]

2.3.7 Transform and quantization

HEVC applies two-dimensional DCT-like integer transform (Discrete Cosine Transform) [39] on the prediction residual. The transforms can be applied to square blocks of size of 4x4, 8x8, 16x16 and 32x32 where smaller size transforms are embedded in larger size transforms and on the rectangular blocks, where the row transform and column transform have different sizes and [6]. A transform related to Discrete Sine Transform (DST) is used in HEVC which is used for intra (4x4) luma blocks (intra-prediction modes). The quantizer structure of H.264/AVC has been the base for HEVC quantizer, in which the quantization parameter (QP) ranges from 0-51 for video sequence of 8-bit depth and it is mapped to a quantizer step size whose value doubles whenever the QP value increases by 6 [15]. Delta QP is the form in which a QP value can be transmitted for a quantization group which can be as small as 8x8 samples. Delta QP is calculated using QP predictor which uses a combination of left, above and previous QP values.

2.3.8 In-Loop filtering

HEVC introduces in-loop filtering to minimize noise and artifacts caused due to lossy compression. There are two types of in-loop filters in HEVC which are deblocking filter and Sample Adaptive Offset (SAO) [6]. To diminish the visibility of blocking artifacts, deblocking filter is used and is applied to samples present at block boundaries. The accuracy of the reconstruction of original signal is improved by SAO and is applied to all samples. Figure 2.13 shows the in-loop filtering method in HEVC decoder [15].

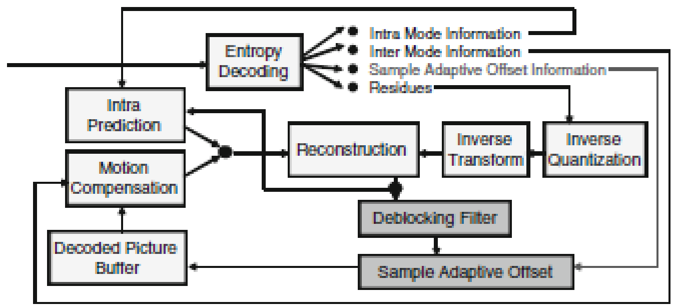


Figure 2.13 HEVC decoder with deblocking filter and SAO [15]

2.3.9 Entropy Coding

The entropy coding is a lossless compression technique which uses the statistical properties of the data to efficiently compress the data and the number of bits which represent the data is logarithmically proportional to the probability of the data, i.e., frequently occurring content of the data will be represented with fewer bits while infrequently occurring data will be represented with many bits [15]. The type of entropy coding used in HEVC is context-adaptive binary arithmetic coding (CABAC) which is also used in H.264/AVC. CABAC provides better compression than most other entropy coding techniques. Figure 2.14 shows the block diagram of CABAC used in coding standard [15]. The main elements of CABAC are binarization, context modeling and binary arithmetic coding.

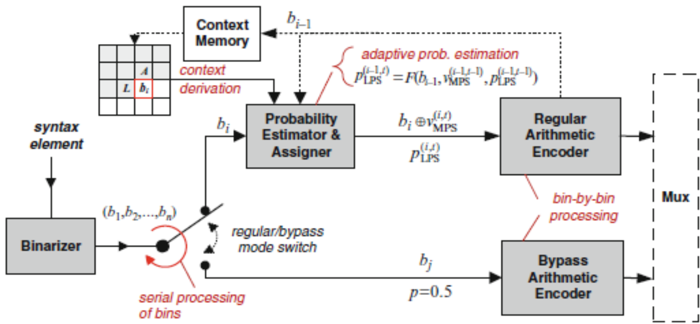


Figure 2.14. Block diagram of CABAC [15]

2.3.10 Parallel processing

Slices and tiles have to be encoded and decoded in parallel in HEVC. HEVC uses multi thread decoder to decode a single picture with threads with two types of tools. One of the tools is tiles which allows the rectangular regions of a picture to be independently encoded and decoded. Tiles allow random access to specific regions of a picture in a video stream. The other tools is Wavefront Parallel Processing (WPP) which allows each slice to be broken into coding units (CUs) and each CU can be decoded based on information from the preceding CU. Wavefront parallel processing is achieved without breaking prediction dependencies and use maximum context in entropy coding. Figure 2.15 illustrates tiles and wavefront parallel processing [20].

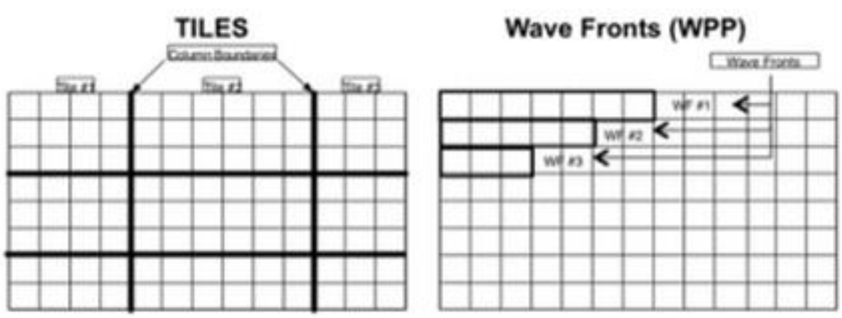


Figure 2.15 Illustration of Tiles and Wave fronts [20]

Chapter 3

SCREEN CONTENT CODING

3.1 Natural Videos v/s Screen Content Videos

The type of video which is captured by a video camera is a natural video content while a video material which consists of computer graphics and camera captured content, video with text overlay, animations and cartoons are all called as screen content or computer generated videos [65]. Figure 3.1 represents camera captured video and Figure 3.2 (a) – 3.2 (d) represents screen content/computer generated videos.

Figure 3.1. Camera captured video content

(a) (b) (c) (d)

Figure 3.2. Images of screen content: (a) slide editing (b) alpha bending (c) video with text overlay (d) mobile display

There are several technical differences between natural video and screen content videos. A camera captured video uses wide range of colors to represent the video content and the values of pixels are close to one another in the content. In screen content videos, the colors that represent the video content are highly saturated or colors are limited in number and therefore, screen content typically has several major colors [22]. Figures 3.3 through 3.6 show difference between camera captured image and screen content image. Figures 3.3 and 3.4 show camera captured and histogram of the image in RGB color format. Figures 3.5 and 3.6 show screen content image and histogram of the image in RGB color format.

Figure 3.3. Image captured in a camera

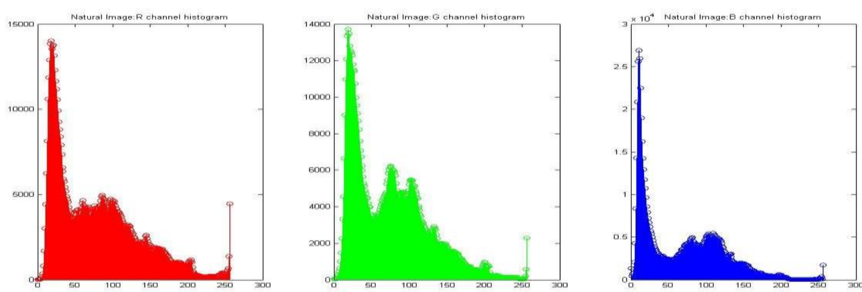


Figure 3.4. Histogram of the camera captured image in RGB color format

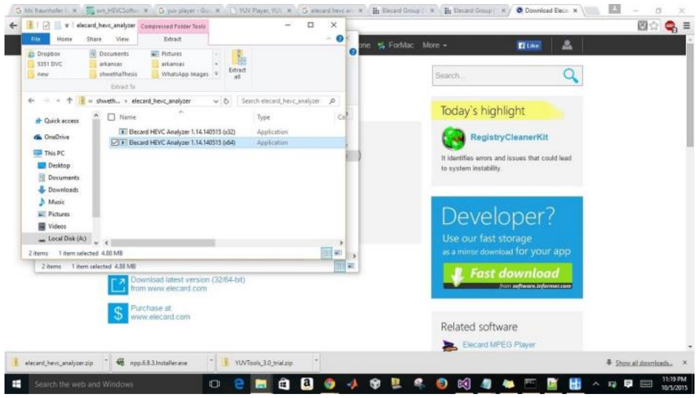


Figure 3.5. Image with screen content (web browsing)

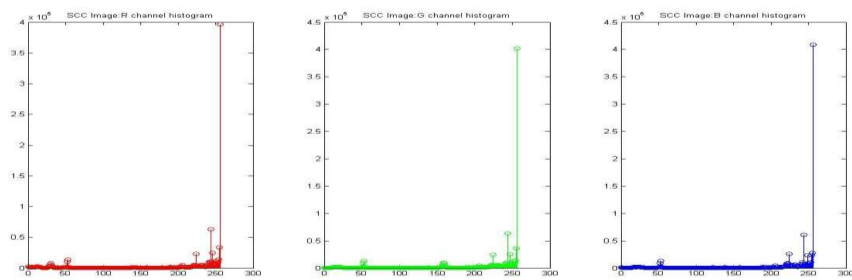


Figure 3.6. Histogram of the screen content image in RGB color format

Screen content has characteristics such as text, shape and graphics and therefore the content is structurally different than camera captured images. Screen content consists of uniformly flat regions and repeated patterns, high contrast and sharp edges, no sensor or capturing noise [64]. Thus properties of screen content demand for a different coding tool other than that is being used for natural videos and coding techniques that are proposed for natural videos cannot provide the best coding efficiency for screen content [22].

3.2 SCC on HEVC framework

The tools and techniques incorporated into HEVC version 1 are mainly based on the coding performances on camera captured content and focuses on applications with 4:2:0 with 8-bit depth video contents. Several applications such as digital video broadcasting, compression of high dynamic range content, screen content coding have content of 4:2:2 or 4:4:4 chroma format and sample bit depth more than 8- bits per sample. Therefore such applications required certain coding and compression efficiency improvements in version 1 of HEVC [63]. The extensions of HEVC version 1 include HEVC Range Extension (HEVC- RExt) and HEVC screen content coding extension (HEVC- SCC) [63], [64]. The highlight of HEVC- RExt is to support 4:2:2 and 4:4:4 chroma formats with 10- bit depth and beyond. The tools added into HEVC- SCC concentrates mainly on coding screen content keeping HEVC version 1 and HEVC- RExt as the foundation.

Early screen content coding techniques that were proposed during the development of HEVC provided considerable compression efficiency of screen content. The Residual Scalar Quantization (RSQ) uses transform skip and directly quantizes the intra prediction residual and Base Colors and Index Map (BCIM) uses only limited number of colors in screen content [66][35]. The intra and inter transform skip modes proposed, completely skip the transform process without changing the HEVC coding structure [36] [37]. Dictionary and Lempel-Ziv coding schemes show the exploitation of repeated patterns in in screen content [67]. Several such techniques lead to an extension of HEVC version 1 to HEVC- SCC to mainly focus on coding screen content more efficiently.

Figure 3.7 shows the encoder block diagram of HEVC- SCC based on HEVC framework [64]. Several changes and new tools are introduced in HEVC- SCC encoder while HEVC- SCC decoder is capable of decoding HEVC version 1 bitstreams, results being identical to HEVC version 1. Figure 3.8 shows the decoder block diagram of HEVC- SCC [80]. The new coding tools incorporated into HEVC- SCC encoder are discussed in the next section.

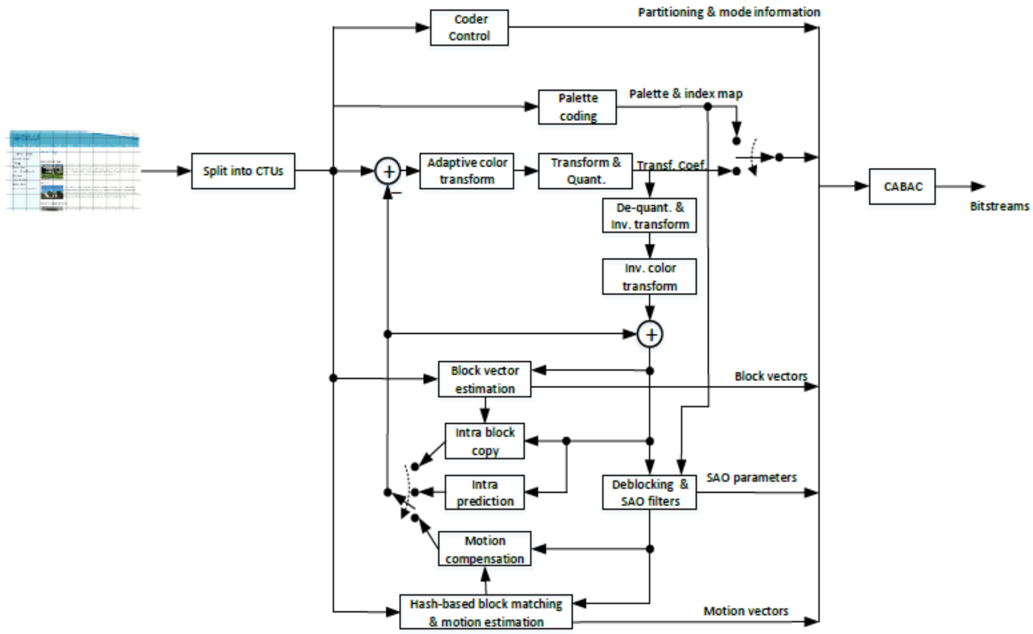


Figure 3.7. Encoder block diagram of HEVC- SCC [64]

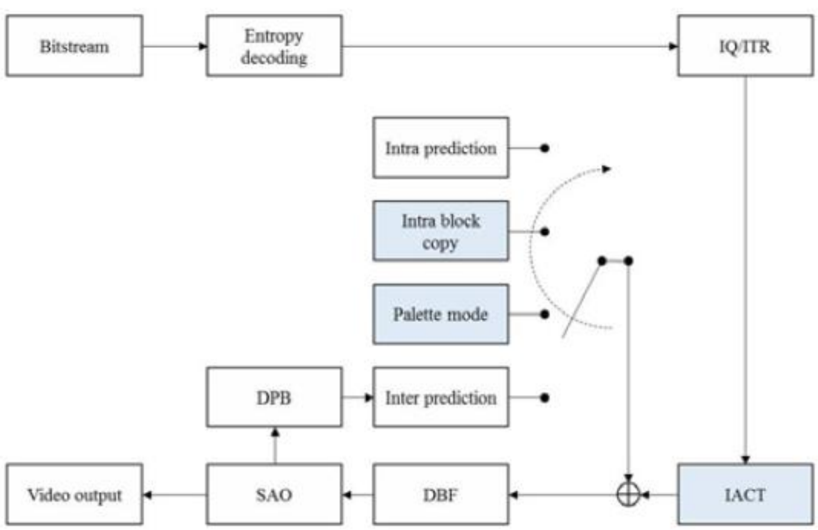


Figure 3.8. Decoder block diagram of HEVC- SCC [80]

3.3 Coding Tools

The important and efficient coding modules are implemented in HEVC- SCC extension. The coding tools are:

3.3.1 Intra block copy

Intra block copy (IBC) mode performs like an inter mode prediction but the PUs of IBC coded CUs predict reconstructed blocks in the same picture. IBC takes the advantage of exploiting the repeated patterns that may appear in screen content. IBC performs inter-like motion compensation within the same block. IBC mode is an additional mode along with intra mode and inter mode. Similar to inter mode, IBC uses block vectors to locate the predictor block [68]. Figure 3.8 shows IBC mode [64]. There are several differences between inter mode and IBC, such as IBC uses current picture as reference if the current picture is not fully decoded.

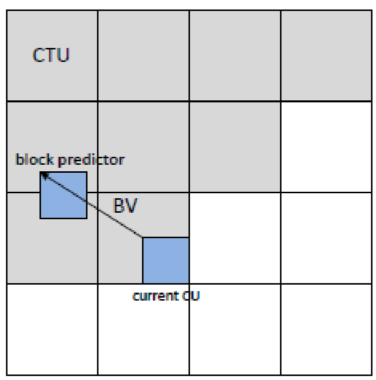


Figure 3.9. Intra block copy prediction in the current picture [64]

3.3.2 Palette mode

Palette mode identifies the limited number of distinct major colors in the screen content and the palette represents the color components and an index corresponding to the color component is signaled in the bit stream. Figure 3.9 shows an input block being divided into major colors and an index map representing the structure of the input block [75]. The detailed description of palette mode and its implementation is discussed in Chapter 4.

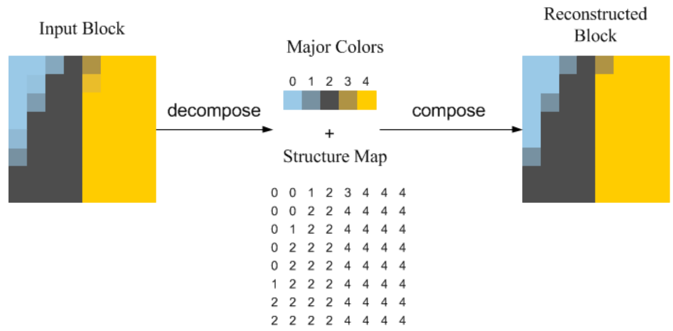


Figure 3.10. Dividing an input block into major colors and index (structure) map [75]

3.3.3 Adaptive color transform

The fundamental idea of adaptive color transform (ACT) is to exploit the inter-color component correlation and reduce the redundancy between the components in RGB/YUV sequences in the 4:4:4 chroma format by enabling the adaptive color- space conversion in every block. The encoding steps before ACT and after ACT are same as in HEVC- RExt. The complexity is reduced by implementing fixed color space transforms; for lossy coding RGB to YCoCg transform is used and for lossless coding lifting-based approximation YCoCg-R to RGB is used [64]. Figure 3.10 shows the ACT implemented in the encoder side consisting of forward and reverse color-space transforms [69]. ACT also implements the concept cross- component prediction to minimize any inter- component redundancy [70].

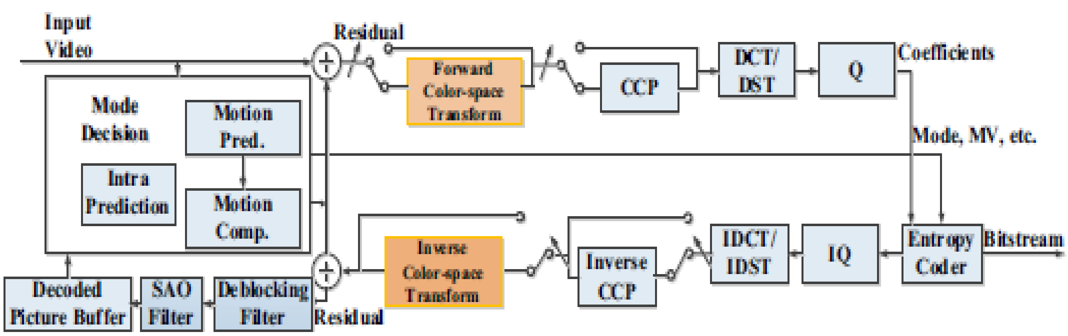


Figure 3.11. Implementation of ACT in encoder side [69]

3.3.4 Adaptive motion vector resolution

Screen content videos have discrete motion or almost aligned motion with sample positions in the picture. Therefore, unlike camera captured content, screen content need not use fractional motion compensation vectors and instead uses integer or full-pixel motion predictor vectors which have only integer values and therefore bits representing fractional values need not be signaled [71].

Chapter 4

PALETTE CODING

4.1 Palette Table Derivation and Coding

Nearly two decades ago a study on color table, now known as palette, was conducted. Since then several modifications were made to get the best outcome of the palette coding technique. The implementation of palette based coding for screen content was introduced in working draft 1 of HEVC-SCC [73]. In this section, palette table generation is discussed and the coding of the table is reviewed.

Each sample in CU can be represented in a set of distinct color values and this set is referred to as palette. Figure 4.1 shows a CU block being represented in palette mode [64].

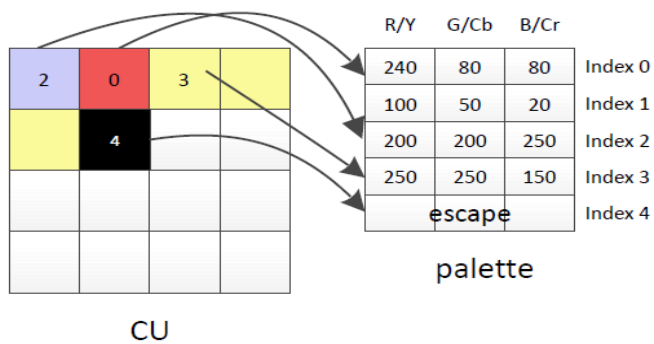


Figure 4.1. CU encoding using palette mode [64]

As seen in Figure 4.1, samples 0,2,3 are palette entries to the palette table and sample 4 is an escape color. The derivation of the palette entries can be lossy or lossless. Depending on the type of coding (lossy/lossless), the palette entries are selected to fit in the palette table.

In lossy coding, the color samples are derived using K-means clustering algorithm, where K is the palette table size. In current CU with N number of pixels, the colors values are denoted as $c=\{c_0, c_1, c_2,...c_{N-1}\}$. The very first sample of the CU is added as the first entry to the palette and following samples undergo sum of absolute distances (SAD) from the current palette entries. If the distortion for each sample is less than a threshold value for the palette entry with respect to the minimum SAD, the sample is added to the cluster belonging to the palette entry or otherwise, the sample is added as the new entry to palette. This method is called as color clustering and divides the colors of N pixels into K(K≤N) sets S={S_0, S_1, S_2, S_{K-1}}. A centroid for a cluster is calculated if the number of samples added to the cluster exceeds a threshold. The following equation (4.1) is used to reduce the within-cluster distortion, where u_i is the centroid of S_i [73].

$$arg\ min \sum_{k=0}^{k-1} \sum_{c \in S_i} ||c - u_i|| \qquad (4.1)$$

Generally the centroid of a cluster becomes the new palette entry for that cluster. However, entries of current palette table can be predicted from the corresponding predictor palette table and predicted entry can be more suitable palette entry than the centroid. To select the most suitable entry to the palette table, rate-distortion analysis is performed. The process is continued until all the clusters are addressed or till the maximum palette size is reached. If the cluster has only one single pixel and corresponding palette entry is not in predictor palette, it is considered as an escape color. Any duplicate palette entries are removed and their clusters are merged.

In lossless coding, a histogram of the samples in the CU is calculated and the histogram is sorted in the decreasing order of the frequency of occurrence of colors . Most frequently occurring histogram entries are placed on top in the palette entry table. If histogram entries are occurring only once and are not present in palette predictor, such histogram entries are converted to escape colors [74].

A palette table predictor is maintained while coding the palette entries. The palette predictor and the maximum palette size is signalled in sequence parameter set (SPS). The palette predictor is initialized using initialization entries signalled in the picture parameter set (PPS) at the beginning of each CTU row, each slice and each tile. After coding CU in palette mode, palette table is updated. Every palette entry in the palette predictor is signalled using a reuse flag to indicate its usage in the current palette. Then the entries that were not used are updated. The palette table is updated untill all the entries in the previous palette predictor that were not used are updated or the maximum palette predictor size is reached. The reuse flags are signalled using run-length coding of zeros and the new entries are signalled using Golomb code of order 0. Figure 4.2 shows the process of updating the predictor palette [64].

predictor palette

Index	G/Y	B/Cb	R/Cr
0	G0	B0	R0
1	G1	B1	R1
2	G2	B2	R2
3	G3	B3	R3
4	G4	B4	R4
5	G5	B5	R5

current palette

Reuse flag	Index	G/Y	B/Cb	R/Cr
1	0	G0	B0	R0
0				
1	1	G2	B2	R2
1	2	G3	B3	R3
0				
0				
	3	G3N	B3N	R3N
	4	G4N	B4N	R4N

Updated predictor palette

Index	G/Y	B/Cb	R/Cr
0	G0	B0	R0
1	G2	B2	R2
2	G3	B3	R3
3	G3N	B3N	R3N
4	G4N	B4N	R4N
5	G1	B1	R1
6	G4	B4	R4
7	G5	B5	R5

Reused palette entries (3)

New palette entries (2), *signalled*

Figure 4.2. Construction of palette table and updating of predictor palette [64]

4.2 Palette Index Map and Coding

After deriving palette table corresponding to each sample, every sample in CU is assigned the index of nearest palette entry. Before coding palette indices, two-dimensional array of the palette indices of palette coded CU needs to be converted into one-dimensional vector. To do this, horizontal- transverse scan and vertical- transverse scan are used, thus increasing the average run length by grouping identical palette indices together. Figure 4.3 shows transverse scan pattern for a palette coded CU [64]. The scan order is explicitly signalled in the bit- stream. In the following description horizontal scan is assumed.

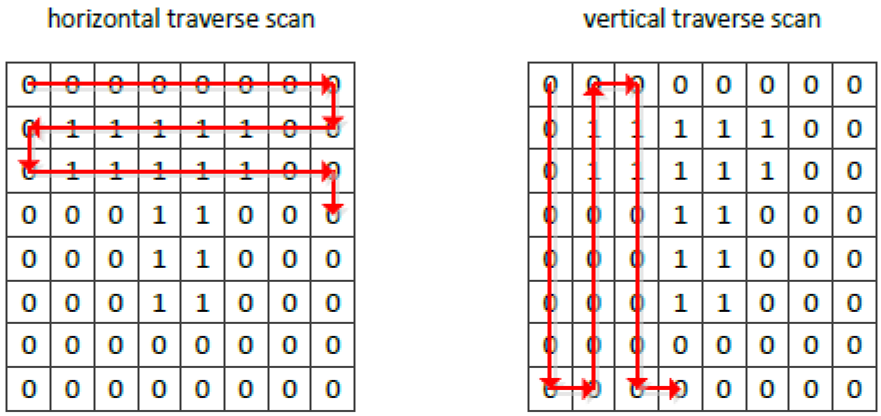


Figure 4.3. Horizontal and Vertical Transverse scan [64]

In screen content, it is observed that consecutive rows and columns exhibit redundancy and to exploit such areas two palette sample modes are used to predictively code, INDEX and COPY_ABOVE modes. Each sample is assigned to either INDEX or COPY_ABOVE mode.

In the COPY_ABOVE mode, the palette index for the current pixel is copied from the palette index of the pixel sample in the row above and only run value is signalled which specifies the number of subsequent samples that are coded using COPY_ABOVE mode. In INDEX mode, palette index is explicitly signalled in the bit stream, followed by run value corresponding to the number of subsequent pixels having same palette index as the current pixel. The mode is signalled using a flag except for the top row or when the previous mode was COPY_ABOVE mode. Figure 4.4 shows the coding of palette indices using COPY_ABOVE and INDEX mode [74].

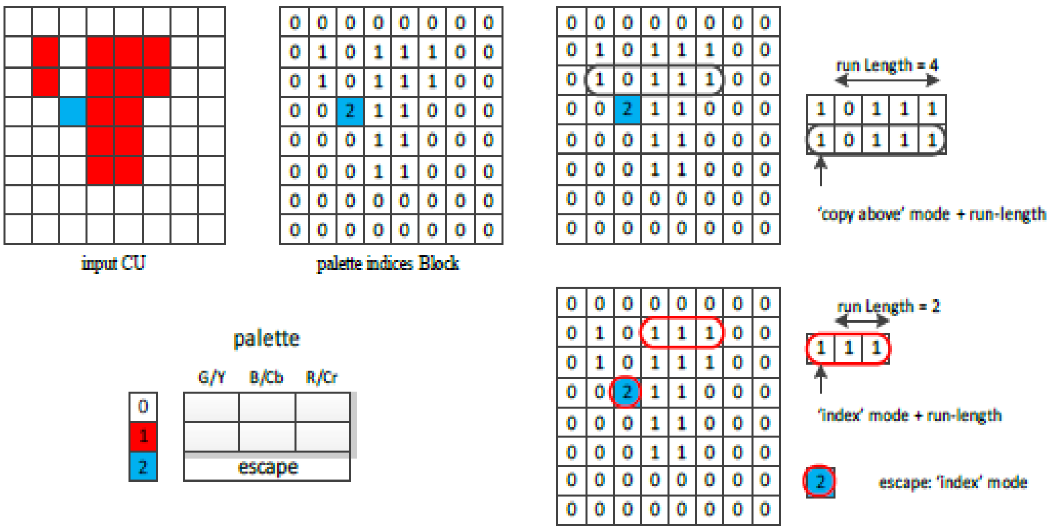


Figure 4.4. Coding of palette indices [74]

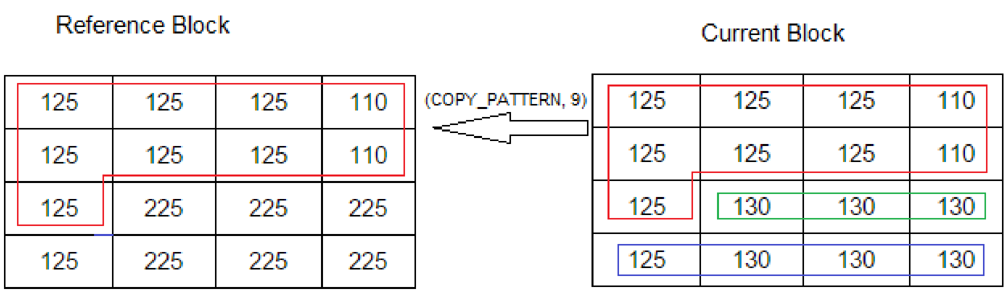
The palette mode uses run-length coding technique for coding of palette indices. The run coding uses a concatenation of unary code and exponential Golomb code of order zero. A run of zero is represented as "0". A run of length (L≥1) is represented as a concatenation of "1" and the exponential Golomb code(order zero) representation for (L-1). Both prefix and suffix are truncated based on the maximum possible run value when the run continues to the end of the block. Table 4.1 shows the binarization process for the palette run value [64].

Table 4.1 Binarization for the palette run value [64]

value	Prefix	Suffix
0	0	-
1	10	-
2-3	110	X
4-7	1110	XX
7-15	11110	XXX
…	…	…

4.3 Proposed Algorithm – New palette mode COPY_PATTERN

In screen content, it can be observed that repeated index map patterns exists in different CU blocks of the picture. While the present coding technique of palette indices exploits the local redundancy of pixels in the same block, there is a potential need for reducing the redundancy of non-local contents in the screen picture. This problem can be approached by introducing a new predictively coding mode, along with INDEX and COPY_ABOVE modes, to address the non-local repeated patterns. The solution to this problem is stated as follows. The illustration of the new coding mode is shown in Figure 4.5.



MODE	PALETTE INDEX	RUN
COPY_PATTERN	-	9
INDEX	130	3
COPY_ABOVE	-	4

Figure 4.5. Illustration of COPY_PATTERN mode.

The proposed algorithm is a combination of palette mode and intra block copy mode. When a CU is being coded in a palette mode, an additional predictive coding mode COPY_PATTERN is introduced. If a sample is chosen to code in COPY_PATTERN mode, it will copy the sample value in the prediction block directly. Like INDEX and COPY_ABOVE mode, a run value is specified corresponding to the number of samples to be copied in COPY_PATTERN mode. The index value is not explicitly signaled in COPY_PATTERN mode. The motion information of the reference block is signalled in the same way as the inter mode with size 2Nx2N and reference block can be generated by motion compensation. The motion information is obtained during inter mode decision and no additional motion estimation burden is required [72]. For this thesis, full frame search algorithm is chosen in the intra block copy. The algorithm flowchart is shown in figure 4.6.

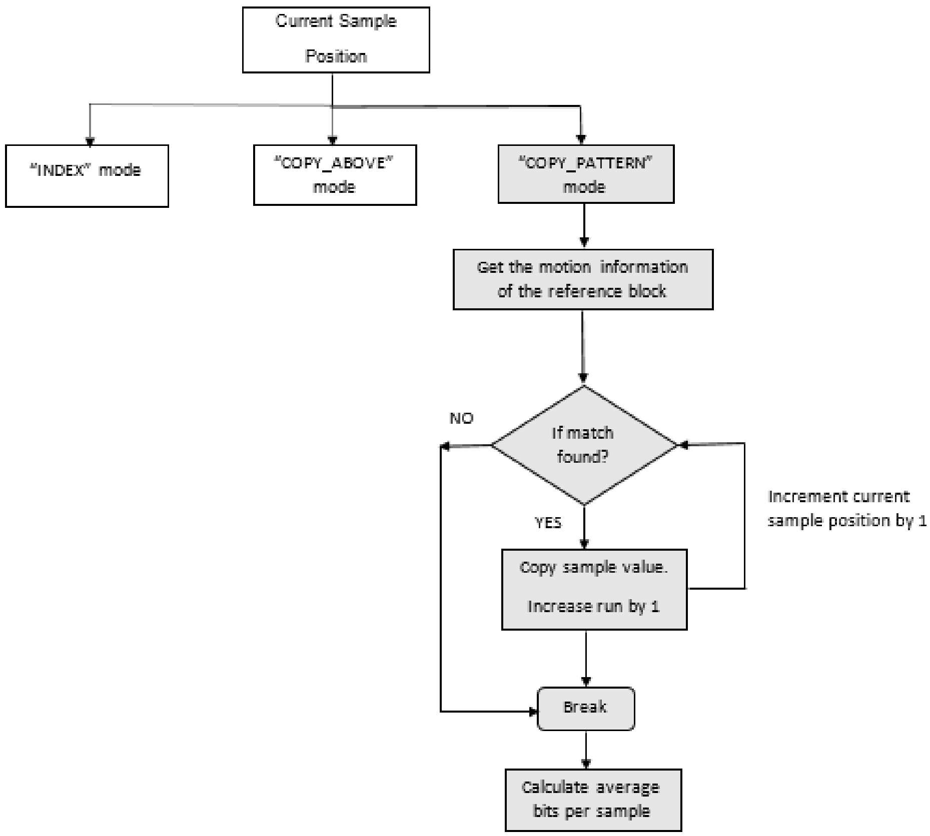


Figure 4.6. Algorithm flowchart. Blocks in grey represent proposed scheme for new palette mode COPY_PATTERN

For each sample, either INDEX or COPY_ABOVE or COPY_PATTERN is possible and the run for each mode is determined. Then the average number of bits per sample position (cost of coding) is calculated which includes the run and index value (if necessary) is calculated. The mode for which the cost is lower is selected, thus giving bit-rate reduction. The mode for which the cost is lower is selected, thus giving bit-rate reduction.

The results of using the new predictive coding mode COPY_PATTERN are shown in the next chapter and analysis is conducted with respect to the existing coding modes.

Chapter 5

RESULTS AND ANALYSIS

5.1 Test Conditions

The common test conditions (CTC) for lossless coding used during the development of HEVC – SCC [78] were used to generate the results for this thesis. Test sequences that were used are screen content videos and are shown in Table 5.1. The test sequences are in YCbCr color format and 4:4:4 chroma sampling format. Two configurations were used, i.e. all intra (AI) and random access (RA) in lossless cost mode.

The work was done using an Intel Core 5 with Microsoft Windows 8.1 64-bit version running with 6 GB RAM at a speed of 2.5GHz.

Simulations were conducted on HEVC reference software Screen Content Model HM 16.4 SCM 4.1 [42] with Visual Studio 2013 [43] and 10 frames were encoded for each test sequence.

Table 5.1 Test sequences used [77]

No.	Sequences	Resolution
1.	pcb_layout	1920X1080 (1080p)
2.	ppt_doc	1920X1080 (1080p)
3.	twist_tunnel	1280X720 (720p)
4.	web_browsing	1280X720 (720p)

5.2 Measuring Quality PSNR

The most commonly used measure for objective quality in reconstructed images is the peak signal-to-noise ratio (PSNR) [45]. The PSNR is defined in 5.1.

$$PSNR = 20\ \log_{10}\frac{MAX^2}{MSE} \quad (5.1)$$

Here MAX is the maximum pixel value in the image, i.e. MAX = 2^B - 1, where B is the amount of bits per pixel. MSE is the Mean Square Error, defined in 5.2.

$$MSE = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[I(i,j) - K(i,j)]^2 \qquad (5.2)$$

Here I and K represents the original and reconstructed image respectively. M and N represent the height and width of the image. PSNR is measured in decibels (dB). We will use the PSNR quality metric to quantify the resulting visual quality we achieve when encoding. By studying the PSNR and the number of bits used, we can get a good idea of how well our encoder performs.

5.2 Implementation Results

To evaluate intra and inter compression efficiency between proposed and original algorithm, “encoder_intra_main_scc” and “encoder_randomaccess_main_scc” configurations are used in HM 16.4 SCM 4.1 respectively. Lossless coding is carried out. Tables 5.2 and 5.3 show bitrates (Kbit/s) and encoding time (s) for 1080p and 720p resolution test sequenecs. Figures 5.1 and 5.2 show average bitrate savings % plot for all intra and random access profiles.

Table 5.2 Results of proposed algorithm vs. original for all intra mode

ALL INTRA					
	Anchor (SCM4.1)+Proposed		Anchor (SCM4.1)		
Test Sequences	Bitrate (kbps)	Encoding Time (s)	Bitrate (kbps)	Encoding Time (s)	Bitrate savings %
pcb_layout	8750.780	991.228	8990.928	926.382	-2.671
ppt_doc	20539.234	918.051	21348.336	827.073	-3.790
twist_tunnel	20821.718	655.940	21157.488	619.981	-1.587
web_browsing	24845.408	325.850	25377.576	312.417	-2.097

Table 5.3 Results of proposed algorithm vs. original for random access mode

RANDOM ACCESS					
	Anchor (SCM4.1)+Proposed		Anchor (SCM4.1)		
Test Sequences	Bitrate (kbps)	Encoding Time (s)	Bitrate (kbps)	Encoding Time (s)	Bitrate savings %
pcb_layout	897.846	287.749	917.040	283.497	-2.093
ppt_doc	2304.690	333.224	2368.912	325.732	-2.711
twist_tunnel	18691.261	3616.270	18849.408	3392.373	-0.839
web_browsing	3011.996	167.154	3067.800	159.498	-1.819

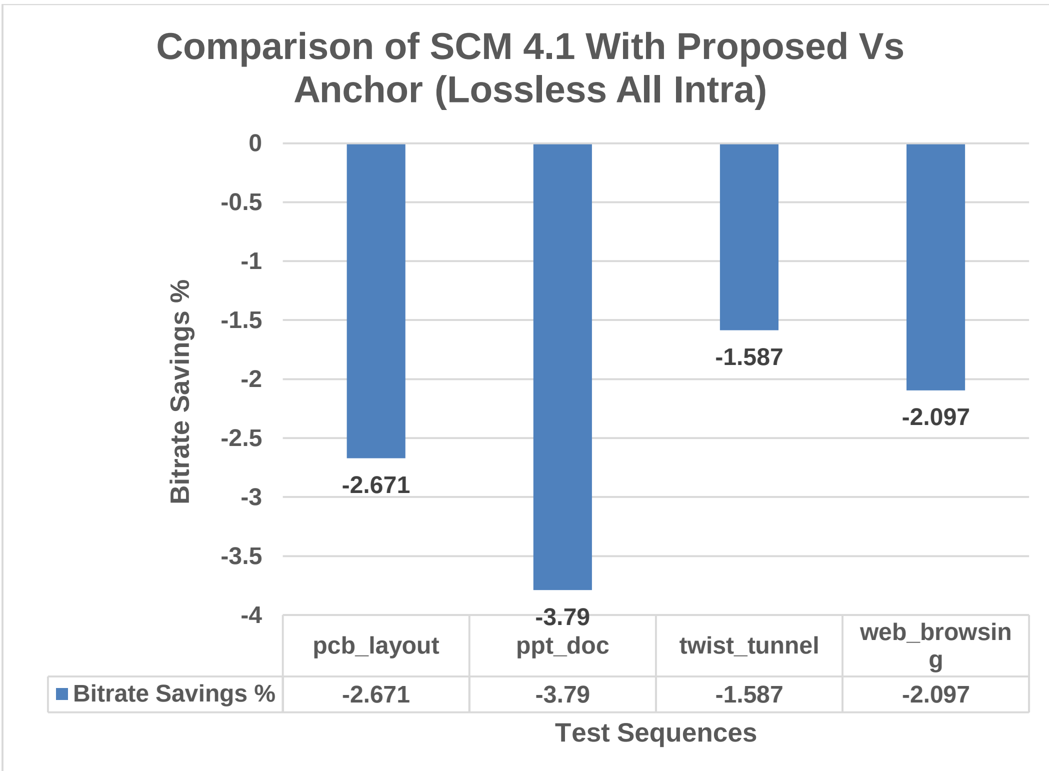


Figure 5.1 Average bitrate savings for Proposed vs. Anchor under all intra profile.

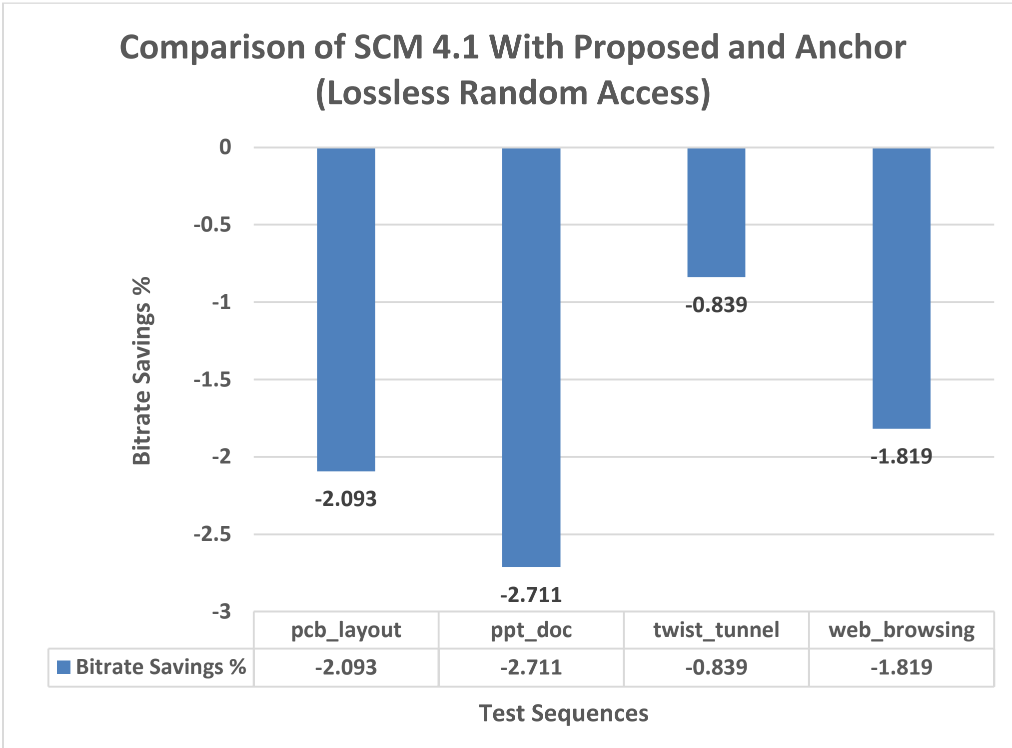


Figure 5.2 Average bitrate savings for Proposed and Anchor under random access profile.

5.5 Coding Tools of SCM

The tools of SCM, intra block copy (IBC), palette mode and adaptive color transform (ACT) are analyzed in this section. Tables 5.4, 5.5 and 5.6 show the coding performance compared to anchor not having the specific coding tool in SCM-4.1, for YUV 4:4:4 sequences. Bitrates (kbps), encoding time (s) and respective bitrate savings are shown in these tables. Figures 5.3 – 5.5 show bitrate saving achieved by each coding tool in SCM-4.1. Figure 5.6 shows the comparison of encoding time (s) of coding tools.

Table 5.4 Comparison of SCM-4.1 with versus without IBC

Comparison of SCM4.1 With Vs. Without Intra Block Copy (IBC) (Lossless All Intra)					
	Anchor (SCM 4.1)		Anchor Without IBC		
Test Sequences	Bitrate (kbps)	Encoding Time (s)	Bitrate (kbps)	Encoding Time (s)	Bitrate savings %
pcb_layout	8990.928	926.382	11358.544	529.210	-20.844
ppt_doc	21348.336	827.073	30411.216	602.687	-29.800
twist_tunnel	21157.488	619.981	22290.024	278.520	-5.081
web_browsing	25377.576	312.417	30220.848	244.794	-16.03

Table 5.5 Comparison of SCM-4.1 with versus without palette mode

Comparison of SCM4.1 With Vs. Without Palette Mode (Lossless All Intra)					
	Anchor (SCM 4.1)		Anchor Without Palette Mode		
Test Sequences	Bitrate (kbps)	Encoding Time (s)	Bitrate (kbps)	Encoding Time (s)	Bitrate savings %
pcb_layout	8990.928	926.382	20724.800	920.093	-56.660
ppt_doc	21348.336	827.073	36322.176	839.570	-41.230
twist_tunnel	21157.488	619.981	27239.712	606.834	-22.328
web_browsing	25377.576	312.417	29075.928	289.147	-12.72

Table 5.6 Comparison of SCM-4.1 with versus without ACT

Comparison of SCM4.1 With Vs. Without Adaptive Color Transform (ACT) (Lossless All Intra)					
	Anchor (SCM 4.1)		Anchor Without ACT		
Test Sequences	Bitrate (kbps)	Encoding Time (s)	Bitrate (kbps)	Encoding Time (s)	Bitrate savings %
pcb_layout	8990.928	926.382	8992.240	795.296	-0.014
ppt_doc	21348.336	827.073	21339.728	675.995	+0.04
twist_tunnel	21157.488	619.981	21161.184	586.161	-0.020
web_browsing	25377.576	312.417	25373.232	254.698	+0.017

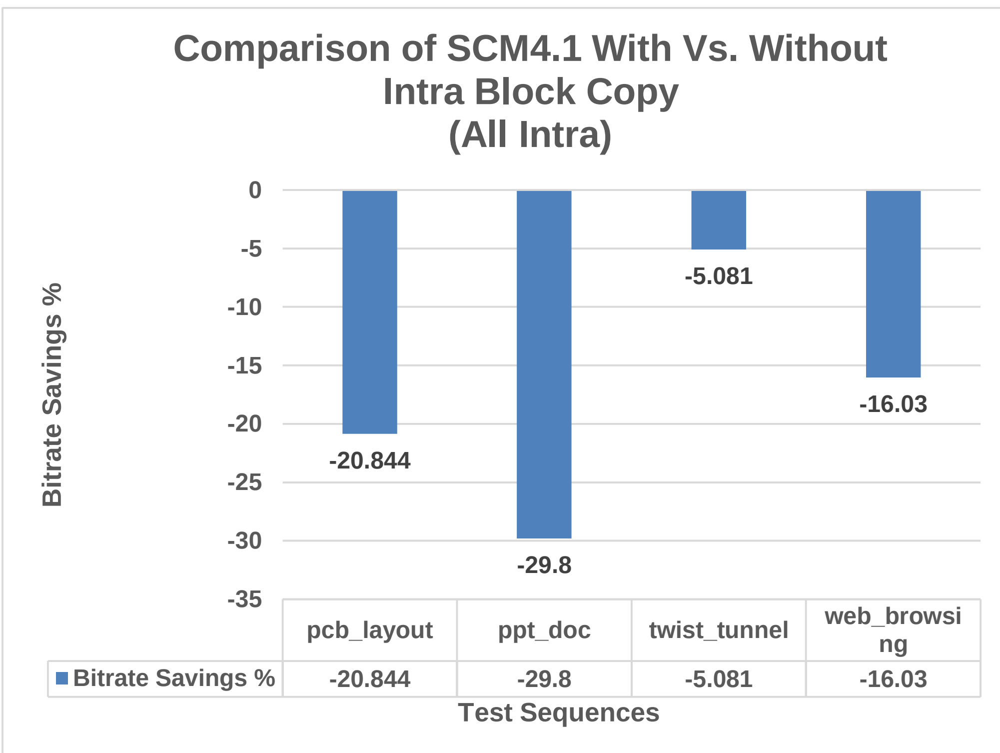


Figure 5.3 Bitrate savings achieved by IBC

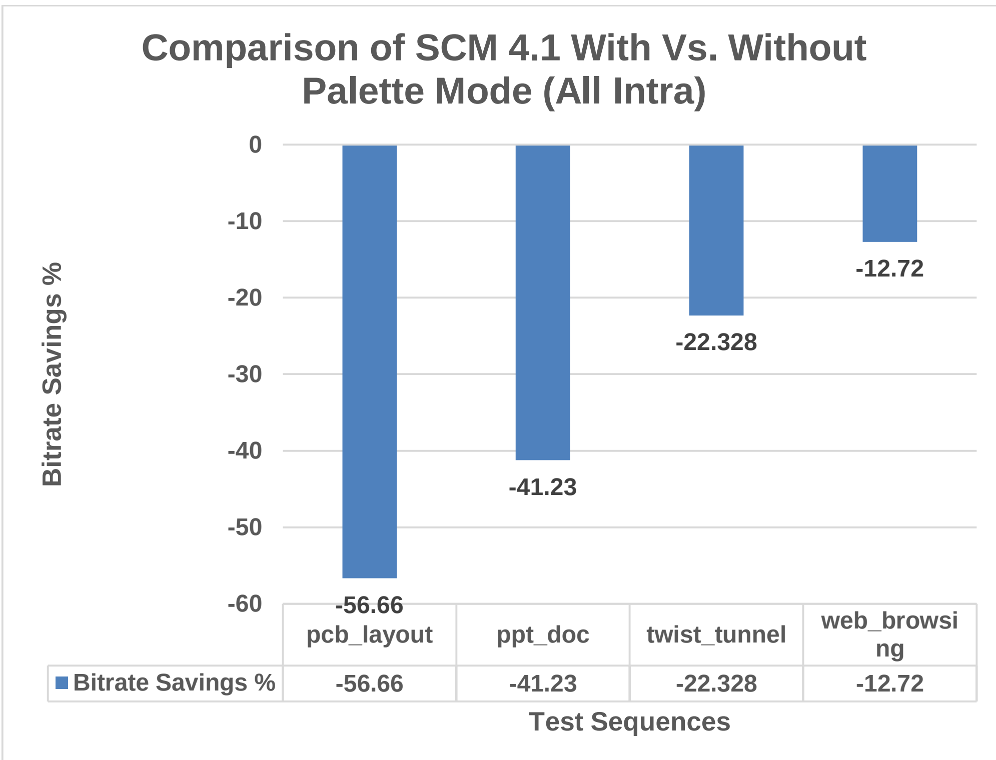


Figure 5.4 Bitrate savings achieved by palette mode

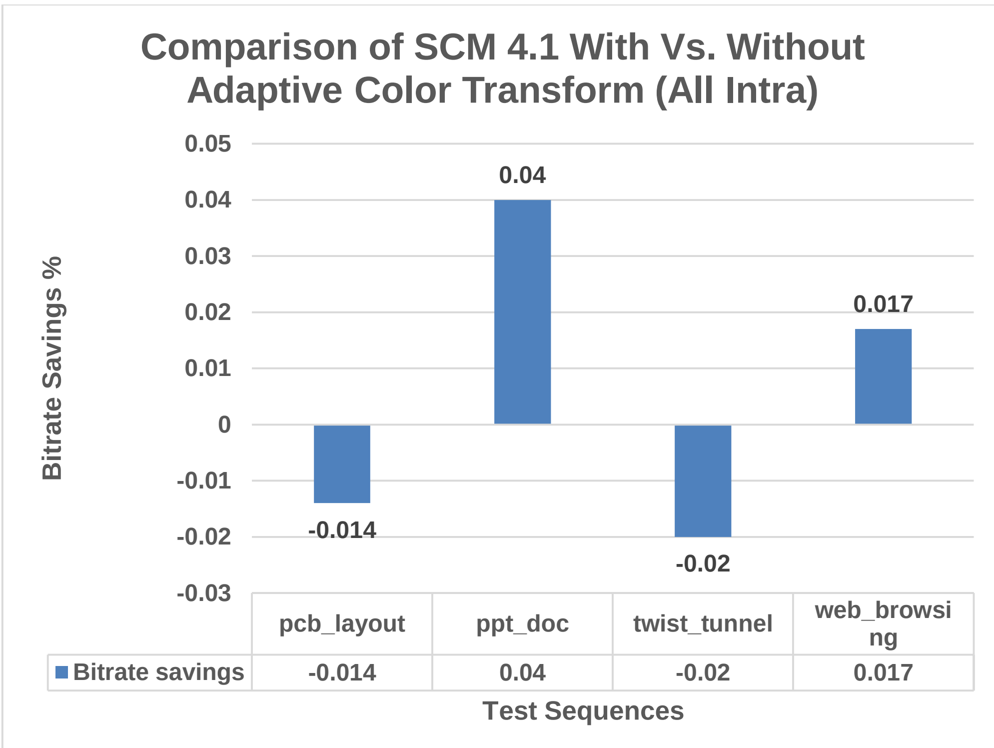


Figure 5.5 Bitrate savings achieved by ACT

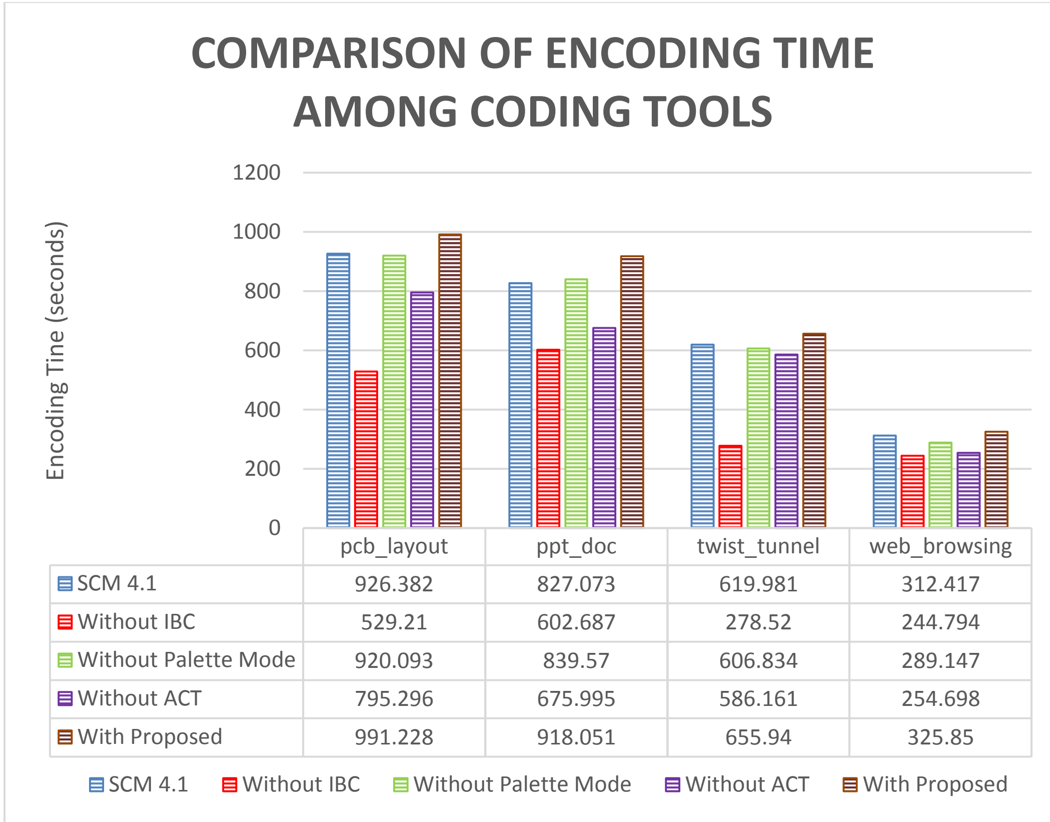


Figure 5.6 Encoding time (s) of coding tools

5.6 Summary

Simulation results show that the proposed algorithm for palette mode coding can achieve 1.5% - 3.7% bitrate savings under AI and 0.8% - 2.7% bitrate savings under RA for lossless condition, respectively. The increase in encoding time (s) is observed and methods to reduce encoding time are discussed in future work. From Figures 5.1, 5.3, 5.4 and 5.5, average bitrate savings given by proposed method are 2.53%, IBC is 17.94%, palette mode is 33.23% and ACT gives negligible bitrate savings under lossless all intra profile for YUV- format sequences.

Chapter 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

A detailed study of Screen Content coding technique “Palette Mode” is conducted. A new coding technique for coding palette index map, COPY_PATTERN, is proposed and implemented on HM reference software HM16.4 for Screen Content Coding SCM 4.1 [42]. The present coding modes for palette index map deals with local correlation of the pixels and reduces local pixel redundancy, which act as the first stage for coding the index map. The proposed method exploits the non-local pixel correlation and is the second stage of coding the palette index map. The combination of coding techniques to exploit local and non-local sample correlation contributes to the multi-stage prediction coding scheme of palette index map. The implementation of the proposed algorithm provides average bitrate savings of 2.5% for all intra and 1.86% for random access profiles in lossless mode.

6.2 Future Work

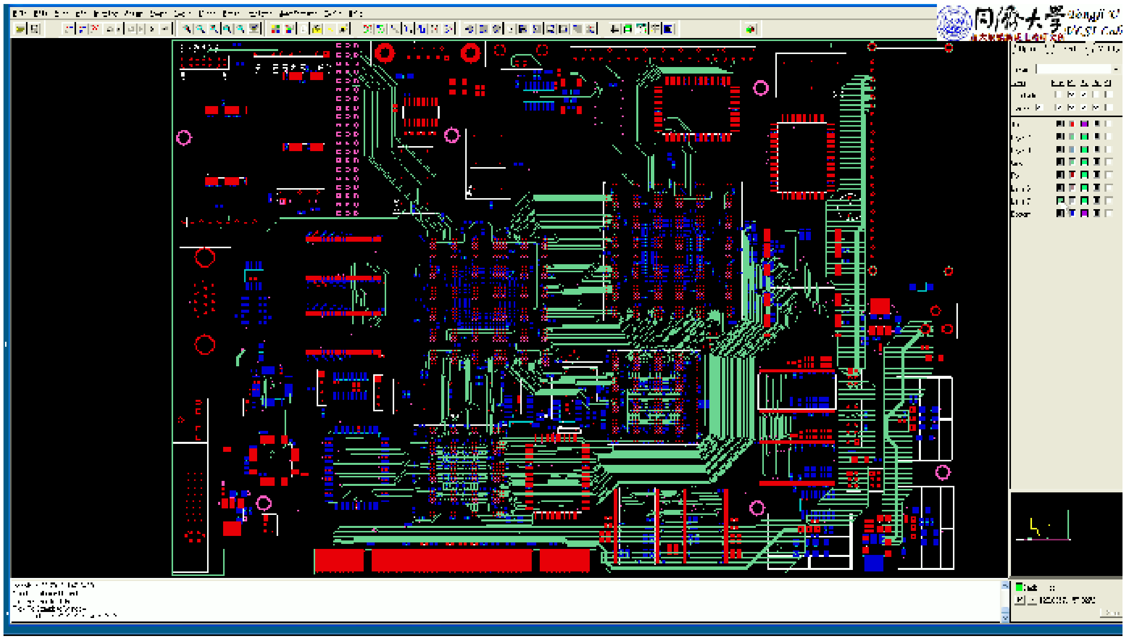
Coding of screen content videos with coding tools of SCM increase the complexity of the encoder and hence increases the encoding time. Based on this observation, the future work can be on reducing the encoding time by parallelizing the independent methods on the encoder side. Also implementation of fast algorithms for screen content coding tools can help in decreasing the encoding time.

The coding performance of HEVC on coding surveillance camera videos opens the door for further improvements. The data volume of surveillance sequences is huge since the surveillance cameras keep shooting all the time. Besides, surveillance sequences need to be preserved for a long period for reviewing. So, it is important to pay attention to the coding of surveillance video sequences [76].

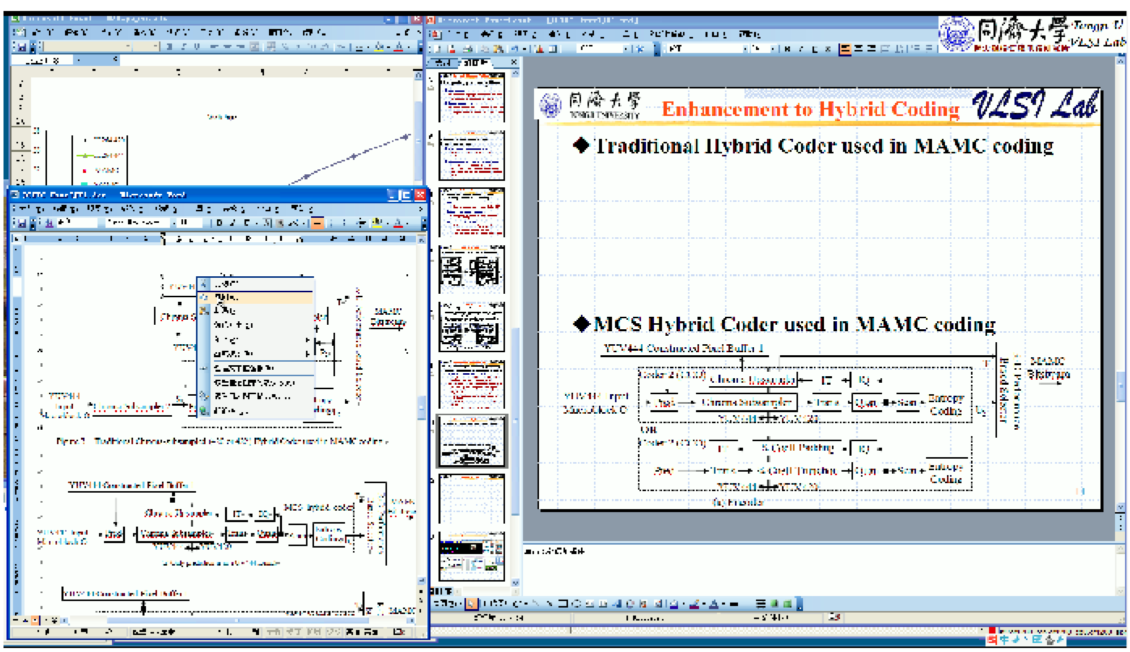
Appendix A

Test Sequences [77]

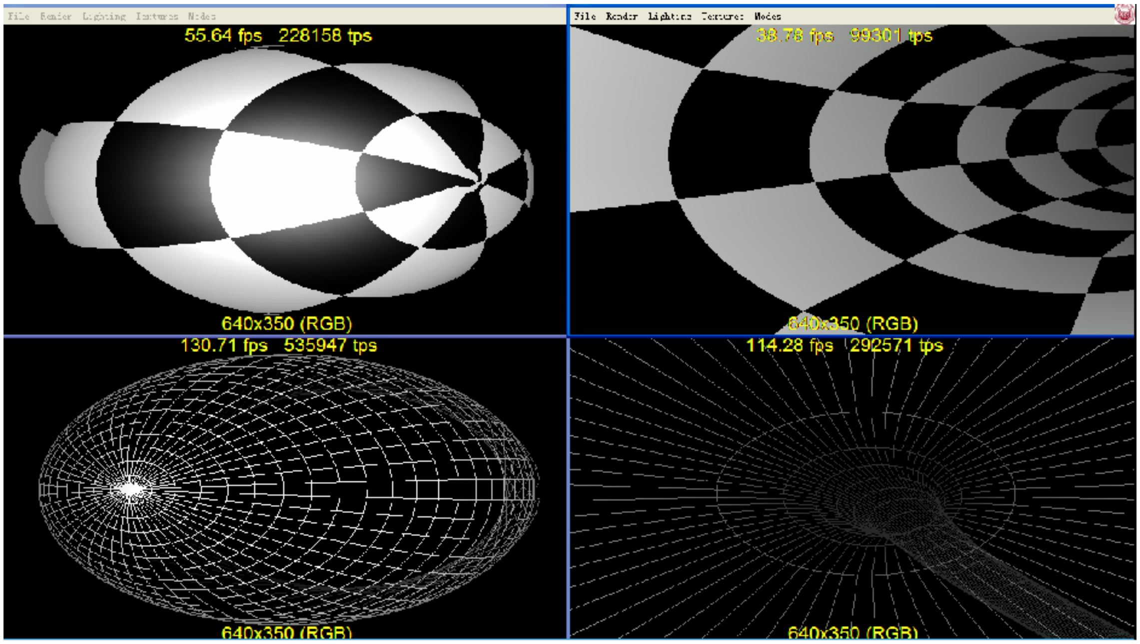
A1. PCB Layout (Resolution: 1920x1080)



A2. PPT Document (Resolution: 1920x1080)



A3. CG Twist Tunnel - Animation (Resolution: 1280x720)



A5. Web Browsing (Resolution: 1280x720)

Appendix B

Acronyms

ACT - Adaptive Color Transform

AI – All Intra

AVC – Advanced Video Coding

AMVP- Advanced Motion Vector Prediction

BCIM - Base Colors and Index Map

BV – Block Vector

CABAC - Context Adaptive Binary Arithmetic Coding

CAVLC - Context Adaptive Variable Length Coding

CTU- Coding Tree Unit

CU - Coding Unit

DBF- Deblocking Filter

dB - DeciBels

DFT – Discrete Fourier Transform

DCT – Discrete Cosine Transform

DST – Discrete Sine Transform

DPB - Decoded Picture Buffer

IBC - Intra Block Copy

FDIS - Final Draft International Standard

HD - High Definition

HEVC - High Efficiency Video Coding

HTB - High Throughput Binarization

IACT – Inverse Adaptive Color Transform

ITU-T - International Telecommunication Union (Telecommunication Standardization Sector)

IEC - International Electrotechnical Commission

ISO – International Standards Organization

JBIG - Joint Binary Image Experts Group

JPEG - Joint Photographic Experts Group

JCT-VC - Joint Collaborative Team on Video Coding

LCU - Larger Coding Unit

MPEG - Moving Picture Experts Group

MRC - Mixed Raster Content

MSE – Mean Square Error

PSNR – Peak Signal-to-Noise Ratio

PPS – Picture Parameter Set

PU – Prediction Unit

RA – Random Access

RSQ - Residual Scalar Quantization

SAO - Sample Adaptive Offset

SCC - Screen Content Coding

SPS – Sequence Parameter Set

TU - Transform Units

UHD - Ultra-High-Definition

VCEG – Video Coding Experts Group

VCL - Variable Code Length

WPP - Wavefront Parallel Processing

References

[1] G.J. Sullivan et al, "Overview of the High Efficiency Video Coding (HEVC) Standard," IEEE Transactions on Circuits and Systems for Video Technology, vol.22, no.12, pp.1649-1668, Dec. 2012

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

[3] Advanced Video Coding for Generic Audio-Visual Services, ITU-T Rec. H.264 and ISO/IEC 14496-10 (AVC), ITU-T and ISO/IEC JTC 1, May 2003

[4] B. Bross, et al, High Efficiency Video Coding (HEVC) Text Specification Draft 9, document JCTVC-K1003, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Oct. 2012

[5] K.R. Rao, D. N. Kim and J. J. Hwang, "Current video coding standards: H.264/AVC, Dirac, AVS China and VC-1," 42nd Southeastern Symposium on System Theory (SSST), pp. 1-8, 7-9 March 2010

[6] K. McCann, et al, “High Efficiency Video Coding (HEVC) Encoder Description v16 (HM16)”, JCTVC-R1002, July 2014

[7] B. Bross, Y. Wang and T. Wiegand; High Efficiency Video Coding (HEVC) text specification draft 10 (For FDIS & Final Call), JCT-VC, Doc. JCTVC-L1003, Geneva, Switzerland, Jan. 2013

[8] M.T. Pourazad, et al, "HEVC: The New Gold Standard for Video Compression: How Does HEVC Compare with H.264/AVC?" IEEE Consumer Electronics Magazine, vol.1, no.3, pp.36-46, July 2012

[9] Article on HEVC: http://www.vcodex.com/images/uploaded/342512928230717.pdf

[10] ZTE communications; Special Topic on Introduction to the High Efficiency Video Coding Standard, P. Wu and M. Li, Vol. 10 No. 2, June 2012:

http://wwwen.zte.com.cn/endata/magazine/ztecommunications/2012/2/201207/P020120711312919366435.pdf

[11] K.-L. Hua, et al, "Inter Frame Video Compression With Large Dictionaries of Tilings: Algorithms for Tiling Selection and Entropy Coding," IEEE Transactions on Circuits and Systems for Video Technology, vol.22, no.8, pp.1136-1149, Aug. 2012

[12] HM Software manual for version 15: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/branches/HM-15-dev/doc/software-manual.pdf

[13] B. Bross, et al, “High efficiency video coding (HEVC) text specification draft 6,” JCTVC-H1003, Feb. 2012

[14] Article on Inter frame coding: http://en.wikipedia.org/wiki/Inter_frame

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

[16] A. BenHajyoussef and T. Ezzedine, “Analysis of Residual data for High Efficiency Video Coding (HEVC)”, IJCSNS International Journal of Computer Science and Network Security, Vol. 12, No. 8, pp. 90-93, Aug. 2012

[17] V. Sze and M. Budagavi, "High Throughput CABAC Entropy Coding in HEVC," IEEE Transactions on Circuits and Systems for Video Technology, vol.22, no.12, pp.1778-1791, Dec. 2012

[18] C. C. Chi, et al, "Improving the parallelization efficiency of HEVC decoding," 19th IEEE International Conference on Image Processing (ICIP), pp.213-216, Sept. 30, 2012-Oct. 3, 2012

[19] M. Alvarez-Mesa, et al, "Parallel video decoding in the emerging HEVC standard," IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp.1545-1548, 25-30 Mar. 2012

[20] Article in Design & Reuse, “Development of a 4K 10-Bits HEVC Encoder”: http://www.design-reuse.com/articles/33379/4k-10-bits-hevc-encoder-development.html

[21] G.J. Sullivan, et al, "Standardized Extensions of High Efficiency Video Coding (HEVC)," IEEE Journal of Selected Topics in Signal Processing, vol.7, no.6, pp.1001-1016, Dec. 2013

[22] W. Zhu, et al, "Screen Content Coding Based on HEVC Framework," IEEE Transactions on Multimedia, vol.16, no.5, pp.1316-1326, Aug. 2014

[23] W. Zhu, et al, "Compound image compression by multi-stage prediction," IEEE Visual Communications and Image Processing (VCIP), pp.1-6, Nov. 2012

[24] R. de Queiroz, R. Buckley and M. Xu, “Mixed raster content (MRC) model for compound image compression,” in Proc. IS&T/SPIE Symp. on Electronic Imaging, Visual Communications and Image Processing, vol.3653, pp. 1106–1117, 1999

[25] W. Pennebaker and J. Mitchell “JPEG Still Image Data Compression Standard” Norwell, MA, USA: Kluwer, 1993

[26] A. Skodras, C. Christopoulos and T. Ebrahimi, “The JPEG 2000 still image compression standard,” IEEE Signal Process Mag., vol. 18, no.5, pp. 36–58, Sep. 2001

[27] H. Hampel, et al "Technical features of the JBIG standard for progressive bi-level image compression", Signal Process.: Image Communication Journal, Vol. 4, no. 2, pp. 103-111, Apr. 1992

[28] W. Ding et al, “Block-based fast compression for compound images,” in Proc. 2006 IEEE International Conference on Multimedia and Expo, pp. 809–812, July 2006

[29] Access to HM 16.1 Reference Software: http://hevc.hhi.fraunhofer.de/

[30] Z. Ma et al, "Advanced Screen Content Coding Using Color Table and Index Map," IEEE Transactions on Image Processing, vol.23, no.10, pp.4399-4412, Oct. 2014

[31] G. Braeckman, et al, "Lossy-to-lossless screen content coding using an HEVC base-layer," 18th International Conference on Digital Signal Processing (DSP), pp. 1- 6, 1-3 July 2013

[32] G. Braeckman, et al, "Visually lossless screen content coding using HEVC base-layer," Visual Communications and Image Processing (VCIP), pp. 1- 6, 17-20 Nov. 2013

[33] H. Yang et al, "Subjective quality assessment of Screen Content Images," Sixth International Workshop on Quality of Multimedia Experience (QoMEX), pp. 257-262, 18-20 Sept. 2014

[34] M. Budagavi and D. Kwon, "Intra motion compensation and entropy coding improvements for HEVC screen content coding," IEEE Picture Coding Symposium (PCS), pp.365-368, 8-11 Dec. 2013

[35] C. Lan et al, "Intra and inter coding tools for screen contents," in JCTVC-E145, pp. 16–23, Geneva, Switzerland, Mar. 2011.

[36] C. Lan et al, "Intra transform skipping," in JCTVC-I0408, Geneva, Switzerland, Apr. 2012

[37] X. Peng et al, "Inter transform skipping," in JCTVC-J0237, Geneva, Switzerland, Apr. 2012

[38] W. Zhu, J. Xu, and W. Ding, "Screen content coding with multi-stage base color and index map representation," JCTVC-M0330. Incheon, Korea, Apr. 2013.

[39] N. Ahmed, T. Natarajan and K. R. Rao, "Discrete cosine transform," IEEE Transactions on Computers., vol. C-23, no. 1, pp. 90–93, Jan. 1974

[40] D. Marpe, H. Schwarz and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no. 7, pp. 620–636, July 2003

[41] Special issue on Screen Content Video Coding and Applications, IEEE Journal on Emerging and Selected Topics in Circuits and Systems (JETCAS), Final manuscripts due on 22nd July 2016

[42] Access to HM 16.4 SCM 4.1 Reference Software: http://hevc.hhi.fraunhofer.de/

[43] Visual studio 2013: http://www.dreamspark.com

[44] Tortoise SVN: http://tortoisesvn.net/downloads.html

[45] ITU-T. Objective perceptual multimedia video quality measurement in the presence of a full reference. Recommendation ITU-T J.247, page 15, August 2010.

[46] G. Bjontegaard, "Calculation of Average PSNR Differences between RD-curves," VCEG-M33, 13th meeting: Austin, Texas, USA, Apr. 2001.

[47] C. Lan, G. Shi, and F. Wu, “Compress compound images in h.264/mpge-4 avc by exploiting spatial correlation,” IEEE Transactions on Image Processing, vol. 19, no. 4, pp. 946–957, Apr. 2010.

[48] Introduction to video compression: http://www.videsignline.com/howto/185301351;jsessionid=B2FYP22SRT0TEQSNDLOSKHSCJUNN2JVN?pgno=1

[49] Joint Collaborative Team on Video Coding Information web: http://www.itu.int/en/ITU-T/studygroups/2013-2016/16/Pages/video/jctvc.aspx

[50] Camera captured video sequence download link: http://media.xiph.org/video/derf/

[51] Link to download software manual for HEVC: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/

[52] Multimedia processing course website: http://www.uta.edu/faculty/krrao/dip/

[53] 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

[54] Video Codec for Audiovisual Services at px64 kbit/s, ITU-T Rec. H.261, version 1: Nov. 1990, version 2: Mar. 1993.

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

[56] Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1.5 Mbit/s—Part 2: Video, ISO/IEC 11172-2 (MPEG-1), ISO/IEC JTC 1, 1993.

[57] Coding of Audio-Visual Objects—Part 2: Visual, ISO/IEC 14496-2 (MPEG-4 Visual version 1), ISO/IEC JTC 1, Apr. 1999 (and subsequent editions).

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

[59] I. E. G. Richardson, “HEVC: An introduction to High Efficiency Video Coding”, www.VCodex.com, 2013

[60] Original images of Barn in color, RGB and YUV formats: https://commons.wikimedia.org/wiki/File:Barn_grand_tetons_rgb_separation.jpg#filelinks

[61] S. Winkler, C. J. van den B. Lambrecht, and M. Kunt (Editors), "Chapter 10: Vision and Video: Models and Applications", Vision models and applications to image and video processing, Springer, p. 209, July 2001.

[62] Article on 4k and 8k videos: http://www.shutterangle.com/2014/shooting-4k-video-for-2k-delivery-bitdepth-advantage/

[63] D. Flynn, et al, “Overview of the Range Extensions for the HEVC Standard: Tools, Profiles and Performance”, IEEE Transactions on Circuits and Systems for Video Technology, pp. 1-15, (Early Access).

[64] J. Xu, R. Joshi and R. A. Cohen, “Overview of the Emerging HEVC Screen Content Coding Extension”, IEEE Transactions on Circuits and Systems for Video Technology, pp. 1-14, (Early Access).

[65] H. Yu, K. McCann, R. Cohen, and P. Amon, “Requirements for an extension of HEVC for coding of screen content,” ISO/IEC JTC1/SC 29/WG 11 Requirements subgroup, San Jose, California, USA, document MPEG2014/N14174, Jan. 2014.

[66] C. Lan, J. Xu, F. Wu, and G. J. Sullivan, “Screen content coding,” 2nd JCT-VC meeting, Geneva, Switzerland, document JCTVC-B084, July 2010.

[67] T. Lin, et al, “Mixed chroma sampling rate high efficiency video coding for full-chroma screen content,” IEEE Transactions on Circuits Systems for Video Technology, vol. 23, no. 1, pp. 173–185, Jan 2013.

[68] B. Li and J. Xu, “Non-SCCE1: Unification of intra BC and inter modes,” 18th JCT-VC meeting, Sapporo, Japan, document JCTVC-R0100, July 2014.

[69] L. Zhang, et al, “Adaptive color-space transform for HEVC screen content coding,” IEEE Data Compression Conference (DCC), 2015, Snowbird, Utah, USA, Apr. 2015.

[70] A. Khairat, et al, "Adaptive cross-component prediction for 4:4:4 high efficiency video coding," IEEE International Conference on Image Processing (ICIP), pp. 3734-3738, Paris, France, Oct 2014.

[71] B. Li, et al, "Adaptive motion vector resolution for screen content," 19th JCT-VC meeting, Strasbourg, France, document JCT-VC-S0085, Oct. 2014.

[72] W. Zhu, et al, "Palette coding with inter-prediction," Document of Joint Collaborative Team on Video Coding, JCT-VC-U0064, June 2015.

[73] X. Xiu, et al, "Palette-based Coding in the Screen Content Coding Extension of the HEVC Standard", IEEE Data Compression Conference (DCC), Apr. 2015.

[74] R. Joshi, et al, "Screen content coding test model 4 (SCM 4)", Document of Joint Collaborative Team on Video Coding, JCT-VC-T1014, Feb. 2015.

[75] W. Ding, Y. Lu, and F. Wu, "Enable efficient compound image compression in H.264/AVC intra coding", IEEE International Conference on Image Processing. IEEE, vol. 2, pp. II–337, Oct. 2007.

[76] H. Zhang, et al, "Surveillance sequences for video coding development", Document of Joint Collaborative Team on Video Coding, JCT-VC-V0092, Oct. 2015.

[77] Link to Screen content Test Sequences: http://pan.baidu.com/share/link?shareid=3128894651&uk=889443731

[78] H. Yu, et al, "Common test conditions for screen content coding," 20th JCT-VC meeting, Geneva, Switzerland, document JCT-VC-T1015, Feb. 2015.

[79] Access to Joint Collaborative Team on Video Coding (JCT-VC) documents: http://phenix.int-evry.fr/jct/

[80] Y. Ahn, et al, "Analysis of Screen Content Coding Based on HEVC", IEIE Transactions on Smart Processing and Computing, vol. 4, no. 4,pp- 231-236, Aug, 2015.

[81] V. Sze, M. Budagavi and G.J. Sullivan (Editors), "High efficiency video coding: Algorithms and architectures", Springer 2014.

[82] M. Wien, "High Efficiency Video Coding: Coding Tools and Specification", Springer, 2014.

[83] I.E.G. Richardson, “Coding video: A practical guide to HEVC and beyond,” Wiley, 11 April 2016.

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

Biographical Information

Nagashree Mundgemane was born in Karnataka, India in 1990. She received her Bachelor's degree in Electronics and Communication from Visvesvaraya Technological University, Karnataka in 2013. She joined The University of Texas at Arlington to pursue her Master's degree in Fall 2013. She has worked in Multimedia Processing Lab under Dr. Rao from Fall 2014 to Fall 2015. After graduation, she plans to pursue her career in the fields of Signal Processing and Communications to make the best use of the knowledge acquired.