INTRA PREDICTION STRATEGIES FOR LOSSLESS COMPRESSION IN HIGH EFFICIENCY VIDEO CODING

Thesis

Submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

by

SHILPA KAMATH S.

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING,
NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA,
SURATHKAL, MANGALURU - 575025, INDIA

NOVEMBER, 2020

DECLARATION

I hereby ***declare*** that the Research Thesis entitled **INTRA PREDICTION STRATEGIES FOR LOSSLESS COMPRESSION IN HIGH EFFICIENCY VIDEO CODING** which is being submitted to the ***National Institute of Technology Karnataka, Surathkal*** in partial fulfillment of the requirement for the award of the Degree of ***Doctor of Philosophy*** in **Department of Electronics and Communication Engineering** is a ***bonafide report of the research work carried out by me***. The material contained in this Research Thesis has not been submitted to any University or Institution for the award of any degree.

Shilpa Kamath S.
Reg. No. 155001EC15F09
Department of E&C Engineering
NITK, Surathkal - 575025

Place: NITK-Surathkal
Date: 06/11/2020

CERTIFICATE

This is to certify that the Research Thesis entitled **INTRA PREDICTION STRATEGIES FOR LOSSLESS COMPRESSION IN HIGH EFFICIENCY VIDEO CODING** submitted by **SHILPA KAMATH S.** (Register Number: 155001EC15F09) as the record of the research work carried out by her, is accepted as the *Research Thesis submission* in partial fulfillment of the requirements for the award of degree of **Doctor of Philosophy**.

Dr. Aparna P.
Research Guide
Assistant Professor
Department of E&C Engineering
NITK, Surathkal - 575025

Chairman-DRPC
(Signature with Date and Seal)

Dedicated To

My Beloved Family

Acknowledgements

The possible outcome of any research work depends much on the quality of education received, quality of the teachers, research resources and encouraging environment. Pursuing the doctoral program at NITK, Surathkal renders all these aforementioned benefits that have resulted in the possible successful outcome of the research work carried out. At this point in time, I take immense pleasure to owe my sincere gratitude and thankfulness to the following people for helping me to carry out the research work successfully.

I express profound gratitude to my research supervisor Dr. Aparna P., Assistant Professor, Department of E&C Engineering, for her valuable guidance. She has always been very supportive, helpful, patient and motivating during the tenure of work. I exhibit my thankfulness to Dr. Laxminidhi T., HOD, Department of E&C Engineering for his encouragement and administrative support. I also thank Dr. M. S. Bhat and Dr. Sripathi U., former Heads, Department of E&C Engineering for their motivation.

The acknowledgement will be incomplete without expressing my appreciation to Dr. Abhilash Antony, Associate Professor, Department of E&C Engineering, MITS, Kerala for his relentless support, guidance, encouragement, and patience. His timely bits of advice, suggestions, and solutions to the bottleneck problems encountered during the work is simply immeasurable.

I am deeply indebted to my doctoral committee member Dr. Jidesh P. for his insightful comments and valuable suggestions put across during the oral presentations. A special thanks to all the faculty members, staff and peer research scholars at NITK for making things go easy through their support and/or motivation. My heartfelt thanks to Asha C. S. for her priceless inputs and support in the need of the hour. I take this opportunity to thank Roshni, Amitha, Tilak and Rathish as well, for their timely support.

Before concluding, I delightfully acknowledge the co-operation and moral support that I have received from my parents, husband, family members and pet Champion. I thank them especially for having faith in my decisions throughout this journey.

Shilpa Kamath S.

Abstract

HEVC, an abbreviation for high efficiency video coding, is a digital coding standard for videos developed by the JCT-VC committee to address the bandwidth and storage space requirements, associated while handling the high definition multimedia content. Sophisticated coding tools and mechanisms are deployed into the framework making it far more superior to its predecessor standard H.264 both in terms of compression efficiency and quality of reconstruction, but at the cost of increased complexity. This thesis is mainly based on the sample-based intra prediction strategies to improve the prediction accuracy that results in the enhancement of the compression efficiency, for the lossless mode of HEVC. Lossless coding becomes imperative in certain applications like video analytics, video surveillance, etc., that mandate distortion-free data reconstruction.

The main focus of the thesis is to mitigate the spatial redundancy persistent due to coherence, smoothness, illumination and shadowing effects in the natural video sequences. These issues can also challenge another class of multimedia content commonly referred to as the screen content (SC) sequences as a result of certain peculiarities which they exhibit. Therefore, the prediction generation stage of the CODEC plays a significant role in an attempt to minimize the entropy using the superior intra prediction strategies. The gradient dependent predictor, context-based predictor, and the improvised blend of predictors which is based on the penalizing factor modify the intra prediction mechanisms of HEVC to emerge as the highlight of this thesis. The overall algorithmic performance is evaluated by deriving the savings in bit-rate and run-time. Additionally, the comparisons made with the several state-of-the-art prediction techniques reveal that significant improvements in coding gains with reasonable computational complexity and at par savings in run-time is attained using the proposed methods. The algorithmic modifications are embedded into the HEVC reference software provided by the JCT-VC and its validation is performed using the HEVC test sequences along with another class of natural sequences referred to as the Class 4K.

Keywords: HEVC; Lossless; Compression; Intra; Prediction; Redundancy.

Contents

List of Figures

List of Tables

Nomenclature

SYMBOL	MEANING
$\lvert . \rvert$	Absolute operator
$\lfloor . \rfloor$	Floor operator
$>>$	Right bit-shift operator
$\sum$	Summation operator
β	Internal bit-depth
μ	Mean
σ^2	Variance
θ_{disp}	Displacement parameter
min	Minimum value
max	Maximum value
C_{Gmin}	Causal pixel corresponding to minimum gradient value
$C_{x,y}$	Current/Target pixel
N	Pixel to north of $C_{x,y}$
W	Pixel to west of $C_{x,y}$
NE	Pixel to north-east of $C_{x,y}$
NW	Pixel to north-west of $C_{x,y}$
C	Array of causal pixels
$C[i]$	Causal pixel corresponding to i^{th} direction
D_i	Sum of absolute differences between the patches $P_h^g[m]$ and $P_h^g[n]$ in the i^{th} direction
$fract$	Distance measured at (1/32)nd pixel accuracy between R_i and the projection
$G_{pf,i}$	Penalizing factor
G_H	Gradient in horizontal direction
G_V	Gradient in vertical direction
G_D	Gradient in diagonal direction
G_{AD}	Gradient in anti-diagonal direction
G_i	Gradient in the i^{th} direction
G_{min}	Minimum gradient value
P_{DC}	Block DC value

$\hat{P}[x, y]$	Generated prediction
$\hat{P}_h[x, y]$	Horizontal prediction
$\hat{P}_v[x, y]$	Vertical prediction
$P_h^g[m]$	Patch of pixels around the pixel $h[m]$
R_{above}	Reference samples from the row on top of the target block
R_{left}	Reference samples from the column on left of the target block
$R_{samples}$	$4L$+1 reference samples representing R_{above} and R_{left}
$R_{1,array}$	1D array of reference samples formed using $R_{samples}$
$\hat{R}[x, y]$	Generalized notation for a reference sample at location $[x, y]$
R_i and R_{i+1}	Reference samples essential to generate the prediction
$S(C_{x,y})$	1D array to hold the 4-bit binary pattern
$SAD(P_h^g[m], P_h^g[n])$	Sum of absolute differences between the patches $P_h^g[m]$ and $P_h^g[n]$
wt_m	Weighted factor corresponding to the m^{th} causal neighbor

Abbreviations

AI	All Intra
AVC	Advanced Video Coding
B	Bi-predictive
CABAC	Context-based Adaptive Binary Arithmetic Coding
CALIC	Context-based, Adaptive, Lossless Image CODEC
CB	Coding Block
CD	Compact Disc
CfP	Call for Proposals
CIP	Combined Intra Prediction
CODEC	enCOder/DECoder
CTB	Coding Tree Block
CTU	Coding Tree Unit
CU	Coding Unit
DCT	Discrete Cosine Transform
DPCM	Differential Pulse Code Modulation
DPP	Difference-based Planar Prediction
DST	Discrete Sine Transform
DT	Decoding Time
DTM	Directional Template Matching
DVD	Digital Video Disc
ET	Encoding Time
GAP	Gradient Adjusted Predictor
GASP	Gradient Adaptive Sample-based intra Prediction
GDP	Gradient Dependent Prediction
GSSWAP	Gradient-oriented Selection with Sample-based Weighted Angular Prediction
HD	High-Definition
HEVC	High Efficiency Video Coding
HM	HEVC test Model
I	Intra
IAP	Intensity-variant Angular Prediction
IBP	Improvised Blend of Predictors

IEC	International Electrotechnical Commission
ISAP	Improved Sample-based Angular Prediction
ISDN	Integrated Services Digital Network
ISO	International Organization for Standarization
ITU-T	International Telecommunication Union - Telecommunications Sector
JCT-VC	Joint Collaborative Team on Video Coding
JVT	Joint Video Team
LB	Lowdelay B
LOCO-I	LOw-COmplexity LOssless COder for Images
LP	Lowdelay P
MB	Macroblock
MC	Motion Compensation
MED	Median Edge Detection
MMSE	Minimum Mean Square Error
MPEG	Moving Pictures Experts Group
MPM	Most Probable Mode
MSAP	Modified Sample-based Angular Prediction
MV	Motion Vector
NAL	Network Abstraction Layer
NTSC	National Television System Committee
P	Predictive
PAL	Phase Alternating Line
PB	Prediction Block
PPS	Picture Parameter Set
PSNR	Peak Signal-to-Noise Ratio
PSTN	Public Switched Telephone Network
PU	Prediction Unit
PWM	Pulse Width Modulation
QP	Quantization Paramater
RA	Random Access
RD	Rate-Distortion
RDO	Rate-Distortion Optimization
RDPCM	Residual Differential Pulse Code Modulation
RMD	Rough Mode Decision

RPS	Reference Parameter Set
RQT	Residual Quad-Tree
SAD	Sum of Absolute Differences
SAP	Sample-based Angular Prediction
SATD	Sum of Absolute Transformed Differences
SC	Screen Content
SCM	Screen Content Module
SDC	Sample-based DC
SPS	Sequence Parameter Set
SVS	Scanscope Virtual Slide
SWP	Sample-based Weighted Prediction
TB	Transform Block
TGAPP	Threshold-controlled Gradient Adaptive Planar Prediction
TIFF	Tagged Image File Format
TU	Transform Unit
UHD	Ultra-High Definition
VCEG	Video Coding Experts Group
WPP	Wavefront Parallel Processing
WQVGA	Wide Quarter Video Graphics Array
WSI	Whole Slide Images
WVGA	Wide Video Graphics Array
XGA	eXtended Graphics Array

Chapter 1

INTRODUCTION

In recent years, the progression in the capture, transmission and display technologies have paved way for an increase in the multimedia services, thereby demanding efficient compression strategies to overcome the constraints in bandwidth and storage space. The associated challenges are commonly confronted when dealing with high-resolution video content. As per the CISCO visual networking index forecast, an exorbitant increase in the total number of internet video users is expected from nearly 1.4 billion in the year 2016 to as closely as 1.9 billion by the year 2021, owing to the emerging mediums like the LIVE internet video, virtual reality, augmented reality, etc. This implicitly makes the video content to dominate the overall digital network traffic. As a result, the video traffic is estimated to make a rise from 67% in the year 2016 to around 80% by the year 2021 on an average, of which approximately 22% will be contributed by the high-resolution video contents. The possibly arising concerns have been envisioned by the video coding experts from the ISO/IEC MPEG and ITU-T VCEG organizations. To tackle these issues globally, in the year 2010 they collaborated to deliver immense synergy through the formation of the joint collaborative team on video coding (JCT-VC) and issued a formal joint call for proposals (CfPs) on video compression technology. During the first meeting of the JCT-VC held in Dresden during April 2010, the received responses to the CfP were evaluated and the project name was titled as *high efficiency video coding* (HEVC). The JCT-VC team presided by the chairs met four times a year and after their relentless efforts, the first version of HEVC was made available in January 2013.

HEVC intends to serve as a promising video coding standard, rendering a classic solution to the primary necessities (Sullivan et al. (2012)). The collaboratively standardized HEVC is also referred to as MPEG-H Part 2 (ISO/IEC 23008-2) and H.265 by the

ISO/IEC and ITU-T respectively. It incorporates a sophisticated coding mechanism than the one present in its predecessor, the de facto standard H.264/MPEG-4 Part 10 (Richardson (2010)). HEVC standard is designed not just to enhance the compression efficiency, but also for ease of the transport system integration, data loss resilience, implementability using parallel processing architectures, etc. Without any compromise on the video perceptual quality, HEVC provides a bit-rate reduction of 50% over its predecessor standard H.264. The newly introduced quad-tree structure and other robust coding tools help HEVC to attain superior compression efficiency through a reduction in bit-rate while retaining the video quality (Hang et al. (2010)).

1.1 Motivation

The video content will continue to impose a severe burden on today's global network due to the rapid growth in the usage of the video application and services, along with the increasing popularity of the high definition (HD) video and emergence of ultra-high definition (UHD) (4K and 8K sequences) video formats. Delivering a good quality video to the end-user at a lower bit-rate and higher compression ratio has become a challenging task for the researchers working in the video domain. The history of video compression standardization reveals that it has been dominated by the efforts to maximize the compression capability while minimizing the encoder/decoder computational complexities and improvising the other aspects such as data loss, robustness, etc. Currently, HEVC has established an increased compression efficiency of nearly twice the compression capability of the current video compression standard H.264 (Pourazad et al. (2012)). It intends to support the video sequences with HD resolution and beyond. The standard also aims to mitigate the burden on the global networks, thereby making easy streaming of the HD videos onto the mobile devices.

The lossless mode of HEVC renders distortion-free reconstruction of the image/video data (Sayood (2002)). Currently, there is a steady growing demand for lossless image/video coding in real-world applications such as video surveillance, video analytics, medical imaging, etc. For instance, few applications of video analytics may demand lossless transmission of the video data from the point of acquisition to a remotely located processing unit. In such a scenario, there mustn't be any fidelity loss occurring due to lossy compression. In addition to this, several multimedia applications exist such as video gaming, video conferencing, desktop sharing, remote education etc., that have given rise to a class of sequence

termed as the screen content (SC) video sequences (Xu et al. (2015)). Such sequences are either computer-generated or digitally acquired videos containing a substantial amount of synthetic graphics, text, web pages or animation. Very often, the traditional video coding standards are challenged by the peculiarities of these video sequences such as sharp edges and repeating patterns. On the contrary, gradual variation in the pixel intensities due to the illumination and shadowing effects may demand efficient intra prediction strategies for the removal of the spatial redundancy from the natural video content. A large amount of spatial redundancy also exists due to coherence, smoothness and inherent correlation within the neighboring pixels of the image block. The prediction generation stage can thus play a crucial role in the minimization of the entropy to enhance compression efficiency. Hence, there is a need to find a suitable prediction mechanism that produces residual values with the entropy as low as possible. Improper selection of the predictor can result in inducing erroneous data that could risk permanent manipulation of the original content.

All these factors serve as the motivation to enhance the prediction accuracy through efficient utilization of the spatial correlation within the image block, thereby improvising the compression efficiency of the HEVC standard in lossless mode. In general, the improved intra/inter prediction strategies along with the newly introduced coding mechanisms help HEVC to overcome most of these challenges.

1.2 State-of-the-art Literature Review

The literature comprises of several state-of-the-art lossless image compression strategies in addition to certain video compression techniques. Most of the algorithms aim supremacy in compression with reasonable computational complexity, while few of them relatively work on enhancing the compression efficiency regardless of the complexity. A few of the sample-based techniques that have rendered commendable performance are discussed in this section.

1.2.1 General Image Compression Strategies

The adaptive strategy for compression proposed by Stewart et al. (1994) highlights the various image features that contribute to the compressibility of the image resulting in minimal resource utilization. The proposed adaptive scheme for compression depends on the selection of features that benefit the classification process of the image blocks using the

optimum compression approach. The low-complexity lossless coder for images (LOCO-I) (Weinberger et al. (1996)) proposed by the Hewlett-Packard uses the median edge detection (MED) predictor. It mainly uses the neighboring pixels in the North (N), West (W) and North West (NW) directional orientations to the target pixel $C_{x,y}$. The N and W pixels represent the prediction value in case of detection of vertical and horizontal edges respectively. An edge is detected between (NW, $max\{N, W\}$) and ($C_{x,y}$, $min\{N, W\}$), while NW is larger than N and W. Similarly, when NW is smaller than N and W, $max\{N, W\}$ is the prediction value as $max\{N, W\}$ and $C_{x,y}$ are considered to be the same objects. Under remaining conditions, the planar predictor ($N + W - NW$) is considered as the prediction value. However, the MED predictor limits the prediction value between $min\{N, W\}$ and $max\{N, W\}$.

Seemann and Tisher (1997) proposed a generalized algorithm that performs a locally adaptive compound prediction process incorporating the blending mechanism of a simple set of linear sub-predictors, based on the imposition of the penalty term. The suitable sub-predictors have been tested for greyscale images, color images, and audio data. The prediction scheme is designed to support noisy as well as smooth data, making use of minimal tunable parameters. The context-based, adaptive, lossless image CODEC (CALIC) presented by Wu and Memon (1997) outperforms the remaining image compression techniques available in the literature, in terms of compression efficiency with relatively lesser time and space complexities. Robust performance is achieved by suitably modeling the image data. It makes use of the gradient adjusted predictor (GAP), an adaptive non-linear predictor that adapts to the local intensity gradients around the target pixel. The algorithm considers a neighborhood area of seven causal pixels in the N, W, NW, NN, WW, NE and NNE directional orientations to gather the gradient information. Threshold values are chosen after performing rigorous experimentation to identify and categorize the edges as strong, normal or weak.

The prediction stage in the lossless gray-scale coder for images proposed by Kwon et al. (1999) is based on context-based minimum mean squared error (MMSE) to derive the context model. The directional differences are computed in the four visual directions to identify the edge strength and region smoothness. Accordingly, contexts are defined for adaptive prediction and relevant weights that are to be transmitted from the encoder to the decoder are generated by applying the MMSE criterion. The gradient-based selective weighting prediction strategy in Knezovic and Kovac (2003) exploits the high amount of correlation that exists between the nearby pixels. Predictive coding serves as a decorrelation

tool before statistical modeling and the entropy coding process. The selection of the suitable contributing neighboring pixels and its coefficients is explicitly done based on the gradient information in the specified four viewing directions. Here, the coefficients represent the weights of neighboring pixels that contribute to the prediction process such that the closest neighbor in terms of the gradient measure bears the largest weight.

1.2.2 Related Works in HEVC

The conventional block-based prediction process for the angular modes in HEVC was replaced by the sample-based scheme to enhance the prediction accuracy. The sample-based angular prediction (SAP) technique proposed by Zhou et al. (2012) primarily depends on the two causal reference sample pair i.e. R_i and R_{i+1}, lying in the immediate proximity to the target pixel in a block of size $L \times L$. On selection of the reference pair R_i and R_{i+1}, they are linearly interpolated using the weighting parameter $fract$ to generate the prediction as provided in (1.1). In general, to predict a given row/column of pixels, maximum of L + 2 samples are needed for any of the 33 angular modes. The pixels in the first row/column of the target block are predicted using the row/column in the previously reconstructed block on the top/left to the block under consideration. The reconstructed samples corresponding to the boundary pixels are padded to facilitate the R_{i+1} sample for predicting the boundary pixel in the succeeding rows/columns, thereby resulting in the zero-angle prediction.

$$\hat{P}[x, y] = ((32 - fract) * R_i + fract * R_{i+1} + 16) >> 5 \tag{1.1}$$

To overcome the shortcoming of zero-angle prediction in SAP, a modified sample-based angular prediction (MSAP) for positive vertical/horizontal modes is proposed in Antony and Sreelekha (2015). The modification lies only in the formation of the reference pair R_i and R_{i+1} for the boundary pixels. Antony and Sreelekha (2017) also proposed the concept of context-based prediction which is employed in the improved sample-based angular prediction (ISAP). It mainly relies on the reference sample R_i being identified as a peak or a valley sample, based on a certain threshold value. When the desired conditions are satisfied, the pixel intensity value corresponding to the reference sample R_{i+1} is determined as stated in Antony and Sreelekha (2017). Once the reference pair R_i and R_{i+1} is chosen, they are linearly interpolated as given in (1.1). In all the remaining cases, the pixels are predicted using the SAP (Zhou et al. (2012)) technique. Further, a gradient adaptive sample-based intra prediction (GASP) technique was proposed by Antony and Sreelekha (2017) that replaces

the conventional block-based planar mode in the HEVC framework. The algorithm uses the GAP predictor (Wu and Memon (1997)) to identify the abrupt transitions and intends to exploit the sample-based benefits to improve the prediction accuracy.

Wige et al. (2013) presented sample-based weighted prediction (SWP), a low-complexity pixel-by-pixel mechanism that merges the linear predictors with the exponentially decreasing weights derived using the non-local means algorithm. A simple mathematical formulation and look-up table approach are used to derive the optimal integer weights, thereby minimizing the complexity in HEVC. While the algorithm has proven to render better performance in the case of natural video sequences, a noticeable reduction in the prediction efficiency is observed for the SC sequences owing to the averaging operation performed around the sharp edges. To overcome this drawback, the SWP algorithm has been subtly modified. In the modified algorithm referred to as the directional template matching (DTM) technique (Wige et al. (2013)), the concept of weighted averaging is replaced with the identification of the pixel that associates a close correlation with the target pixel which will be used as the prediction value.

In residual DPCM (RDPCM) presented by Lee et al. (2013), the DPCM is imposed on the generated residuals at the residue coding stage, after the application of the regular block-wise intra prediction. Furthermore, pixel-by-pixel differential pulse code modulation (DPCM) in Jeon et al. (2014) intends to enhance the prediction accuracy by extracting the gradient information from the surrounding pixels. Additional DPCM is applied based on the residuals obtained, post applying the pixel-by-pixel DPCM. This method is applicable only for horizontal and vertical intra prediction modes. An extension of this work is proposed in Kim et al. (2014), wherein an additional prediction strategy is imposed on the residuals obtained after applying RDPCM.

HEVC based lossless compression of the whole slide pathology images is proposed in Sanchez et al. (2014). These images are highly textured containing several edges and multidirectional patterns, due to the existence of cellular structures and tissues. This serves as a robust motive for enhancement of the prediction performance, where the edge information is dominant. It is accomplished by embedding the sample-by-sample differential pulse code modulation and edge prediction strategy into the intra coding process. Sanchez et al. (2016) then proposed a fast intra approach that makes use of four modes for the prediction of these pathology images. Amongst the four, one of the prediction modes is a novel median edge predictor to predict the edges present in various orientations. The validation mainly performed on these images claims to have accomplished improvements in lossless coding and

attained considerable savings in run-time as well. On the other hand, the work proposed in Chen et al. (2014) aims to retain the edge information which is the vital characteristic feature of the SC video sequences. This is achieved using the selective replacement of the bilinear interpolation by the nearest-neighbor interpolation based on the prediction samples. Sanchez (2015) also proposed a sample-based angular prediction with median and edge prediction for lossless coding of the SC video sequences to achieve significant coding gains. It makes use of angular DPCM, median and edge predictors targeting the directional structures, homogeneous surfaces and regions with significant edges. Further, the technique of piece-wise mapping in HEVC for lossless mode was proposed by Sanchez et al. (2016) to reduce the energy of the intra predicted residual blocks which is imperative for the reconstruction process. This is achieved using the piece-wise mapping (PWM) function, wherein the range of values in the residual blocks are analyzed to apply a suitable PWM function to map the specific residual values to lower levels. The associated PWM parameters are encoded and signaled to the decoder, to perfectly reconstruct the mapped residual blocks.

From the above literature review, it is quite obvious that a lot of research has been carried out to enhance prediction accuracy. It also reveals that the sample-based prediction mechanisms help to attain the same. However, there is a lot of scope for improvising the image compression concepts and apply them to the video sequences. The improvisations can be extended to the block-based intra prediction strategy of HEVC to get benefited from the sample-based approaches and achieve superior coding gains.

1.3 Research Objectives

Based on the literature review, the following research objectives are proposed to develop a robust and efficient video compression algorithm.

1. To design a gradient-based algorithm to modify the HEVC planar and angular prediction modes, by incorporating threshold-controlled gradient adaptive strategy for planar prediction and gradient-oriented selection with a sample-based weighted technique for angular prediction

2. To design a context-based algorithm to suitably modify the conventional intra prediction process of HEVC in accordance with the local image characteristics, by incorporating difference-based planar prediction, combined sample-based DC prediction and intensity-variant angular prediction strategies

3. To design an algorithm to blend the suitable sub-predictors based on the derived penalizing factors for capturing the dominant image characteristics, by modifying the conventional planar and angular intra prediction process of HEVC

The evaluation of the algorithm is quite essential to check its performance. It is achieved by -

- Performing the analytical and simulation studies on the compression techniques to quantify the improvements in compression efficiency
- Evaluation of the algorithmic computational complexities through the comparison of its encoding and decoding time of the compression technique
- Derivation of the intra prediction modes distribution to account for the samples that favor the modification
- Conduction of the residual analysis at the decoder to understand the reliability of the algorithm

1.4 Contributions of the Thesis

This section gives a brief insight into the thesis contributions.

- **Gradient-based predictor for lossless coding of HEVC planar and angular modes:** The proposed gradient dependent prediction mechanism is a composite of the threshold-controlled gradient adaptive planar predictor and gradient-oriented selection with a sample-based weighted angular predictor. Mode dependent templates of the causal neighbors are used to compute the gradient, which is the preliminary step in the process of extracting the edge information. The derived gradient value forms the basis for the generation of the prediction based on certain criteria. The major highlight of the proposed method is the prominent improvements in coding gains for the natural sequences without increasing the overall complexity.
- **Context-based predictor for lossless coding of HEVC planar, DC and angular modes:** The proposed context-based predictor adapts suitably to the characteristic features of the data block based on the contextual background of the neighborhood samples around the target pixel. The variability in the sum of absolute differences

and local intensity values are used to derive the context of the target pixel in case of planar and angular prediction modes respectively, in association with the gradient data. Meanwhile, the proposed sample-based DC prediction technique performs the averaging of proximally located causal samples and the DC value which is generated using the neighborhood blocks. The simulation results reveal that the proposed predictor is a promising candidate for lossless compression of the natural and SC video sequences.

- **Heuristic blend of sub-predictors for efficient prediction of HEVC planar and angular modes:** The proposed improvised blend of predictor is based on the past predictions of the neighborhood around the target pixel. The mechanism of blend helps the predictor to adapt well to the targeted regions with smooth variations and edges by imposing the penalizing factor. The simulation results reveal that the proposed prediction strategy achieves superior coding gains over the HEVC anchor for the natural and SC video sequences. The results also claim that the proposed predictors outperform the state-of-the-art prediction schemes as well.

1.5 Thesis Outline

This section briefly narrates the chapter-wise organization of the research thesis for a better understanding.

Chapter 2 presents a general introduction to the necessity and basics of video compression. It focuses on the evolution of video compression standards and provides minimal insight into the associated technical aspects of these coding standards developed over the decade. A detailed description of the fundamental building blocks of HEVC and its coding tools has been provided in this chapter. It also expounds the software and hardware specifications that exhibit a vital role in the validation of the proposed algorithms, under standard common test conditions using the sequences in the test dataset.

Chapter 3 presents objective 1 that intends to enhance the compression efficiency of the HEVC standard through the sample-based modification in the prediction mechanism of the planar and angular modes. This has been accomplished using the threshold-controlled gradient adaptive strategy for planar prediction and gradient-oriented selection with a sample-based weighted strategy for angular prediction. Simulation results of the standalone predictors and their combination are presented and analyzed.

Chapter 4 presents objective 2 as a context-based algorithm that is based on local image characteristics. It is achieved using the difference-based planar prediction, combined sample-based DC prediction and intensity-variant angular prediction strategies implemented at the pixel level and implanted into the planar, DC and a specific angular mode of the HEVC framework. The section on the simulation results discusses the behavior of the standalone predictors and their suitable combinations, demonstrating the superiority of the proposed work in comparison with the HEVC anchor.

Chapter 5 presents objective 3 as a pixelwise prediction algorithm that blends the suitable sub-predictors based on the penalizing factors to capture the dominant image characteristics. The algorithm being incorporated into the planar and a specific angular mode has been designed to precisely derive an optimum predictor to handle the regions featuring smooth variations and several edges. For a fair comparison, the chapter includes the performance results of the various state-of-the-art prediction techniques in addition to the algorithms proposed in this thesis.

Chapter 6 summarizes the contribution of the proposed algorithms in the preceding chapters and derives the conclusion from the work that constitutes the thesis. It also suggests the future scope of work in the focused research area.

Chapter 2

FUNDAMENTALS OF HEVC

2.1 Need for Video Coding Standards

Any digital video comprises a sequence of images or frames. Each video image resembles the projection of several objects possessing depth, texture, and illumination in a 3D scene onto a 2D plane (Richardson (2002)). An illusion of motion is created to the viewers by sampling the video signal temporally i.e. displaying these frames swiftly in succession at a specified frame rate. A higher frame rate entitles the viewer to visualize a smoother appearance of the motion present in the video scene, but at the cost of more samples to be captured and stored. In a simple illustration, consider a 90 minute 1080p raw full HD video content of resolution 1920×1080 ($frame\ width \times frame\ height$) pixels and aspect ratio 16:9. To store this video content, a storage space with capacity of 90 ($minutes$) * 60 ($seconds/minute$) * 24 ($frame\ rate$ in $frames/second$ (fps)) * (1920×1080) ($resolution$ in $pixels$) * 3 ($number\ of\ bytes/pixel$) = 806.2 GB is required. The calculated value far exceeds the capacity of the currently available optical disc storage facilities i.e. 50 GB dual-layer Blu-ray disc. This illustration makes it evident that digital video consumes a large amount of storage or requires more bandwidth to increase the transmission capacity. Therefore, there is a need to compress the videos before they are stored or transmitted and decompress the same prior to their display, for any application in which storage capacity or transmission bandwidth is a constraint. Figure 2.1 emphasizes the various current day scenarios where video compression is extremely imperative to curb the overall video traffic.

Achieving optimized usage of the bandwidth and storage space for digital videos through video compression has led to the evolution of several video coding standards. Without

compression, it would be challenging to transmit the data through the bandwidth-limited wireless link. The video coding standards define a toolkit for video compression to create a standard-compliant bitstream. Compression efficiency is defined as the ability to minimize the bit-rate (bits per unit time) of the video content to attain a suitable level of video quality for its representation. In general, a higher bit-rate implies a classy video quality, but at the cost of more data storage space and a need for broader bandwidth to transmit and playback the video content. Bit-rate minimization is achieved by exploiting the temporal redundancy between the frames and spatial redundancy within the frame, thereby eliminating the insignificant information.

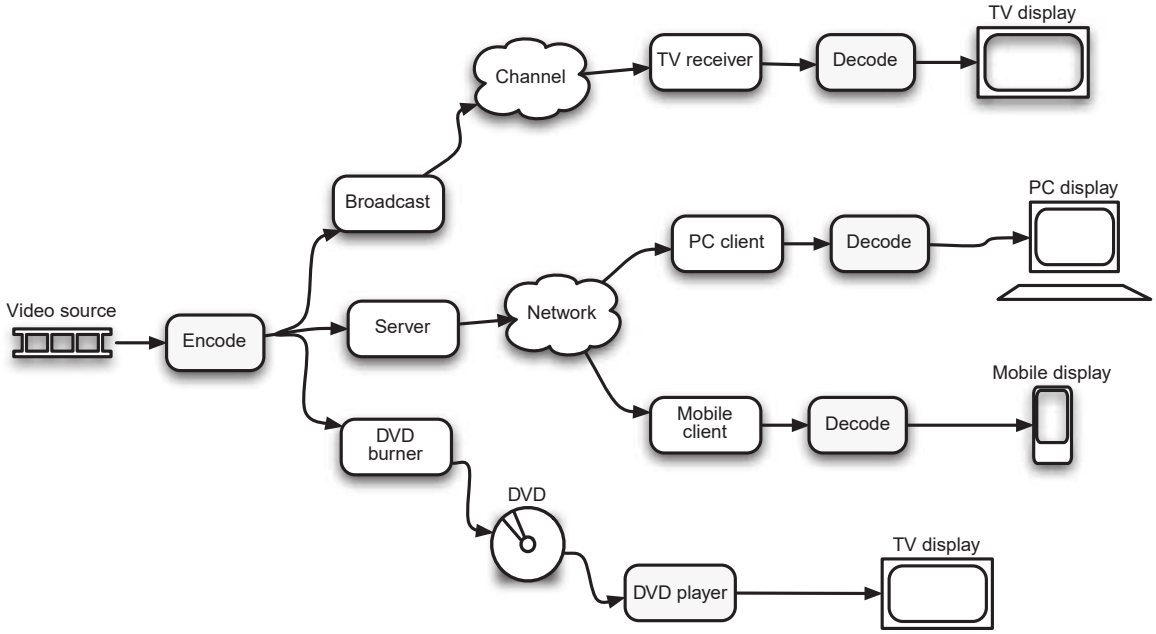


Figure 2.1: Current day scenarios that mandate video compression

2.2 Basics of Video Compression

Any digital video is a representation of the spatiotemporally sampled video scene in the digital form. The process of elimination of the redundant information persistent in the original video sequence is termed as $encoding$ and the amount of time taken to accomplish this task is called the $encoding\ latency$. The process of performing the inverse actions to playback the compressed video through its reconstruction as closely as possible to the original content is referred to as the $decoding$. The conjoint interoperability of the encoding and decoding processes form the basis of a CODEC (enCOder/DECoder). A CODEC can

hence be defined as a tool that uses an algorithm to systematically compress the raw video data into a compact form, suitable for broadcasting and transmitting over an internet stream or for storage on a disc. Meanwhile, at the receiving end, the CODEC in the television, computer or disc player, uncompresses the data prior to the display of the video content on the screen.

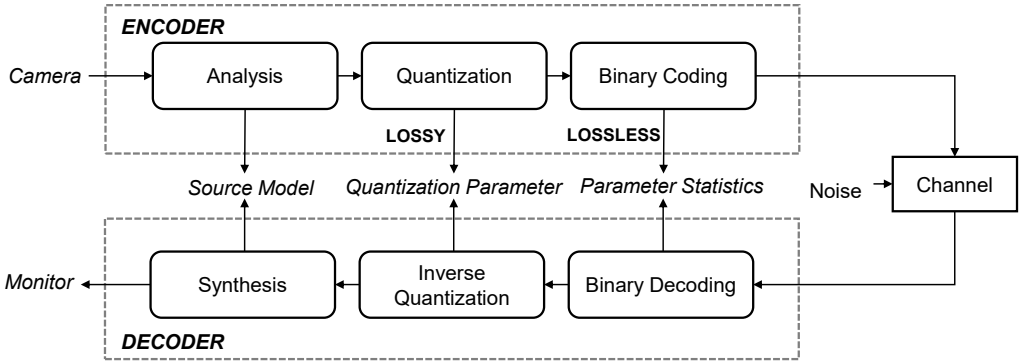


Figure 2.2: General block diagram of encoder and decoder
Courtesy: Wang et al. (2002)

Figure 2.2 describes the basic components of a video coding system (Wang et al. (2002)). The efficacy of such a system is determined to a large extent by the source model that is adopted for modeling the video content. The source model may make certain assumptions on the spatial and temporal correlation between the pixels of a sequence. It may also consider the shape and motion of the object or illumination effects. At the encoder, the digitized video sequence is first described using parameters of the source model. In the succeeding step, parameters of the source model are quantized into a finite set of symbols, thereby introducing a certain amount of loss. The lossy quantized parameters are finally mapped into binary codewords using a lossless coding technique. The resultant bitstream is then transmitted over the communication channel. At the decoder, the quantized parameters of the source model are retrieved by reversing the binary encoding and quantization processes performed at the encoder. The synthesis algorithm finally reconstructs the decoded video frame that is to be projected on a display device.

The process of encoding deals with three types of video frames, namely I-frames, P-frames, and B-frames as shown in Fig. 2.3. Here, the display order of the frames will be in the manner as shown in the figure varying numerically from $1 \rightarrow 8$. However, the decoding order is based on the type of frame and thus will be: $1 \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow 6 \rightarrow 5 \rightarrow 8 \rightarrow 7$. The I-frame, which stands for intra frame, is a self-contained frame that is independently

encoded without any reference to the preceding or succeeding frames. The encoding of the P-frame or predictive inter-frame depends on the previous I- or P-frames. The bi-predictive inter-frame commonly referred to as the B-frame, is a frame that relies on the preceding and succeeding I or P frames. Now, to avoid transmission of the damaged bitstream, an I-frame is placed first and at regular intervals within a video sequence. This is essential to implement fast-forward, rewind and scene change detection. The I-frames do not contribute much to the compression savings as they are intra coded, thereby consuming more number of bits on encoding. On the other hand, although the P-frames typically require fewer bits than the I-frames, they are more susceptible to transmission errors due to their dependency on the previous reference frames during the process of encoding.

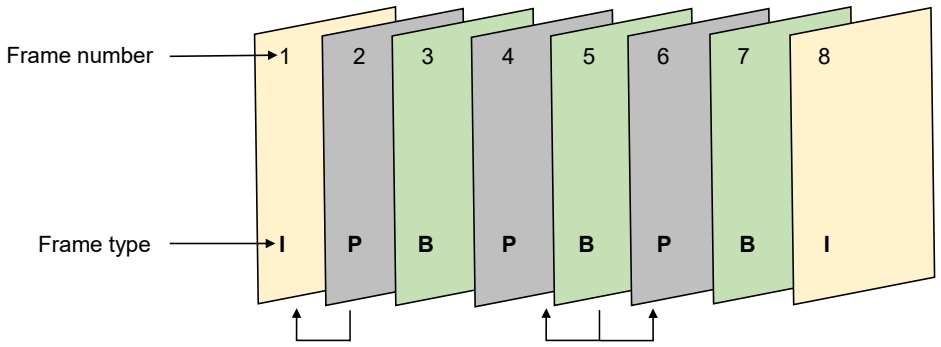


Figure 2.3: Typical I, P and B frame positioning within a video sequence

In general, compression can be either lossy or lossless. Lossy compression means permanent removal of the irrelevant spatial and temporal redundant information from the video sequence. Temporal redundancy refers to the fact that consecutive frames are typically fairly similar to each other. Very often, these consecutive frames are almost identical, except that there might be a subtle amount of shift due to the motion. This kind of redundancy can be exploited at the block level by storing the difference or error between the current frame and previous/succeeding frame that represents the amount of shift as illustrated in Fig. 2.4. Since the differences are usually small, they can be encoded using fewer bits than the number of bits needed to represent the original pixel values. This process is known as motion compensation (MC) and the amount of shift is indicated by the motion vector (MV) as shown in Fig. 2.5. The MV is the key element in the motion estimation process which takes fewer bits on encoding, thereby saving the bandwidth during the transmission of the encoded bitstream.

Figure 2.4: An illustration for temporal redundancy

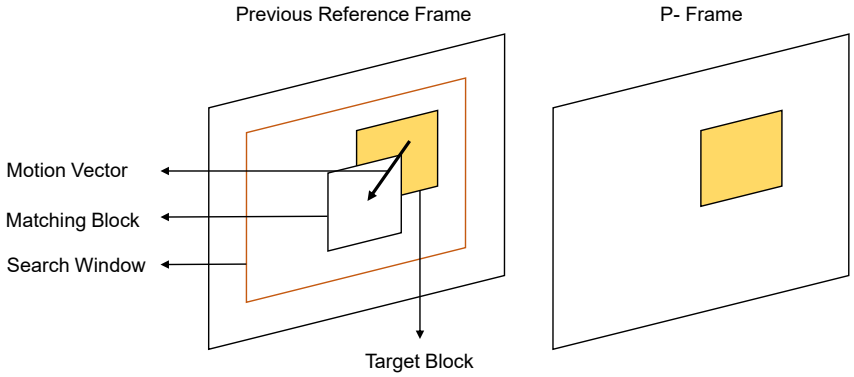


Figure 2.5: Block-based motion compensation

On the other hand, spatial redundancy implies that the pixels located close to each other spatially typically have similar values as shown in Fig. 2.6.

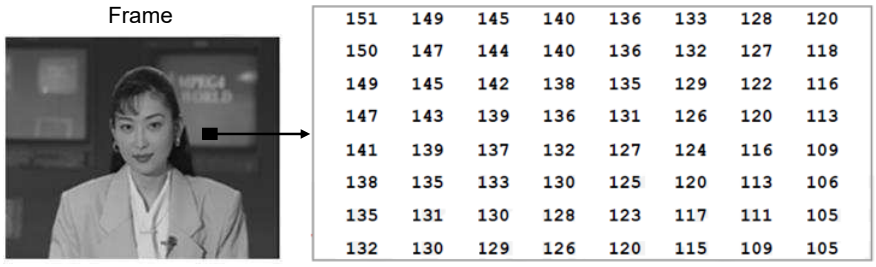


Figure 2.6: An illustration for spatial redundancy

Meanwhile, the lossless compression technique represents the image/video content with fewer bits in a manner that the original content is reconstructed without any error or distortion, retaining the archival quality. This is achieved by mainly bypassing the quantization and transformation blocks at the encoder and its inverses at the decoder. The lossless feature is extremely beneficial whenever certain applications like video analytics, video surveillance, medical data compression, etc. mandate distortion-free reconstruction of the original

content. Additionally, there are certain other multimedia applications such as cloud video gaming, video-conferencing, desktop sharing, etc. that require to maintain the original video integrity through lossless compression. Due to the enormous growth in the demand for these applications currently, there has been a lot of interest developed towards the lossless compression techniques.

2.3 Evolution of the Video Coding Standards

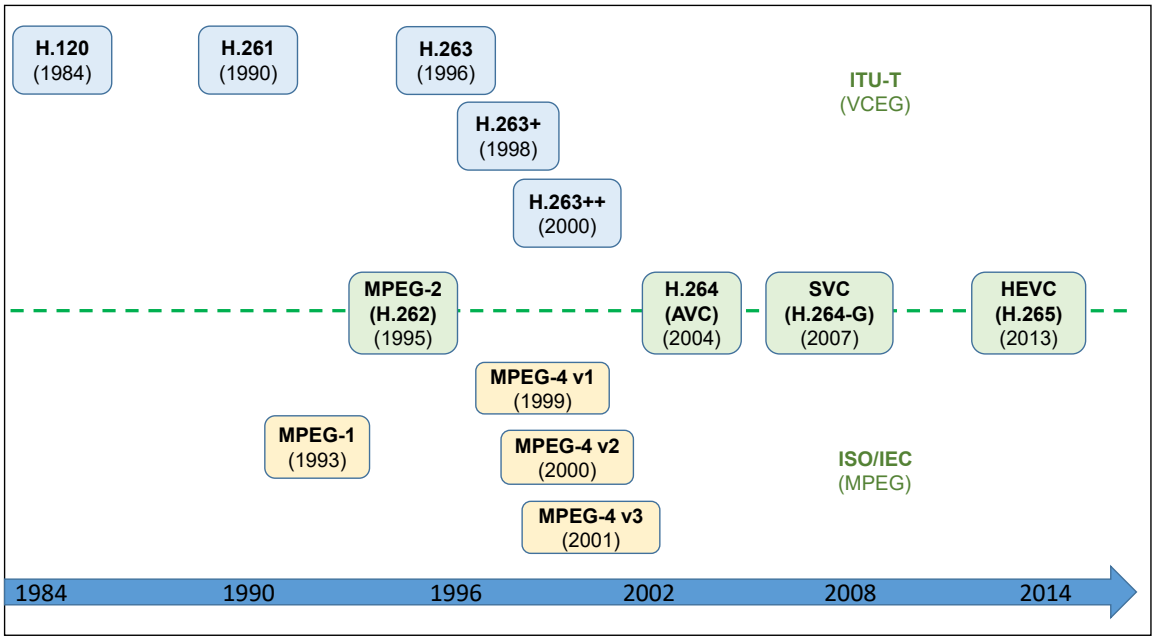


Figure 2.7: Evolution of the digital video coding standards

The global network is immensely burdened due to the plethora of multimedia services made available to the end-user. Better compression tools are now a global necessity due to the advancement in multimedia services and proliferation of high-resolution digital videos to avoid the blocking, ringing and flickering effects in the video. The ISO/IEC and ITU-T, two giant organizations pioneered the process of video coding standardization. The moving pictures experts group (MPEG) is a working group of experts from the ISO and IEC, intending to set standards for audio and video compression along with its transmission. Meanwhile, another working group of experts from ITU-T formed the video coding experts group (VCEG) with a goal to provide video and image coding standards for conversational and non-conversational services. This group has been responsible for the standardization

of the H.26x video coding standards. When the video coding technology was in its infancy around the early 1980s, several notable international standards developed individually or jointly by these groups have served as the powerful compression tools behind the commercial success of digital videos. Figure 2.7 depicts the evolution of the digital video coding standards developed over four decades. These coding standards vary from each other in terms of compression ratio, computational complexity, etc.

The first digital video compression standard H.120 was developed in the year 1984 by the consultative committee for international telephony and telegraphy (CCITT, now ITU-T). This standard was not quite successful as the visual quality was inadequate, hence there is no CODEC available for this format. Then H.261 was developed by the CCITT for video conferencing and videotelephony applications. It emerged as the first practical video coding standard to use the processing unit termed as the macroblocks in its video CODEC. The MPEG-1 standard developed by the ISO/IEC aimed to compress the digital video and audio down to 1.5 kbps. It emerged as a compatible lossy audio and video format as it is been used in a large number of products and technologies even in the present day. In the subsequent years, the first outcome of the joint effort by the ISO/IEC and ITU-T was developed and named as MPEG-2/H.262. It has been termed as the generic coding of the motion pictures with the associated audio information for storage and broadcast applications at typical bit-rates of 3-5 Mbps and above. Next, the ITU-T developed H.263 standard to facilitate videotelephony via circuit- and packet-switched networks, supporting a range of channels from low bit-rates (20-30 kbps) to higher bit-rates (several Mbps). Further, H.263+, H.263++ standards came into existence as extensions of H.263 to support a wider range of transmission scenarios. The underlying framework of these extensions remained the same, apart from the fact that the capabilities of H.263 have been further enhanced to improve the compression efficiency and provide additional facilities such as superior robustness against loss of data in the transmission medium. The ISO/IEC then came up with MPEG-4 standard and its versions, aiming certain major applications like media streaming, voice, and television broadcasting, by supporting a wide range of bit-rates from 20 kbps to higher bit-rates.

Table 2.1 briefly summarizes the applications and compression quality of the major video coding standards (Akramullah (2014)). The standards have played a pivotal role in setting up the technology and ensuring the interoperability among the products that were developed by various manufacturers.

Table 2.1: Summary on the prime video coding standards developed over the decades

Coding Standards	Applications	Compressed Data Rates
H.120	Video conferencing using digital group transmission	Streams at 1544 kbps for NTSC and 2048 kbps for PAL
H.261	Video conferencing and videotelephony over the ISDN	64 kbps and its integral multiples
MPEG-1	Storage of digital video and audio on CD	1.2–1.5 Mbps
MPEG-2/H.262	Favors video on demand, DVD storage and digital television broadcasting	3–5 Mbps
H.263	Videotelephony via circuit- and packet-switched networks and multimedia on low bit-rate networks like PSTN, ISDN, and wireless networks	$>$20 kbps
H.263+	Video conferencing, mobile multimedia content for playback, internet video streaming, surveillance and monitoring	Supports lower bit-rate than H.263
MPEG-4	Media streaming, voice and television broadcasting	$>$20 kbps and $<$1.5 Mbps
MPEG-4 Part 10/H.264	Videotelephony, DVD storage, internet streaming applications, low and high-resolution digital video	Enhanced PSNR compared to MPEG-4 for the same bit-rate

In the year 2004, the joint effort by the ISO/IEC and ITU-T under the title joint video team (JVT) resulted in MPEG-4 Part 10/H.264. It is the present-day popular video compression standard that provides a gain close to 50% in terms of compression efficiency and supports a wide range of video resolutions up to 4K. Currently, the infrastructure is incapable to transmit the HD and UHD video content to the customer owing to the present-day heterogeneous wireless links, scarce wireless spectrum, and data-intensive video applications, that will degrade the video quality. To combat these issues, HEVC promises to render a solution to overcome the drawbacks through its sophisticated coding mechanisms than the one that exists in its predecessor video coding standard H.264.

2.4 Advanced Video Coding - The Current Standard

The advanced video coding (AVC), known as H.264/MPEG-4 Part 10, is the most popular video compression standard used for video recording and distribution. The standard was ratified in the year 2003 by the JVT formed with experts from the ITU-T VCEG and ISO/IEC MPEG standardization organizations. The H.264 technology design supports the coding of video that addresses both conversational (eg. videotelephony) and non-conversational (eg. storage, broadcast, or streaming) applications. It has a block-based coder/decoder where each frame is divided into small square macroblocks (MBs). The coding tools are applied to MBs rather than to whole frames, thereby minimizing the computational complexity and improvising the accuracy of motion prediction (Juurlink (2012)). Figure 2.8 elaborates on the process of encoding and decoding using the H.264 standard.

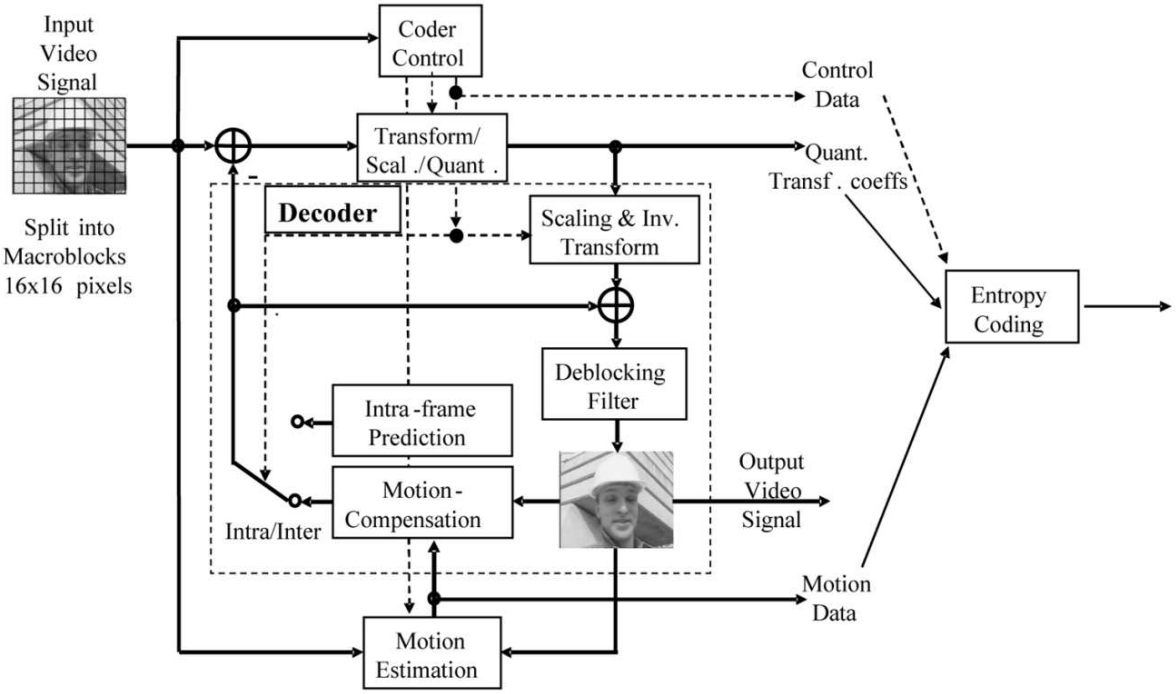


Figure 2.8: H.264 CODEC
Courtesy: Wiegand et al. (2003)

H.264 makes use of a lossy predictive, DPCM coding technique for the prediction process. It offers a maximum of nine intra coding modes that include one DC and eight directional modes, based on the block sizes i.e. 4×4, 8×8, or 16×16. The mode that gives the least rate-distortion (RD) cost value is considered as the best-suited prediction mode.

Blocks of large sizes result in lesser prediction accuracy but reduces the signaling cost of the prediction mode as fewer bits are needed to be transmitted. It carefully handles the information about the slice, picture, and sequence by providing suitable error protection in the lossy environment, which would otherwise render the data useless as in the case of the previous standards.

2.5 High Efficiency Video Coding - An Overview

An overview of the HEVC encoder and decoder is elaborated in Fig. 2.9. The video pictures are fed to an encoder that encodes these pictures into a bitstream. The order of the bitstream is the same as the decoding order i.e. the order in which the pictures are decoded at the decoder. HEVC bitstream is made up of encoded syntax elements termed as the network abstraction layer (NAL) units, each of which contains an integer number of bytes. The initial two bytes of the NAL unit represents the NAL unit header, while the remaining bytes of the NAL unit renders the payload information. Few of the NAL units carry the parameter sets loaded with the control information pertaining to a single or several pictures, while the other NAL units carry the coded samples belonging to an individual picture. The decoder decodes these NAL units to generate the decoded pictures. Both the encoder and decoder store these pictures in a decoded picture buffer. The pictures stored in this buffer are termed as the reference pictures that facilitate the generation of the prediction signals using the previously coded pictures on coding the current picture.

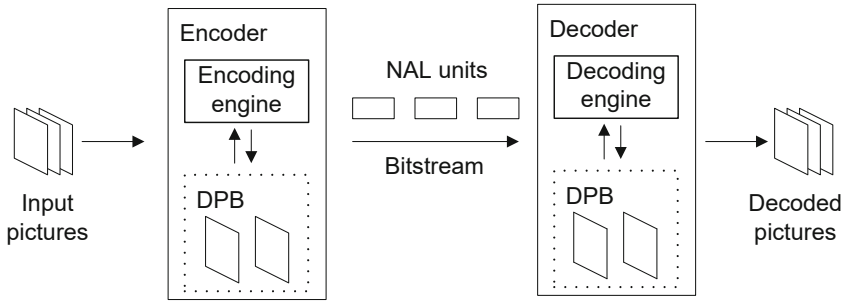


Figure 2.9: Overview of HEVC encoding and decoding
Courtesy: Sze et al. (2014)

The block diagram of a block-based HEVC video encoder represented in Fig. 2.10 highlights the typical encoding and decoding processes involved (Sze et al. (2014),Wein (2015)). Each input picture is made up of three components Y, U, and V. The Y component, also

referred to as the luma represents the brightness, while U and V components represent the color information. The human visual system is sensitive to the luma component than chroma. Hence, sampling is performed to retain just one-fourth of the total number of samples in each of the chroma component corresponding to a luma component, which is commonly termed as the 4:2:0 sampling format. Every sample in the individual color component is represented with 8-bit precision.

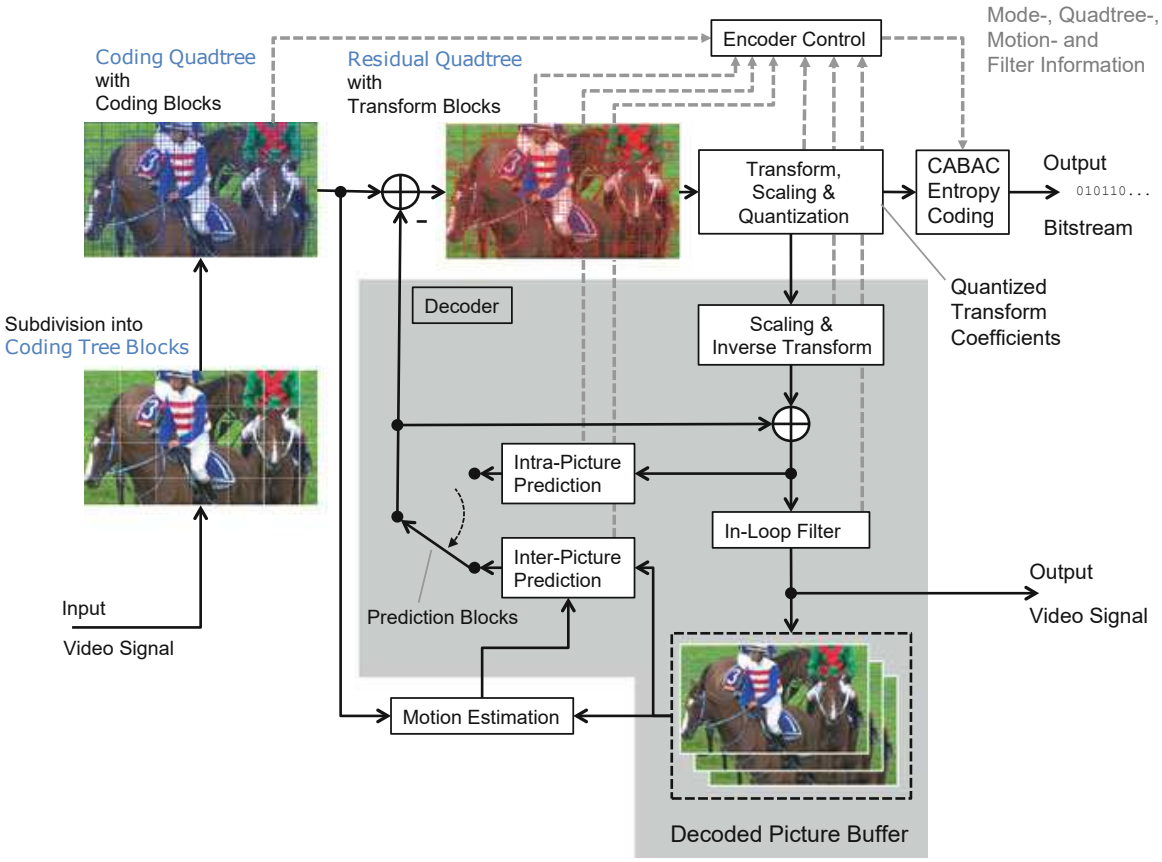


Figure 2.10: Block diagram of HEVC encoder with a built-in decoder
Courtesy: Sze et al. (2014)

The coding tree unit (CTU) is the basic processing unit in HEVC, which constitutes the coding tree block (CTB). The input picture is thus divided into CTBs and the resultant syntax structure is finally placed into the logical data packet termed as a NAL unit. The quad-tree mechanism is used to further partition the CTUs into coding units (CU) that encapsulate the coding blocks (CB) corresponding to the luma and chroma components. These CBs are coded using either intra or inter prediction modes. In the inter mode, the chosen CB is predicted using motion-compensated prediction. While in the intra mode,

each sample of the CB is predicted using the causal neighboring samples of the previously coded blocks within the same picture. The prediction residue, which represents the difference between the original and predicted block is transformed, quantized and entropy coded. Meanwhile, the encoder duplicates the decoder processing loop such that both will generate identical predictions for the subsequent data. Hence, the quantized transform coefficients are inverse scaled and inverse transformed in the same manner as in a decoder, resulting in the decoded prediction residue. This is then added to the prediction value, with the result being processed by an in-loop filter comprising of sample-adaptive offset and deblocking filters. The subsequent output generated represents the reconstructed CB which is stored in the memory and made available at the decoder.

2.6 HEVC Structure and Coding Tools

This section briefs on the structure and various coding tools of HEVC (Sullivan et al. (2012)). As per the HEVC specification document issued by the JCT-VC, the term $block$ refers to an area-specific to a color component (e.g. luma, chroma). Meanwhile, the term $unit$ includes the $blocks$ of all encoded color components (Y, U, and V) associated with an area, along with the syntax elements and block-specific prediction information like the type of prediction, motion vectors, etc.

2.6.1 Coding Tree Unit

The square-shaped coding tree unit (CTU) forms the core of the coding layer in HEVC. Each CTU consists of a luma Coding Tree Block (CTB), corresponding chroma CTBs and the associated syntax elements as shown in Fig. 2.11. The luma CTB is made up of $2^L \times 2^L$ luma samples and each of the two chroma CTBs constitutes $2^{L-1} \times 2^{L-1}$ chroma samples. The parameter L needs to be loaded in the sequence parameter set (SPS) so as to facilitate the encoder to choose among $L = \{4, 5, 6\}$, corresponding to CTU sizes 16×16, 32×32 and 64×64 respectively.

The selection of larger CTU sizes typically provides better compression efficiency but may increase the memory requirements and computational complexity of the encoder process, thereby inducing encoder/decoder delay. Based on the type of application, the CTU size is chosen by the encoder to provide the best trade-off. Figure 2.12 illustrates the partitioning of a video picture with resolution 1920×1080 into CTUs of size 64×64.

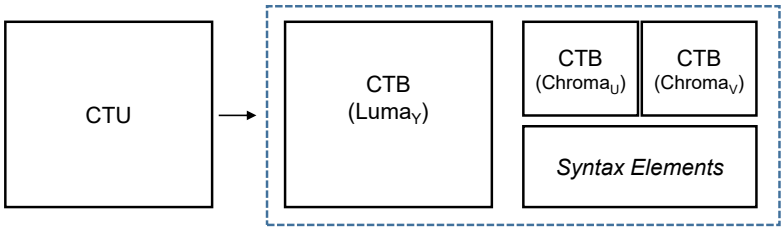


Figure 2.11: A CTU

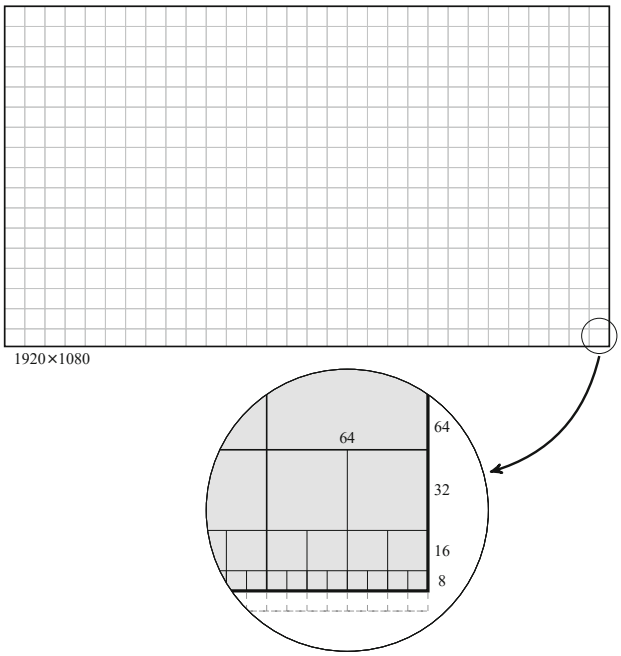


Figure 2.12: An illustration for partitioning of a video picture into CTU of size 64×64
Courtesy: Wein (2015)

2.6.2 Coding Unit

The quad-tree structure also referred to as the coding tree of the CTU specifies its sub-division into the coding unit (CU). A CU consists of a square block of luma samples, two corresponding blocks of chroma samples, associated syntax and information on the type of the prediction. Here, each block is termed as a coding block (CB). The splitting of a CTU into luma and chroma CBs is performed via joint signaling. The CB sizes range between 8×8 and 64×64. A CTU may contain only one CU or may be split recursively into multiple CUs. It is imperative that the horizontal and vertical size of a video picture needs to be an integer multiple of the minimum size of the CU i.e. 8×8. On the other hand, if

the horizontal or vertical size of the video image is not an integer multiple of the CTU size as illustrated in Fig. 2.12, then the CTBs at the borders are implicitly partitioned until the boundaries of the resulting CB blocks coincide with the picture boundary.

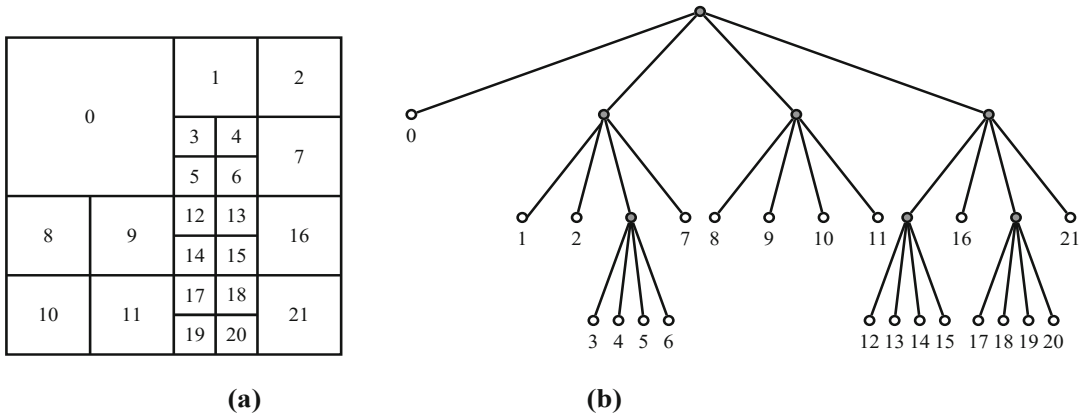


Figure 2.13: Partitioning of CTU into CUs (a) Spatial partitioning (b) Quad-tree representation
Courtesy: Wein (2015)

Figure 2.13 illustrates a typical scenario of quad-tree partitioning of a CTU into CUs. The numbering scheme employed in the spatial partitioning of the CTU is in the order of coding a CU based on the depth-first order. This coding order ensures that all the samples belonging to the CUs on the left and above have already been coded prior to the coding of the samples in the current CU. However, there is an exception in the case when the samples belong to the CU that is located along the frame boundary on the top or left. This scheme is deployed to facilitate the usage of the previously coded samples for the prediction process and its associated coding parameters to predict the coding parameters of the current CU. When the CTU is split into CUs, a $split\_cu\_flag$ is transmitted corresponding to every CU specifying whether the block represents a CU or it is further split into four equally-sized blocks. The hierarchical subdivision process of CTU into CUs halts when the minimum CU size is reached and no flags are transmitted indicating the termination of the split process corresponding to the individual blocks. Additionally, every CU is assigned a coding mode which indicates whether the associated samples are predicted using either intra or inter prediction mechanisms. On the whole, the quad-tree structure provides a unified syntax benefiting the intra/inter predictions in addition to the process of transform coding.

2.6.3 Prediction Unit

To perform the prediction using the inter/intra modes, the CB is further non-recursively partitioned into prediction blocks (PBs). The luma and chroma PBs together with the associated syntax elements constitute the prediction unit (PU). Figure 2.14 depicts the CB partitioning scheme for the intra and inter prediction modes of HEVC. In intra prediction, the size of a PB is identical to the size of a CB (Fig. 2.14(e)), except in case of CB size 8×8 where the PB can be either 8×8 or 4×4 (Fig. 2.14(j)) due to the subsequent partitioning. However, for motion-compensated inter prediction of the CBs, eight partitioning schemes are available. Here, $n = L/4$ and its subscripts T, B, L, and R indicate the position of the sub-block i.e. top, bottom, left and right respectively. The motion-compensated inter prediction supports more number of partitioning schemes that relevantly split into several sub-blocks, thus contributing to compression efficiency. If a CB is split into four PBs, then the resultant PBs represent square blocks each of the same size (Fig. 2.14 (d)). On the other hand, it is observed that when the CB is split into two PBs, two symmetric (Fig. 2.14 (b) and (c)) and four asymmetric (Fig. 2.14 (f), (g), (h) and (i)) partitioning schemes are supported.

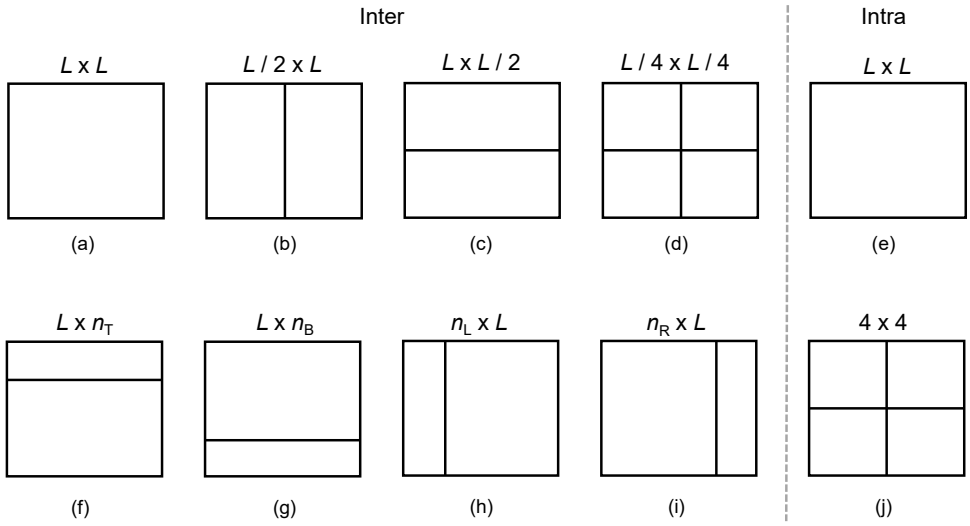


Figure 2.14: Partitioning scheme for inter and intra prediction for a block of size $L \times L$

2.6.4 Transform Unit

A transform unit (TU) encapsulates the luma and chroma transform blocks (TBs) along with the associated syntax elements. The recursive partitioning of the CBs into TBs using the residual quad-tree (RQT) approach is adopted to transform code the generated prediction

residuals. Here, the CB will serve as the root of any transform coded tree. The TB size varies between 4×4 and 32×32. As the maximum supported transform size is 32×32, a CB of size 64×64 is split into four 32×32 TBs. The maximum and minimum supported TB sizes along with the maximum applicable transform quad-tree depth (t_D) are the three essential parameters to configure the SPS to restrict the RQT.

Whenever the assigned value of the parameter t_D is one, the luma CB can either be transform coded as a single TB or split into 4 TBs, but no further splitting is permitted. On the contrary, if the luma CB size is 64×64 and t_D is set to zero, then the CB is forced to be sub-divided into TBs of size 32×32 to comply with the limitations on the transform size. It is also imperative to understand that the same RQT structure is applied to CBs of all color components, except when the CB size is 8×8. For instance, consider a scenario with t_D = 1. Now, as the chroma sub-sampling format of the test video dataset is 4:2:0, the luma CB and corresponding chroma CB size in such a case would be 8×8 and 4×4 respectively. Thus, the luma CB can be transform coded as a single TB or further split into 4 TBs each of size 4×4. But, the chroma CB cannot be transform coded into TB of size 2×2 as HEVC does not support TBs of this size, although t_D value is one. Therefore, no split is permitted in a chroma CB, whenever the CB size is 8×8.

2.6.5 Picture Segmentation

The picture is segmented into slices to gain error robustness and achieve a high level of parallelism. Each slice can be as large as the entire picture itself or as small as a CTU i.e. there may be only one slice or multiple slices in a picture. Each slice follows the raster scan order of the CTUs in the picture for transmission. Every slice constitutes several non-overlapping slice segments. The first slice segment within the slice is always the independent slice segment. Apart from this one, there may be none or several dependent slice segments as depicted in Fig. 2.15 where CTUs shaded in $blue$ belong to the independent slice segments. Here, it is obvious that CTUs numbered $0 \rightarrow 59$ comprises of one independent slice segment and two dependent slice segments. The concept of slicing results in the partitioning of a picture in such a manner that each of the slices with the slice segments is independently decodable from the other slices belonging to the same picture. The slice segment header followed by the slice segment data together constitutes the coded slice segment. While the slice segment header holds the control information for the slice segment, the coded samples are loaded into the slice segment data.

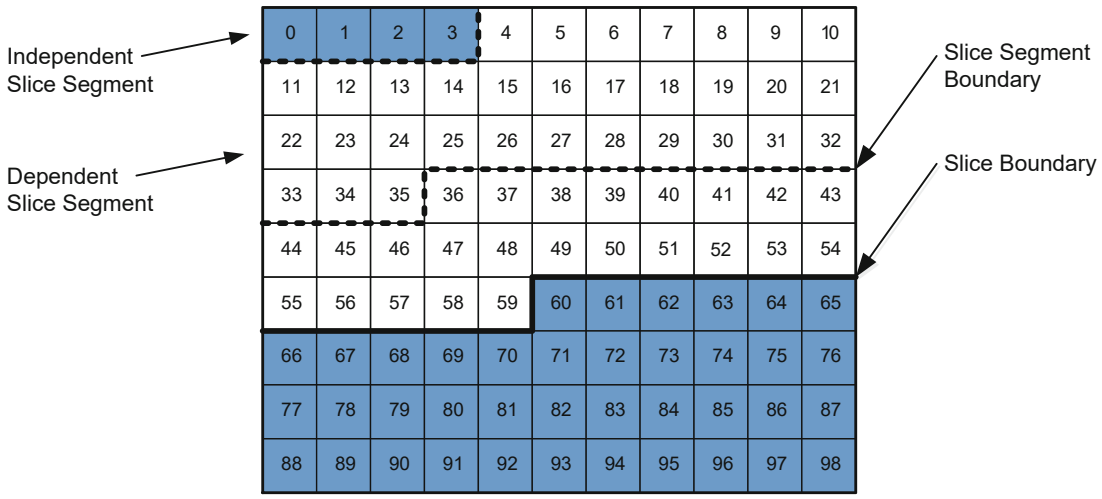


Figure 2.15: An illustration for partitioning a picture into slices and slice segments
Courtesy: Sze et al. (2014)

These slices are transmitted in the form of data packets, with the slice header information preceding the slice data. In the case of data packet loss, the ability to re-synchronize both the decoding and parsing process benefits to gain the error robustness. The slice and slice segments serve as powerful tools for recovery and re-synchronization in such cases. The slice header which holds the specific information for the decoding of the slice data is mainly associated with the independent slice, while a dependent slice does not contain much of the slice header information. Thus, to decode a dependent slice, the header information from the associated independent slice segment is very essential. On the contrary, the transmission of the slice header results in overhead, leading to a reduction in compression efficiency.

2.6.6 Parallelization Tools

To support high-level parallelization, HEVC is facilitated with two new coding tools, namely wavefront parallel processing and tiles. These tools permit multiple partitions within the picture to aid parallel processing.

2.6.6.1 Wavefront Parallel Processing

The HEVC specification includes the wavefront parallel processing (WPP) tool that permits synchronous entropy decoding of multiple CTU rows within a slice at the decoder. The syntax element $entropy\_coding\_sync\_enabled\_flag$ in the picture parameter set (PPS) is

responsible for the functioning of this tool. The slice segment header contains a list of byte-aligned entry points to the bitstream that indicates the beginning of the first CTU in each row.

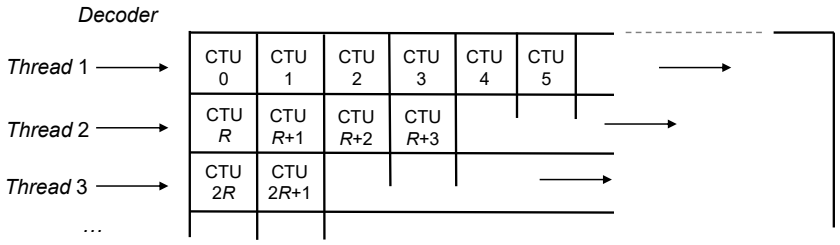


Figure 2.16: Parallel processing of a picture with R CTUs in a row using WPP

Enabling the WPP feature results in each CTU row of a picture to be treated as a separate partition, thus creating threads corresponding to each CTU row for parallel processing. Every thread would be processed in the same manner as the CTU row above it, but with a delay of two CTUs as shown in Fig. 2.16 that depicts R CTUs in each row. This technique was adopted to prohibit any dependencies between the consecutive CTU rows at the partition boundaries, except for the CABAC context variables at the end of each CTU row.

2.6.6.2 Tiles

Another parallel processing tool that is mentioned in the HEVC specification is the tiles. The syntax element $tiles\_enabled\_flag$ in the PPS signifies the usage of this tool. When this feature is enabled, the entire picture is divided into rectangular-shaped groups of CTUs separated by horizontal and/or vertical boundaries. One of the constraints must be true conjointly for a slice segment and tile i.e. either all CTUs present in a slice belong to the same tile or all CTUs in a tile belong to the same slice. The number of tiles and the relevant positioning of their boundaries can be defined in the PPS using the signaling parameters corresponding to the entire sequence or modified on a per picture basis.

The illustration in Fig. 2.17 depicts a scenario of a picture being divided into two tiles, with the number of slices and dependent slice segments as one and four respectively. The CTUs that belong to independent slice segments are shaded in $blue$. The bold dashed lines in the figure indicate the tile separation boundaries, while the dotted lines denote the separation of the slice segments. It also depicts that the tiles follow the tile-based raster scan order during the processing. In general, tiles provide a better compression efficiency in comparison to the slices as the quantum of the spatial correlation exploitation between the

samples of the same tile is more due to the reduction in the spatial distances in the tiles. On the contrary, the compression efficiency starts to deter when there is an increase in the number of tiles.

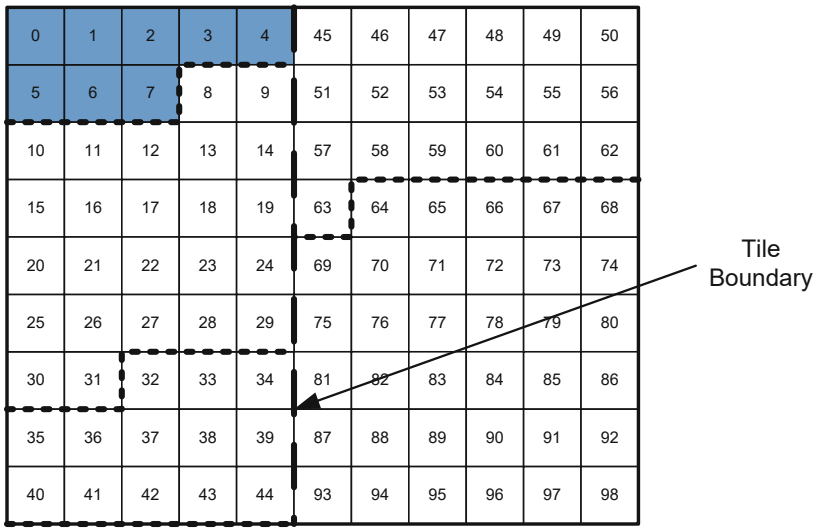


Figure 2.17: An illustration for a tile partitioning within a picture
Courtesy: Sze et al. (2014)

2.6.7 Transform and Quantization

HEVC makes use of the transform coding on the prediction error residuals resulting from the inter/intra predictions. Each TB then serves as an input to a 2D $L \times L$ forward transform at the encoder. The elements of the core transform matrices are derived by approximation of the scaled DCT basis functions. On the other hand, DST is used as an alternate 4×4 transform restricted to only luma transform blocks. This restriction is mainly due to the marginal enhancement in the compression efficiency for the remaining TB sizes. The resultant $L \times L$ transform coefficients are later quantized to round off the transform coefficients by dividing using the quantization step size (Q_{step}). Thus, the process of quantization is an inherently non-linear lossy operation, which cannot be reverted. The Q_{step} value is indicated by the quantization parameter (QP) and their relationship is described as in (2.1), where QP varies between 0-51.

$$Q_{step}(QP) = (2^{1/6})^{QP-4} \tag{2.1}$$

At the decoder, the first inverse quantization of the transform coefficients is performed,

followed by a 2D $L \times L$ inverse transformation. The residual block of quantizer scaled samples is derived on inverse transformation, which is then added to the intra/inter prediction samples to obtain the reconstructed block.

2.6.8 Entropy Coding

Entropy coding, a lossless compression mechanism makes use of the statistical properties for data compression in a manner that the number of bits used to represent the data is logarithmically proportional to the probability of the data. Context-based adaptive binary arithmetic coding (CABAC) is a method of entropy coding that is used in HEVC. The CABAC in its simplest form involves the key elements namely binarization, context modeling, and binary arithmetic coding, as the main algorithmic building blocks as shown in Fig. 2.18.

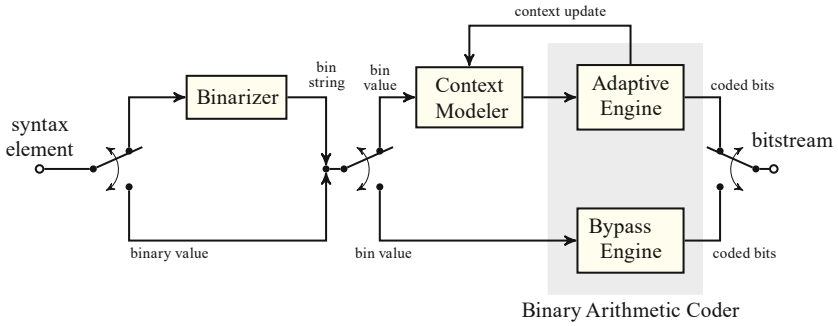


Figure 2.18: Block diagram of CABAC
Courtesy: Wein (2015)

2.6.9 In-loop Filter

The loop filtering is necessary to enhance the reconstruction quality of the picture to be displayed at the decoder. With an aim to mitigate the compression artifacts and enhance the compression efficiency, HEVC makes use of three in-loop filters, namely deblocking filter, sample adaptive offset filter and adaptive loop filter that are operated within the inter-picture prediction loop. Such an arrangement for improvisation not just impacts the quality of the output pictures, but also benefits the reference pictures stored in the buffer that are essential for the prediction process while coding the succeeding pictures.

2.6.10 Decoded Picture Buffer

The decoded reconstructed pictures are stored in the decoded picture buffer prior to their scheduled display. This process is to benefit the scenario when the decoding order of the pictures deviates from the output order of the pictures. The HEVC specification mentions the categorization of these reconstructed pictures i.e. pictures used for short-term referencing, pictures used for long-term referencing, and pictures unused for referencing. The pictures that are utilized for prediction in the current or a future picture of the coded video sequence will be collected in the reference picture set (RPS), where each picture is identified specifically using a picture order count (POC). POC is an index number that denotes the position of the current picture in the output of the coded video sequence. The pictures that are not part of RPS will be marked as pictures to be unused for referencing, thus eliminated from the buffer once displayed. At the onset of decoding a picture, the appropriate RPS will be constructed for all slices present in the picture.

2.7 HEVC Intra Prediction Mechanism

HEVC makes use of the conventional block-based prediction strategy and signals the prediction information inside the bitstream for every CU, indicative of whether the CU will be subjected to intra or inter prediction. The intra prediction strategy in HEVC supports 35 prediction modes i.e. planar ($M0$), DC ($M1$) and 33 angular ($M2 - M34$) modes, to handle the regions with gradual variation in intensities, uniform regions and regions with rich directional textures (Lainema et al. (2012))].

The planar ($M0$) mode in HEVC provides good approximations of the patterns with smooth variation in the luminance values. Additionally, it has the potential to overcome the discontinuities along the block boundaries. This is accomplished by sample-based averaging of the vertical and horizontal predictions as given in (2.2). Figure 2.19 illustrates the same using the previously reconstructed samples {a, b, c, d} as the reference samples for prediction of the target pixel $C_{x,y}$ at specified sample location [x,y] in a block of size $L \times L$.

$$\hat{P}[x,y] = (\hat{P}_h[x,y] + \hat{P}_v[x,y] + L) >> (log_2(L) + 1) \tag{2.2}$$

The horizontal prediction $\hat{P}_h[x,y]$ and vertical prediction $\hat{P}_v[x,y]$ in (2.2) are generated using (2.3) and (2.4). In general, $\hat{R}[x,y]$ represents the reference samples that belong to the blocks on the top (R_{above}) and left (R_{left}) of the current block.

$$\hat{P}_h[x,y] = [(L-1-x) * \hat{R}[-1,y] + (x+1) * \hat{R}[L,-1]] \tag{2.3}$$

$$\hat{P}_v[x,y] = [(L-1-y) * \hat{R}[x,-1] + (y+1) * \hat{R}[-1,L]] \tag{2.4}$$

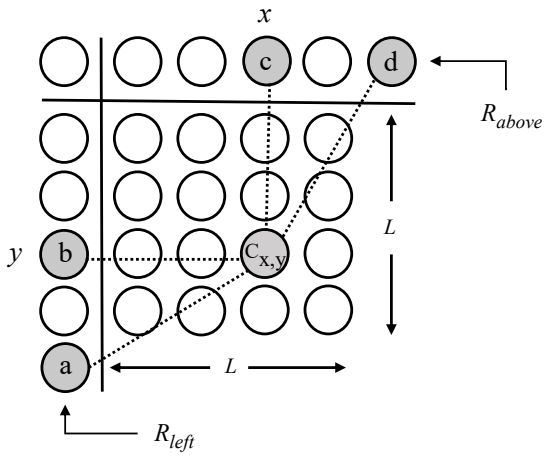


Figure 2.19: An illustration for planar prediction at the sample location $[x, y]$

The DC ($M1$) prediction mode is meant to handle the uniform regions. The generated prediction value in (2.5) represents the average of the $2L$ reference samples from the previously reconstructed neighboring blocks on the top (R_{above}) and left (R_{left}) of the current block as shown in Fig. 2.20.

$$\hat{P}[x,y] = \frac{1}{2L}(\sum_{x=0}^{L-1} \hat{R}[x,-1] + \sum_{y=0}^{L-1} \hat{R}[-1,y]) \tag{2.5}$$

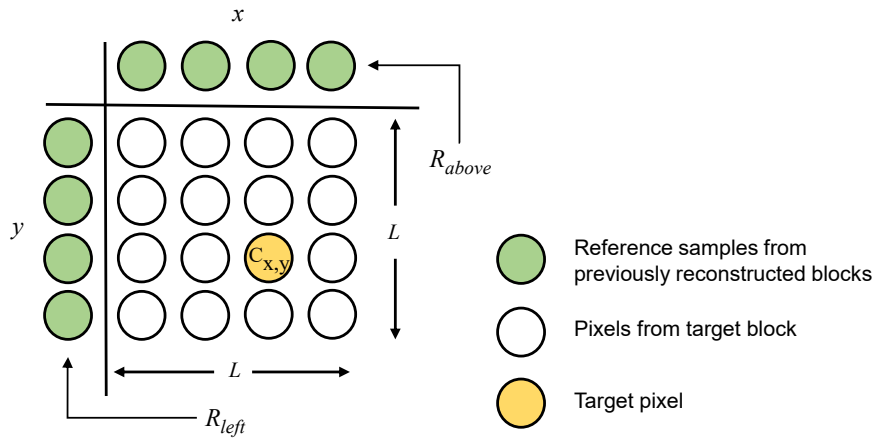


Figure 2.20: DC prediction strategy

The 33 angular modes aim to generate suitable predictions for directionally oriented structures and patterns. They are broadly categorized as vertical ($M18 - M34$) and horizontal ($M2 - M17$) modes. Each of these prediction modes is positioned at spatial increments of the displacement parameter (θ_{disp}) i.e. +/-$\{2, 5, 9, 13, 17, 21, 26, 32\}$ with respect to the vertical (V) and horizontal (H) axis, defined on a 32-sample grid. Here, +ve offset is associated with the positive (($M27 - M34$) and ($M2 - M9$)) modes, while -ve offset corresponds to the negative (($M25 - M18$) and ($M11 - M17$)) modes. The distribution of θ_{disp} values over the 32-sample grid shown in Fig. 2.21 reveals an increased resolution of the prediction angles around the vertical/horizontal directions and a coarser resolution towards the diagonal orientations.

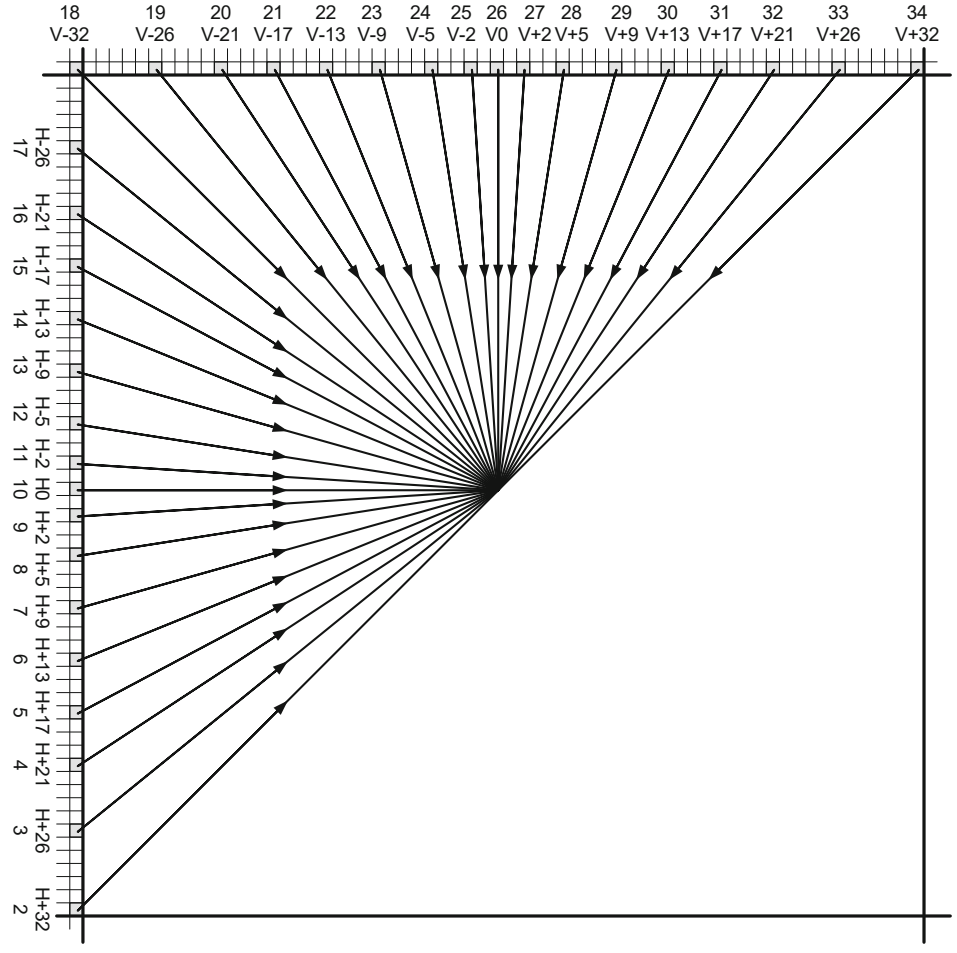


Figure 2.21: Angular intra prediction modes of HEVC with the angle definitions
Courtesy: Lainema et al. (2012)

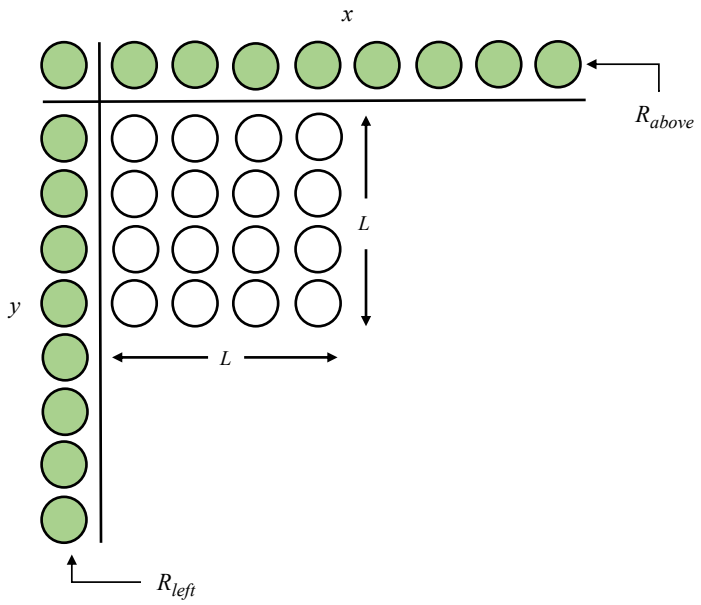


Figure 2.22: Reference samples $R_{samples}$ around block of size $L \times L$

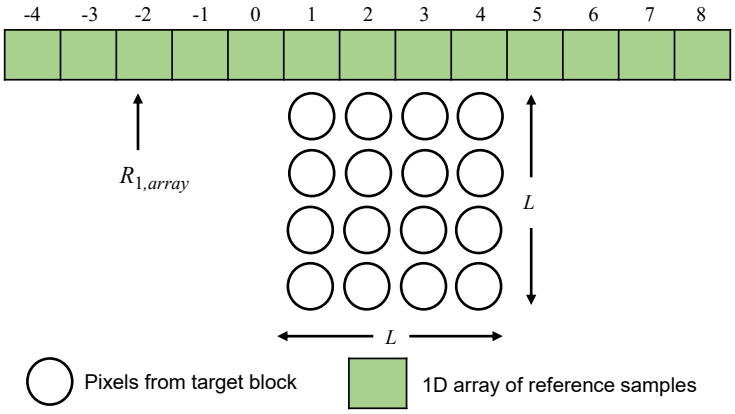


Figure 2.23: 1D array of reference samples $R_{1,array}$ constructed using $R_{samples}$

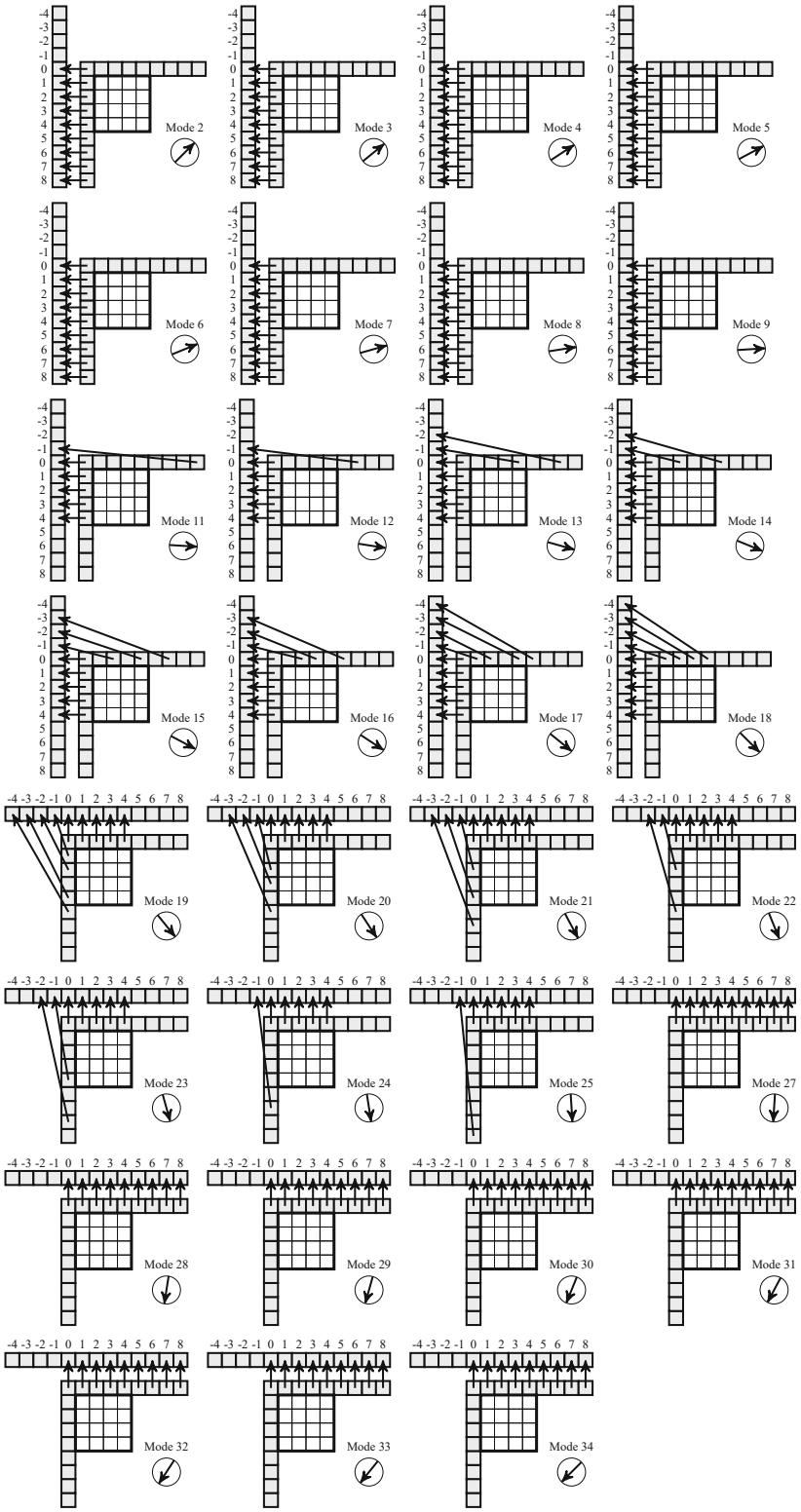


Figure 2.24: Formation of 1D array $R_{1,array}$ for the angular modes ($M2 - M34$)
Courtesy: Wein (2015)

The prediction process essentially requires an initial formation of a 1D array of samples $R_{1,array}$ from the $R_{samples}$ that belongs to the previously reconstructed coding blocks in the closest proximity as shown in Fig. 2.22. The 1D array of samples $R_{1,array}$ constructed using $R_{samples}$ is as shown in Fig. 2.23. Now, the choice of the reference samples that are to be loaded into the array is based on the specified angular direction. Figure 2.24 helps to visualize the formation of the 1D array of reference samples for the prediction process in case of horizontal and vertical prediction respectively. From the figures, it is evident that positive modes ($M2 - M10$) and ($M26 - M34$) utilizes the reference samples from the reconstructed blocks on the left (R_{left}) and top (R_{above}) respectively, while negative modes ($M11 - M25$) make use of the reference samples from the reconstructed blocks located on the top and left side of the target block.

Once the 1D reference array is constructed, then the prediction value $\hat{P}[x, y]$ is generated using the pixel interpolation of the two reference samples present in $R_{1,array}$ i.e. R_i and R_{i+1} as specified in (2.6) at a sample position accuracy of (1/32). Figure 2.25 illustrates the process of prediction generation in case of vertical angular prediction for a positive displacement parameter.

$$\hat{P}[x, y] = (((32 - w_y) * R_i + w_y * R_{i+1}) + 16) >> 5 \tag{2.6}$$

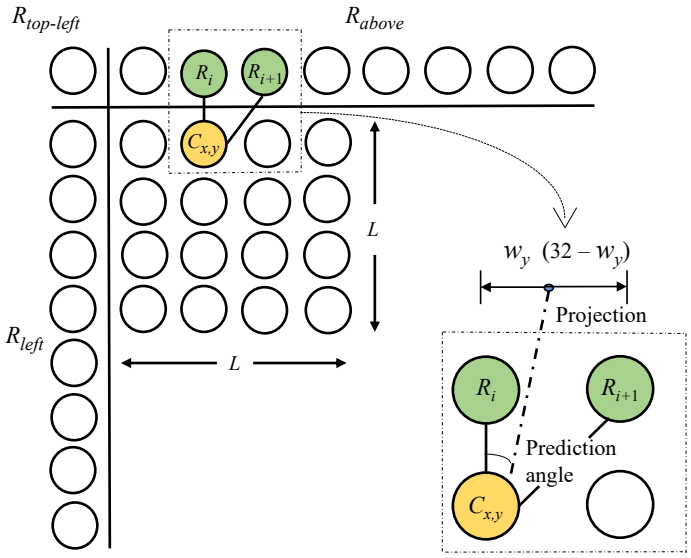


Figure 2.25: Vertical angular prediction process in HEVC

The weighted coefficient (w_y) is computed as given in (2.7) and the operator $>>$ per-

forms the right bit-shift operation. While the variables x and y represent the sample coordinates varying between 0 and $(L-1)$, the coefficient w_y is indicative of a fractional value that is pointed to by the projected displacement which lies between the two reference samples R_i and R_{i+1}.

$$w_y = ((y+1) \cdot \theta_{disp}) \ \& \ 31 \tag{2.7}$$

The reference sample index i is computed using (2.8), where the parameter C_y is derived as provided in (2.9). As the value of θ_{disp} increases, it is evident that the integer value of i increases, resulting in the positional shifting of the reference sample pair along the specified direction.

$$i = C_y + x + 1 \tag{2.8}$$

$$C_y = ((y+1).\theta_{disp}) >> 5 \tag{2.9}$$

To generate the prediction corresponding to a horizontal mode, index y needs to be replaced with x and vice versa in all the computations involved in (2.6)–(2.9). The block-based intra prediction scheme in HEVC is more generalized in comparison with its predecessor H.264. The 35 intra prediction modes are applicable to all supported PB sizes ranging between 4×4 and 32×32. Of these intra prediction modes, DC mode is the least computationally expensive mode as the PB is filled with values derived by averaging the reference samples. The planar mode emerges as the most computationally expensive mode since the encoder performs a bidirectional interpolation function to fill the PB. In the case of angular modes, linear interpolation is done in the chosen direction to fill the PB.

Once the prediction mode has been finalized for a PU, it needs to be signaled efficiently with minimum overhead through intra mode coding (da Silva et al. (2014)). Prior to the final selection of the prediction mode based on rate-distortion optimization (RDO), the three most probable modes (MPMs) are derived to derive the luma intra mode. Meanwhile, HEVC introduces the concept of $derived\ mode$ for the chroma intra mode, which permits efficient signaling of the same prediction mode used by the luma as a chroma prediction mode. The $derived\ mode$ enables the usage of the increased number of directionalities for chroma intra prediction and minimizes the signaling overhead. Thus, any chroma PU is associated with only five chroma intra prediction modes i.e. planar, angular vertical mode ($M26$), angular horizontal mode ($M10$), DC and $derived\ mode$.

Any redundancy in the mode selection is carefully monitored and will be replaced with alternate ones to maximize the signaling efficiency. The selection of the MPMs for every PU is mainly based on the modes of the two neighboring PUs i.e. PU on the left and PU on

the top. If one of the neighboring blocks is not available, then the target PU will be predicted using DC mode. Assume that the PUs on the left and top are predicted using the modes p and q respectively. The three MPMs for the target PU are denoted by MPM[0], MPM[1] and MPM[2]. Derivation of the MPMs is mainly based on the p and q modes assigned to the neighboring blocks. In general, the algorithm for the derivation of the candidate modes can be summarized as below.

When p is not equal to q, MPM[0] = p and MPM[1] = q. The third candidate mode MPM[2] will be based on:

- If p and q are not planar modes, MPM[2] is assigned as planar mode ($M0$)
- If p and q are not DC modes, MPM[2] is assigned as DC mode ($M1$)
- Otherwise, MPM[2] is assigned as vertical angular mode ($M26$)

In case of p being equal to q, then the MPMs are derived based on:

- If p and q are either planar or DC prediction mode, then candidate list appears as:
  MPM[0] = planar mode ($M0$)
  MPM[1] = DC mode ($M1$)
  MPM[2] = vertical angular mode ($M26$)
- If p and q are neither planar nor DC prediction modes, then candidate list appears as:
  MPM[0] = p
  MPM[1] = 2 + (p + 29) mod32
  MPM[2] = 2 + (p - 1) mod32

Apart from the MPMs, the number of luma intra mode candidates that enter the process of full RDO is based on their PU sizes i.e. eight for PU sizes 4×4, 8×8 and three for the remaining PU sizes. Rough mode decision (RMD) process is adopted to minimize the number of intra mode candidates that are being evaluated in the RDO process. Here, the entries of the best-suited candidate modes rely on the sum of absolute transformed differences (SATD). Finally, the intra mode which renders minimum RD cost is chosen as the PU prediction mode.

2.8 Simulation Environment

The JCT-VC has provided a common reference platform in the form of a basic HEVC framework for the validation of the proposed algorithms. The software commonly termed as the

HEVC test model intends to mimic the encoder and decoder. The source code has a modular structured approach written using C++ programming language. The proposed algorithms need to be embedded into this framework through a suitable integrated development environment like Microsoft Visual Studio, Codeblocks etc., which helps to modify the source code and debug the modification. The unmodified version of the framework is referred to as the HEVC anchor wherever relevant. For a fair comparison of the proposed methodologies, the validation process is conducted on the test dataset of the video sequences that possess varying display resolutions and rich textural patterns. Also, the parameters essential to evaluate the performance of the designed algorithms are discussed in the forthcoming sections. Although the HM software renders a sluggish environment, it serves as a testbed for the algorithmic evaluation.

2.8.1 Test Video Sequences

The HEVC test dataset comprises of color video sequences (Test video sequences (2012)) that are represented using the YUV color model. The sequences are subjected to 4:2:0 chroma sub-sampling and possess a bit-depth of 8. They are classified into six classes based on the textural and motion properties along with varying display attributes. The Class A-E sequences represent naturally captured video sequences, while Class F sequences are either computer-generated or SC video sequences. Additionally, another class of natural sequences commonly referred to as the 4K sequences have been considered for the algorithmic evaluation. The details on the various classes of the video sequences are provided in Table 2.2.

Table 2.2: Attributes of HEVC test video sequences

Class	Resolution	Category	# Sequences
A	2560 x 1600	4K x 2K ultra-HD at 30 fps (cropped)	2
B	1920 x 1080	1080p HD at 24, 60 fps	3
C	832 x 480	WVGA at 30, 50, 60 fps	4
D	416 x 240	WQVGA at 30, 50, 60 fps	4
E	1280 x 720	720p at 60 fps	4
F	832 x 480 1024 x 768 1280 x 720	WVGA at 50 fps XGA at 30 fps 720p at 20 fps	3
4K	3840 x 2160	4K x 2K ultra-HD at 120 fps	7

Table 2.3: HEVC test video sequences

Class	Video Sequence	No. of frames	Frame Rate (fps)
A	Traffic	150	30
	People on Street	150	30
B	Kimono	240	24
	Park Scene	240	24
	BQ Terrace	600	60
C	Race Horses	300	30
	BQ Mall	600	60
	Party Scene	500	50
	Basketball Drill	500	50
D	Blowing Bubbles	500	50
	Race Horses	300	30
	BQ Square	600	60
	Basketball Pass	500	50
E	Vidyo1	600	60
	Vidyo3	600	60
	Vidyo4	600	60
	Johnny	600	60
F	Basketball Drill Text	500	50
	China Speed	500	30
	Slide Show	500	20
4K	Beauty	600	120
	Bosphorus	600	120
	Honey Bee	600	120
	Jockey	600	120
	Ready Steady Go	600	120
	Shake & Dry	300	120
	Yacht Ride	600	120

The classwise list of video sequences that are chosen for the algorithmic validation is specified in Table 2.3. The first still image of the natural and computer-generated/SC video sequences is presented in Appendix I. The Class F sequences can be further categorized as text and graphics with motion (TGM), camera-captured content (CC), mixed content (M) and animation (A). Here, the video sequences categorized as M are a combination of TGM and CC. To note that, the validation is performed on the initial 100 frames of the test sequences, except in case of Class 4K sequences where validation is restricted to the initial 10 frames, owing to the timing constraints.

2.8.2 Reference Software

The HEVC test model (HM), a reference software (Rosewarne et al. (2016)) provided by the JCT-VC serves as a common platform to incorporate the proposed algorithms and validate them under the specified standard test conditions presented in Bossen (2012) and Haoping et al. (2015). As the test data set comprises of natural and computer-generated/SC video sequences, the validation is performed using the software version HM 16.12 + SCM 8.3 located in the repository (Fraunhofer (2016)). Here, HM 16.12 represents the basic HEVC framework, while the integrated add-on module commonly referred to as the screen content module (SCM) plays a significant role in handling the Class F video sequences.

In general, the details on the coding tools included to create the bitstream are mentioned in the low computational complexity test settings [8-bit] i.e. Main and Main_SCC for the natural and SC sequences respectively. The Main profile supports four configurations i.e. all intra (AI), lowdelay B (LB), lowdelay P (LP) and random access (RA). In the AI configuration setting, every frame is encoded as an I-picture i.e. intra coded. As there is no inter prediction, this setting is best-suited for applications that demand low delay and higher bitrate. Now, apart from the first frame being intra coded, the remaining frames are encoded as B- and P-pictures in case of LB and LP configurations settings respectively. However, in the case of RA configuration setting, in addition to the first frame being intra coded several I-pictures are periodically inserted between B-pictures to curb the error propagation, owing to re-ordering of the pictures. The Main_SCC profile does not extend support to the LP configuration, hence the validation of Class F sequences is restricted to only three configurations i.e. AI, LB, and RA.

For the aforementioned profiles, the following aspects related to the source video sequence needs to be priorly defined.

- Location of the source video sequence in the test system
- Frame rate
- Source width
- Source height
- Number of frames to be encoded
- Quantization parameter
- Internal bit-depth

2.8.3 Test Environment

Lossless coding can be achieved by bypassing the processing units such as transformation, quantization and its inverses, which affects the faithful reconstruction of the image. In this manner, the residual signal is directly transmitted to the entropy coding stage that results in distortion-free reconstructed frames. Figure 2.26 provides the detailed encoder structure of HEVC in lossless mode.

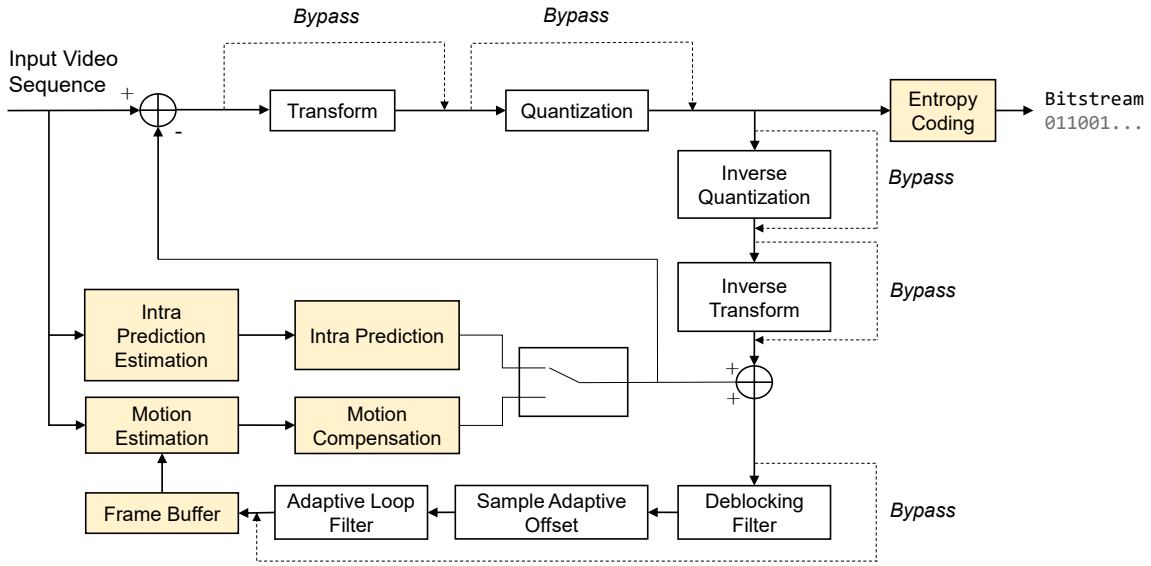


Figure 2.26: Block diagram of the HEVC encoder in lossless mode

Table 2.4: Simulation environment and test conditions

CPU	2.30 GHz Intel (R) Core (TM) i5-6200U processor
RAM	8 GB
OS	Windows 10
Profile/Configuration	Main - AI, LB, LP, RA Main_SCC - AI, LB, RA
QP	0
$TransquantBypassEnable$	1
$CUTransquantBypassFlagForce$	1

Now, to invoke the lossless mode in HEVC, certain modifications are to be incorporated within the Main and Main_SCC profile. As the quantization stage does not play any role in

lossless coding, the QP value is set to 0 and status flags i.e. $TransquantBypassEnable$ and $CUTransquantBypassFlagForce$ indicated in the PPS are set to 1 in the Main and Main_SCC profiles. The entropy coding block will directly procure the generated residual samples since the transformation, quantization and in-loop filtering stages are bypassed in the lossless mode. In addition to these settings, the SC video sequences need certain other modifications in the Main_SCC profile as specified in Haoping et al. (2015). The hardware and software specifications are provided in Table 2.4. Once all the software settings are done and the executables being kept ready, the system console is used to initiate the process of encoding and decoding through suitable instructions (Rosewarne et al. (2016)).

2.8.4 Evaluation Parameters

Whenever coding is performed in lossless mode, the reconstruction is perfect i.e. distortion-free. Hence, it is not required to perform the objective analysis to determine the video quality in case of lossless coding. However, the performance of the proposed algorithm can be analyzed by deriving the $savings$ in terms of bit-rate and run-time, computed using (2.10) and (2.11) respectively.

$$BR_{savings} = \frac{(Bit\_rate_{anchor} - Bit\_rate_{proposed})}{Bit\_rate_{anchor}} * 100\% \tag{2.10}$$

$$Time_{savings} = \frac{(Time_{anchor} - Time_{proposed})}{Time_{anchor}} * 100\% \tag{2.11}$$

Additionally, distribution of the intra prediction modes, coded block distribution for various PU sizes and residual analysis is performed at the decoder side to strengthen the algorithmic performance validation.

Chapter 3

GRADIENT-BASED PLANAR AND ANGULAR PREDICTIONS

3.1 Introduction

The chapter presents gradient-based strategies that focus on the enhancement of the prediction mechanism of the planar and angular intra prediction modes using - (a) threshold-controlled gradient adaptive planar prediction (TGAPP) and (b) gradient-oriented selection with a sample-based weighted angular prediction (GSSWAP). The TGAPP and GSSWAP jointly referred to as gradient dependent prediction (GDP) intends to enhance the overall compression efficiency of the planar and angular intra prediction modes of HEVC. TGAPP, a planar prediction modification features a low complexity mechanism to identify the edges within the frame. Meanwhile, GSSWAP aims to improve the prediction quality of the directional structures within the given frame.

With an increase in the display resolution of the video sequences, larger regions with smooth variations and regions with several intensity variations become quite significant. To achieve efficient compression, the chosen prediction technique must be capable to tune the entire prediction process based on the local region characteristics. Sample-based prediction methods sometimes fail to cater to the capturing of the edge details especially when they are inclined. The predictors presented in this chapter are designed to identify the existence of vertical/horizontal/diagonal edges along with the gradual gradient variation and carry out the prediction process based on the gathered gradient information around the target pixel.

3.2 Threshold-controlled Gradient Adaptive Strategy for Planar Prediction

Threshold-controlled gradient adaptive planar prediction (TGAPP) is a simple blend of the median and gradient adaptive predictors. It primarily computes the gradient values in the vertical and horizontal directions for every target sample within the coding block. Next, the prediction value is derived based on the threshold value (T). Therefore, the local gradient estimation and predefined threshold value emerge as essential parameters to derive an optimal predictor from the sub-predictors. The prediction depends on the 2D image behaviors i.e. whether the pixel is in the context of a horizontal edge, vertical edge or the smooth region. The pseudocode for the proposed algorithm is presented in Algorithm 1 that replaces the conventional planar mode embedded in the HEVC anchor.

Algorithm 1 Pseudocode for TGAPP intra prediction

Compute vertical (G_V) and horizontal (G_H) gradients:

$$G_V = \mid NW - W \mid + 3 \; . \mid NN - N \mid$$

$$G_H = \; 3 \; . \mid WW - W \mid + \mid NW - N \mid$$

if (G_V - $G_H > T$) **then**
  $\hat{P}[x, y]$ = W
else if (G_V - $G_H <$ -T) **then**
  $\hat{P}[x, y]$ = N
else
  $\hat{P}[x, y]$ = (N + W - NW)
end if

In the first phase, the pixel-level computation of the vertical (G_V) and horizontal (G_H) gradients relative to the target pixel $C_{x,y}$ is derived using the five causal pixels located at $\{N, W, NW, NN, WW\}$ directional positions as shown in Fig. 3.1.

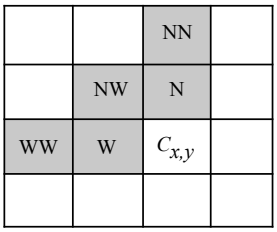


Figure 3.1: Template of the causal samples for gradient computation in TGAPP

Gradient computation is highly essential to capture the level of activity around the target pixel in terms of smoothness, edginess, etc., thus governing the statistical behavior of the prediction residuals. Now, whenever a moderate edge is not detected, then the predictor ($N + W - NW$) is chosen to mainly benefit the region with smooth variations within an image. The prediction value is then derived, based on the comparison between the gradient difference and threshold T. The value of T was chosen to be 32 after an experimental analysis to identify moderate edges. This value was later fixed for the algorithm evaluation performed using the set of HEVC test sequences.

3.3 Gradient-oriented Selection with Sample-based Weighted Strategy for Angular Prediction

The gradient-oriented selection with a sample-based weighted angular prediction (GSSWAP) strategy is initially carried out by averaging the gradients (G_i) around the target pixel in the four specified directions: horizontal (0°), vertical (90°), diagonal (45°) and anti-diagonal (135°). Next, the weighted averages of the nearby causal pixels corresponding to the directional gradient are computed to generate the prediction value for the target pixel. A template consisting of ten causal pixels is considered to estimate the gradient around the target pixel $C_{x,y}$ as elaborated in Fig. 3.2 and Fig. 3.3 for vertical and horizontal angular predictions respectively.

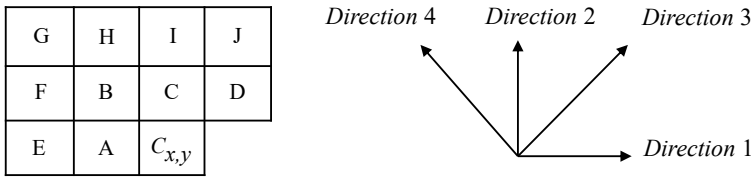


Figure 3.2: Causal pixel template for vertical prediction

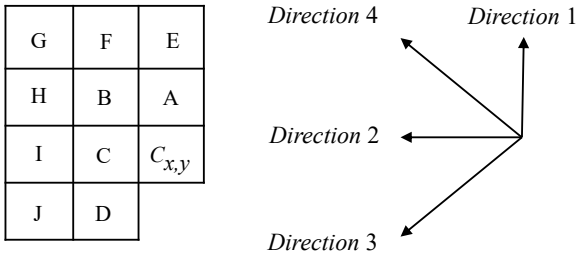


Figure 3.3: Causal pixel template for horizontal prediction

For easy and better understanding, the neighboring pixels are represented using alphabets from A - J and the four directions are denoted using i = 1, 2, 3, 4 corresponding to horizontal, vertical, diagonal and anti-diagonal orientations respectively. The gradients ($\hat{G}_i$) are estimated by interpolation of the absolute difference between the causal pixels in the specified four directions using (3.1)–(3.4).

$$Direction\ 1: \quad \hat{G}_1 = \mid A - E \mid + \mid C - B \mid + \mid D - C \mid + \mid B - F \mid \tag{3.1}$$

$$Direction\ 2: \quad \hat{G}_2 = \mid A - B \mid + \mid C - I \mid + \mid D - J \mid + \mid B - H \mid \tag{3.2}$$

$$Direction\ 3: \quad \hat{G}_3 = \mid A - C \mid + \mid C - J \mid + \mid E - B \mid + \mid B - I \mid \tag{3.3}$$

$$Direction\ 4: \quad \hat{G}_4 = \mid A - F \mid + \mid C - H \mid + \mid D - I \mid + \mid B - G \mid \tag{3.4}$$

The estimated gradient values are now averaged in order to derive the final gradient values (G_i) using (3.5)–(3.8).

$$G_1 = \lfloor \hat{G}_1/4 + 0.5 \rfloor \tag{3.5}$$

$$G_2 = \lfloor \hat{G}_2/4 + 0.5 \rfloor \tag{3.6}$$

$$G_3 = \lfloor \hat{G}_3/4 + 0.5 \rfloor \tag{3.7}$$

$$G_4 = \lfloor \hat{G}_4/4 + 0.5 \rfloor \tag{3.8}$$

The generated prediction value is based on the estimated gradient values. This is accomplished by assigning a higher weight to the immediate causal pixel in the direction of the lower estimated gradient value. The pixel at the minimal pixel distance from $C_{x,y}$ will be the relative pixel to the corresponding directional gradient. Thus, the horizontal and vertical angular predictions are carried out by assigning the immediate causal neighborhood pixels C = {A, C, D, B} as the relative causal pixels corresponding to the gradients G_1, G_2, G_3 and G_4 respectively, elaborated in Fig. 3.2 and Fig. 3.3.

While the displacement parameter (θ_{disp}) is 0 or +32 or -32, two conditions are evaluated. Firstly, when there is only one non-zero gradient, the relative causal pixel in the direction perpendicular to the current directional gradient serves as the prediction value. Next, in the case of more than one non-zero gradients, the prediction value is computed based on the generation of the sample-based weights corresponding to the relative causal neighbors in the gradient direction. In all the remaining cases, the SAP prediction algorithm is applied to the target pixel to generate the prediction value as explained in Section 3.3.1.

The prediction process makes use of the weighted averaging of the surrounding pixels

to predict the target pixel. The prediction value is realised using (3.9).

$$\hat{P}[x,y] = \lfloor \frac{\sum_{m \in K} wt_m \cdot h[m]}{\sum_{m \in K} wt_m} \rfloor \tag{3.9}$$

where $\hat{P}[x,y]$ is the generated prediction and K indicates the number of causal neighbors in C that are involved in the prediction process. The parameter wt_m that represents the integer weights of the relative causal pixels in C is computed using (3.10) as in Wige et al. (2013).

$$wt_m = \lfloor a \cdot b^{-\frac{SAD(P_h^g[m], P_h^g[n])}{h_{dist}}} \rfloor \tag{3.10}$$

Here, a is a factor that typically ensures the major weights are non-zero, b is the basis of the exponentially decaying function for weights which is set to two, $P_h^g[m]$ is a patch around the pixel $h[m]$, $P_h^g[n]$ is a patch around the pixel $h[n]$ and h_{dist} is a prediction modeling parameter. The value of a is derived using (3.11), where the parameter β denotes the internal bit-depth of the video sequence. The prediction process is implemented using 32-bit integer arithmetic. Since four samples are averaged for prediction generation in this implementation, it has to be ensured that the weighted summation in the numerator of (3.9) can also be represented using 32-bit integer arithmetic.

$$a = 2^{31-2-\beta} \tag{3.11}$$

The prediction modeling parameter h_{dist} is dependent on the sub-sampling format of the chroma component. The h_{dist} value for the luma and chroma component is empirically chosen as 5.25 and 3.25 respectively. The sum of absolute differences (SAD) between the two patches associated with the four immediate causal pixels is calculated using (3.12).

$$SAD(P_h^g[m], P_h^g[n]) = \sum_{m,n \in M} \mid h[m] - h[n] \mid \tag{3.12}$$

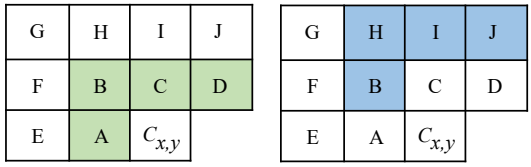


Figure 3.4: Patch corresponding to the target pixel $C_{x,y}$ and causal pixel C

In an illustration for vertical prediction, the patches for the computation of SAD value corresponding to the causal pixel C is as shown in Fig. 3.4. For the specified causal pixel, $SAD(P_h^g[m], P_h^g[n])$ value is computed using the sum of absolute differences between the two co-located pixels within the patch i.e. $P_h^g[m]$ = {A, B, C, D} and $P_h^g[n]$ = {B, H, I, J}. Here M is four, referring to the number of pixels within the considered patch.

In scenarios where the computed weights are zero, the prediction is based on the causal pixel corresponding to the minimum valued directional gradient. Once the minimum directional gradient G_{min} is determined using (3.13), then the corresponding relative causal pixel represented as C_{Gmin} will be assigned as the prediction value based on (3.14). In an illustration where the gradient in the vertical direction has minimal magnitude i.e. $G_{min} = G_2$, the relative causal pixel C_{Gmin} = C will be the generated prediction.

$$G_{min} = min(G_i), \quad where\ i = 1, 2, 3, 4 \tag{3.13}$$

$$\hat{P}[x, y] = C_{Gmin} \tag{3.14}$$

3.3.1 Prediction of the Boundary Pixels

In GSSWAP, when one of the two rows/columns of pixels essential for the prediction of the target pixel in the first row/column of the current block is unavailable, the prediction is generated using the SAP prediction technique. This strategy is an enhancement to the conventional block-based intra prediction technique embedded within the HEVC anchor. It is observed that the reconstructed pixels provide a precise prediction of the pixel to be predicted. Hence, the current row/column of pixels are predicted from the previously reconstructed row/column of pixels. Every target pixel $C_{x,y}$ is predicted using two reconstructed samples R_i and R_{i+1}. In general, to predict a single row/column of an $L \times L$ block, L + 2 reference samples are essential.

Parameter $fract$ is used to determine the two reconstructed reference samples i.e. R_i and R_{i+1} by projecting $C_{x,y}$ onto the reference row/column as in Fig. 3.5. Once the reference samples are chosen, then the prediction is computed using (3.15) which is a linear interpolation of R_i and R_{i+1}. Here, operator >> denotes the right bit-shift operation and the parameter $fract$ represents the distance measured at (1/32)nd pixel accuracy between R_i and the point where the projection from $C_{x,y}$ meets the line that joins R_i and R_{i+1}.

$$\hat{P}[x, y] = ((32 - fract) * R_i + fract * R_{i+1} + 16) >> 5 \tag{3.15}$$

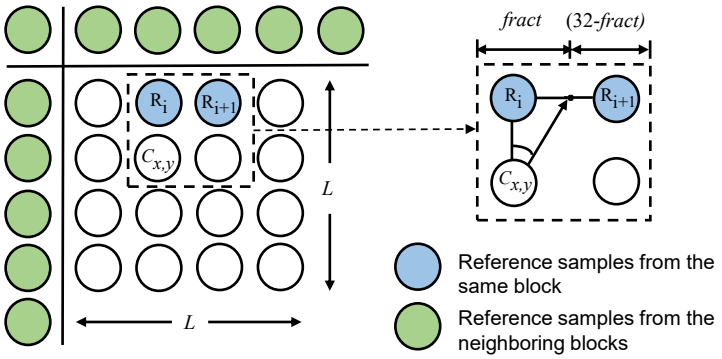


Figure 3.5: SAP based prediction for positive vertical prediction angle

When the boundary pixels in the last row/column are predicted, the prediction value is simply extended to fill the next location in the array to provide the R_{i+1} sample for the prediction of the subsequent row/column of samples. The pseudocode in Algorithm 2 summarizes the integer implementation of the proposed method.

Algorithm 2 Pseudocode for GSSWAP intra prediction

$fract = ((y+1).\theta_{disp})$ & (32 - 1) // vertical modes
$fract = ((x+1).\theta_{disp})$ & (32 - 1) // horizontal modes
if $(x = 0 \,||\, y = 0)$ **then**
  $\hat{P}[x,y] = ((32 - fract) * R_i + fract * R_{i+1} + 16) >> 5$
else
  $\hat{G}_1 = |\, A - E \,| + |\, C - B \,| + |\, D - C \,| + |\, B - F \,|$
  $\hat{G}_2 = |\, A - B \,| + |\, C - I \,| + |\, D - J \,| + |\, B - H \,|$
  $\hat{G}_3 = |\, A - C \,| + |\, C - J \,| + |\, E - B \,| + |\, B - I \,|$
  $\hat{G}_4 = |\, A - F \,| + |\, C - H \,| + |\, D - I \,| + |\, B - G \,|$

  Obtain averaged gradient values: G_1, G_2, G_3 and G_4

  if $(\sum_{i=1}^{4} G_i \neq 0 \;\; \&\& \;\; \theta_{disp} = (0 \,||\, +32 \,||\, -32))$ **then**
    if $(G_i \neq 0 \,|_{i=s} \;\; \&\& \sum_{i=1}^{4} G_i = 0 \,|_{i \neq s})$ **then**
      $\hat{P}[x,y] = C[j];\; where\, j \perp i$
    else
      if $(wt_A + wt_B + wt_C + wt_D = 0)$ **then**
        $\hat{P}[x,y] = C_{Gmin}$
      else
        $\hat{P}[x,y] = \dfrac{A.wt_A + B.wt_B + C.wt_C + D.wt_D}{wt_A + wt_B + wt_C + wt_D}$
      end if
    end if
  else
    $\hat{P}[x,y] = ((32 - fract) * R_i + fract * R_{i+1} + 16) >> 5$
  end if
end if

3.4 Simulation Results, Analysis and Discussions

The savings in bit-rate achieved for the standalone planar (TGAPP) and angular (GSSWAP) predictors and their combination are tabulated in Table 3.1 and Table 3.2. The findings in these tables reveal that the combined performance of the proposed planar and angular predictor i.e. GDP predictor dominates the behavior of the standalone predictors. These tables state that the combined predictor in comparison with the HEVC anchor provides a significant enhancement in compression efficiency by about 8.49%, 1.86%, 2.14%, and 2.38% on an average, in terms of overall bit-rate savings for Main - AI, Main - LB, Main - LP, and Main - RA profile configurations respectively. The overall bit-rate savings derived in comparison with the HEVC anchor is depicted in Fig. 3.6. It can be noticed that a higher percentage of improvement in savings is observed for the AI configuration compared to LB, LP, and RA configurations. This is due to the fact that the AI configuration utilizes only spatial correlation, while the LB, LP, and RA use spatio-temporal correlation.

Table 3.1: Bit-rate savings (%) for the Main - AI and Main - LB profile configurations

Class	**AI**			**LB**		
	TGAPP	**GSSWAP**	**GDP**	**TGAPP**	**GSSWAP**	**GDP**
A	2.63	10.58	11.27	1.25	2.84	3.01
B	2.56	6.22	7.17	0.82	1.48	2.01
C	1.55	5.18	5.37	0.59	1.56	1.64
D	2.55	7.56	8.19	0.68	1.32	1.39
E	3.87	8.51	10.45	0.42	0.88	1.25
Average (%)	**2.63**	**7.61**	**8.49**	**0.75**	**1.62**	**1.86**

Table 3.2: Bit-rate savings (%) for the Main - LP and Main - RA profile configurations

Class	**LP**			**RA**		
	TGAPP	**GSSWAP**	**GDP**	**TGAPP**	**GSSWAP**	**GDP**
A	1.42	3.33	3.53	1.44	3.28	3.47
B	0.95	1.97	2.50	0.98	1.83	2.37
C	0.62	1.68	1.79	0.68	1.83	1.92
D	0.71	1.40	1.48	0.86	1.86	1.94
E	0.62	1.28	1.38	0.85	1.69	2.20
Average (%)	**0.86**	**1.93**	**2.14**	**0.96**	**2.10**	**2.38**

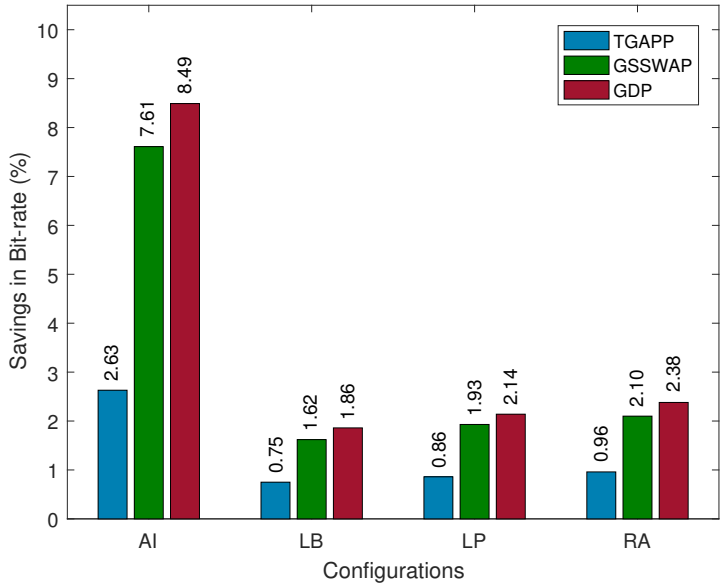


Figure 3.6: Graphical representation of the bit-rate savings in the standalone predictors and their combinations for the Main profile

Table 3.3: Intra prediction mode distribution (%) in the HEVC anchor and GDP predictor

Class	**Video Sequence**	**Anchor**			**GDP**		
		Planar (M0)	**DC (M1)**	**Angular (M2-M34)**	**Planar (M0)**	**DC (M1)**	**Angular (M2-M34)**
A	Traffic	12.8	6.1	81.1	11.1	1.7	87.2
	People on Street	12.1	7.2	80.7	12.4	2.2	85.4
B	Kimono	21.1	8.7	70.2	4.9	11.8	83.3
	Park Scene	15.7	11.1	73.2	7.7	12.9	79.4
	BQ Terrace	8.6	6.5	84.9	15.6	10.3	74.1
C	Race Horses	15.8	7.9	76.3	9.4	4.7	85.9
	BQ Mall	11.9	7.8	80.3	9.1	13.4	77.5
	Party Scene	9.7	6.7	83.6	7.7	7.0	85.3
	Basketball Drill	7.3	5.6	87.1	3.3	3.1	93.6
D	Blowing Bubbles	11.4	5.8	82.8	7.2	2.9	89.9
	Race Horses	10.7	6.1	83.2	10.5	3.5	86.0
	BQ Square	7.4	5.9	86.7	8.2	5.6	86.2
	Basketball Pass	8.4	4.2	87.4	12.7	1.4	85.9
E	Vidyo1	15.1	8.7	76.2	22.0	2.6	75.4
	Vidyo3	14.2	10.9	74.9	22.7	3.4	73.9
	Vidyo4	16.0	9.4	74.6	20.1	3.4	76.5
	Johnny	15.2	13.7	71.1	21.7	10.3	68.0
Average (%)		**12.5**	**7.8**	**79.7**	**12.1**	**5.9**	**82.0**

Table 3.3 provides an insight into the distribution of the intra prediction modes in the HEVC framework and proposed method. The video sequences chosen for analysis possess rich textural and directional properties, resulting in nearly $(2/3)^{rd}$ of the total coding blocks being predicted using the angular modes, followed by the planar and DC modes. Enhancement in prediction accuracies of the planar and angular modes is observed in the form of a substantial rise in the pixels favoring the GDP prediction strategy. In the case of Main - AI profile configuration, it is evident that the net increase in the number of blocks predicted using the proposed scheme is around 1.90%, thus contributing to an increase in the bit-rate savings.

The effectiveness of the proposed predictor can be justified by another performance indicator which is determined using computation of the average absolute error/sample. The findings of the residual analysis performed on the first intra frame recorded in Table 3.4 reveals that there is a considerable amount of reduction in the sum of absolute error/frame with the proposed scheme. This implicitly states that there is a substantial amount of reduction in the magnitude of error per sample, which helps to achieve savings in bit-rate.

Table 3.4: Prediction error analysis in the HEVC anchor and GDP predictor

Class	Video Sequence	Sum of absolute error/frame		Average absolute error/sample	
		Anchor	GDP	Anchor	GDP
A	Traffic	13508500	13348568	3.30	3.26
B	Kimono	4696252	4528474	2.26	2.18
C	BQ Mall	1887416	1829852	4.73	4.58
D	Race Horses	605768	587526	6.07	5.88
E	Vidyo1	1778940	1725494	1.93	1.87

The number of blocks coded with different PU sizes, ranging from 4×4 to 32×32 implicitly signifies the enhancement in prediction accuracy using the GDP predictor. To substantiate this, the number of blocks of different PU sizes considering the first frame of one representative video sequence from each class is computed and presented in Table 3.5 for the Main - AI profile configuration. This tabulation remains the same for Main - LB, LP, and RA as every first frame in each of the four profile configurations is intra coded. From the analysis, it is evident that using the proposed GDP predictor, more number of pixels ($(L \times L) \times Block\ Count$) favor larger PU sizes which is graphically elaborated in Fig. 3.7 for the BQ Mall (Class C) video sequence. This implies that there is a substantial

reduction in the total number of blocks coded using the proposed algorithm in comparison with the HEVC anchor. Reduction in the total number of blocks minimizes the transmission of the control and header information associated with each block, thus achieving additional compression.

Table 3.5: Comparison of the total number of blocks coded using the various PU block sizes in the HEVC anchor and proposed GDP predictor

Class	Video Sequence	Method	Block Count				
			4x4	8x8	16x16	32x32	Total
A	Traffic	Anchor	233240	5458	54	1	238753
		GDP	76456	20974	3014	741	101185
B	Kimono	Anchor	110124	4565	72	1	114762
		GDP	10600	9118	1858	825	22401
C	BQ Mall	Anchor	20236	1025	35	1	21297
		GDP	12496	1084	324	46	13950
D	Race Horses	Anchor	6040	50	0	0	6090
		GDP	2636	529	57	9	3231
E	Vidyo1	Anchor	51200	1440	36	1	52677
		GDP	20912	3620	960	107	25599

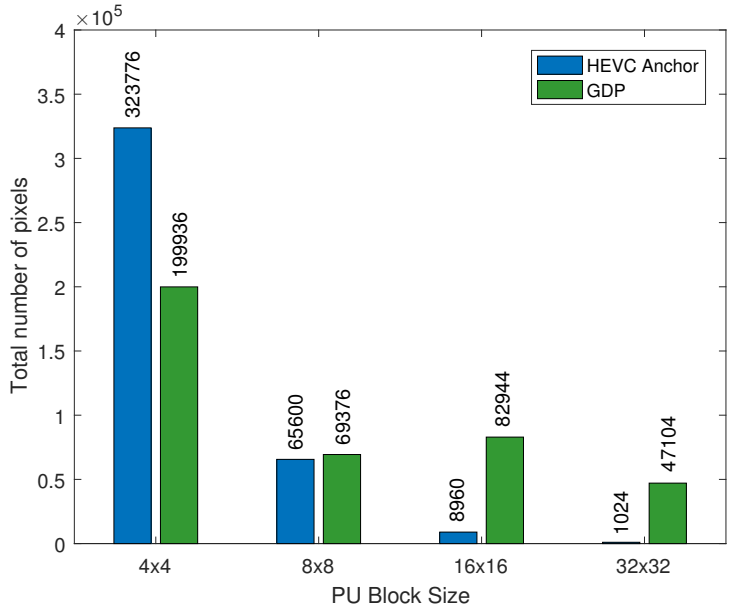


Figure 3.7: Graphical representation of the pixel distribution for the various PU sizes in the BQ Mall sequence

The savings in run-time for the Main profile is tabulated in Table 3.6. It is evident that the run-time savings computed at the encoder and decoder side are subtly less than or at par with the HEVC anchor. However, subtle increase in run-time for certain classes of video sequences is observed which is indicated using the negative sign. On the whole, it can be concluded that the overall existing complexity of the HEVC framework has not increased due to the proposed modifications.

Table 3.6: Run-time savings (%) in the proposed GDP predictor

Class	**AI**		**LB**		**LP**		**RA**	
	ET†	**DT**‡	**ET**	**DT**	**ET**	**DT**	**ET**	**DT**
A	0.23	0.82	0.13	0.48	0.10	0.12	-0.04	0.05
B	0.22	0.16	-0.05	1.08	-0.02	0.26	0.24	0.14
C	1.38	0.45	0.90	0.24	0.06	-0.08	0.06	0.18
D	0.44	1.34	0.92	0.14	0.22	0.06	0.18	0.20
E	0.78	1.26	1.02	0.46	0.40	0.30	-0.06	0.10
Average (%)	**0.61**	**0.81**	**0.58**	**0.48**	**0.15**	**0.13**	**0.08**	**0.13**

$Note-$ † Encoding time, ‡ Decoding time

3.5 Summary

In the proposed work, an efficient gradient dependent prediction strategy is presented that enhances the prediction accuracy of HEVC in the lossless mode. The simulation results highlight that the proposed GDP predictor can minimize the bit-rate by 8.49%, 1.86%, 2.14%, and 2.38% corresponding to the Main - AI, Main - LB, Main - LP, and Main - RA profile configurations in comparison with the HEVC anchor. The reduction in the number of blocks coded using the proposed method implicitly indicates the improvement in prediction accuracy. This suitably helps to minimize the overall bit-rate while maintaining the computational complexity at the encoder and decoder slightly less than or at par with the HEVC anchor. However, one shortfall for the proposed prediction strategy is that it uses the same prediction process for the five different prediction modes. On the whole, the proposed GDP algorithm has been proven beneficial for the natural sequences as prominent enhancement in prediction accuracy and compression efficiency is achieved, without any increase in the run-time.

Chapter 4

CONTEXT-BASED PLANAR, DC AND ANGULAR PREDICTIONS

4.1 Introduction

The chapter presents context-based strategies for enhancing the intra prediction mechanisms of HEVC. In general, the video sequences are commonly characterized by sudden variations in the local statistics. Hence, most of the video compression schemes incorporate adaptive approaches for linear prediction to exploit the presence of the finest details. Several renowned techniques exist for the estimation of the amount and degree of texture within an image, that uses co-occurrence matrices, sum and difference matrices or local gray-scale dependence matrices etc. But then, the entire process to derive the feature set using these techniques is resource-demanding and computationally intensive, resulting in deriving a less complex feature set. The compressibility of an image is related to the range and distribution of the intensities within the region of consideration, average pixel intensity, degree of variability in the local pixel intensity values, pixel entropy, gradient information, etc. These contexts promote the modeling of the prediction error distribution.

In the proposed techniques, the context of the target pixel is derived using the variability of the sum of absolute differences and local pixel intensity values for the planar and angular intra prediction modes respectively, in association with the gradient information. This fusion is done basically to refine the reliability of the results, thereby enhancing the overall compression efficiency of HEVC. In addition to the above, an enhanced prediction mechanism is proposed for uniform regions within a given data block. This mechanism intends to improve the prediction accuracy using the samples from the current and neighboring blocks.

The prediction generation process is based on the previously reconstructed samples in the closest proximity to the target pixel depending on its location along with the surrounding DC content. On the whole, by efficiently utilizing the spatial correlation within the image, the proposed work makes use of the sample-based approach for lossless coding along with the context-based strategies to fine-tune the prediction process. In the forthcoming sections, while the proposed prediction strategies for planar, DC and angular modes are referred to as difference-based planar prediction (DPP), sample-based DC (SDC) and intensity-variant angular prediction (IAP) respectively, their combination i.e. DPP, SDC, and IAP is referred to as combined intra prediction (CIP).

4.2 Contextual Prediction Strategy for Planar Mode

The proposed DPP predictor adapts itself to the local characteristic features based on the contextual background of the causal pixels surrounding the target pixel $C_{x,y}$ within the block. The predictor derives the context around $C_{x,y}$ in the planar mode using the similarity metric sum of absolute differences (SAD) described in (4.1), where M corresponds to the number of pixels within the patch. The metric represented as D_i is computationally less intense due to the simple sum and difference operations, providing a degree of similarity between the two coding blocks.

$$D_i = SAD(P_h^g[m], P_h^g[n]) = \sum_{m,n \in M} \mid h[m] - h[n] \mid \tag{4.1}$$

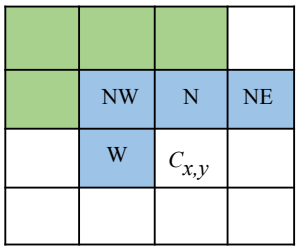


Figure 4.1: An illustration to compute the SAD value D_4

Accordingly, the algorithm derives the contextual information of the four (K:1→4) neighboring directional causal pixels around $C_{x,y}$ in the North (N), West (W), North East (NE) and North West (NW) directional orientations as shown in Fig. 4.1. $D(C_{x,y})$ in (4.2) represents a 1D array that holds the D_i values derived corresponding to the target pixel

$C_{x,y}$ and each of its four neighboring causal pixels C = $\{N, W, NE, NW\}$ located in the vertical, horizontal, diagonal and anti-diagonal directions respectively.

$$D(C_{x,y}) = \{D_1, D_2, D_3, D_4\} \tag{4.2}$$

Figure 4.1 illustrates the computation of the D_i value between the target pixel $C_{x,y}$ and its causal neighbor NW. Here, the patch $P_h^g[m]$ in blue corresponds to $C_{x,y}$ and patch $P_h^g[n]$ in green corresponds to NW respectively. Each patch consists of four (M) pixels. The absolute difference is now computed between every causal pixel $h[m]$ in $P_h^g[m]$ and its co-located pixel $h[n]$ in $P_h^g[n]$ to derive the SAD value D_4. Once $D(C_{x,y})$ is formed using the four SAD values, its mean (μ_d) value and variability in the absolute differences (σ_d^2) over the specified region is computed using (4.3) and (4.4).

$$\mu_d = \frac{1}{K}\sum_{i=1}^{K} D_i \tag{4.3}$$

$$\sigma_d^2 = \frac{1}{K}\sum_{i=1}^{K} (D_i - \mu_d)^2 \tag{4.4}$$

When σ_d^2 is zero, it indicates that the area surrounding the target pixel is homogeneous in nature, thereby generating the prediction value using the conventional planar prediction. Meanwhile, the implication for a non-zero σ_d^2 is that the region is non-homogeneous, thus deriving a 4-bit binary pattern for the same. This is achieved by creating a list $S(C_{x,y})$ which is defined and generated as given in (4.5). It holds four values that is based on the SAD metric D_i, keeping μ_d as a referencing parameter. Each list element can be now interpreted as a binary value to form a 4-bit binary pattern and serve as a basis for context based classification. This is essentially done to group together the regions of a similar pattern.

$$S(C_{x,y}) = [S_1\, S_2\, S_3\, S_4] \tag{4.5}$$

$$S_i = \begin{cases} 0 & D_i > \mu_d \\ 1 & else \end{cases}, \quad where\, i = 1, 2, 3, 4$$

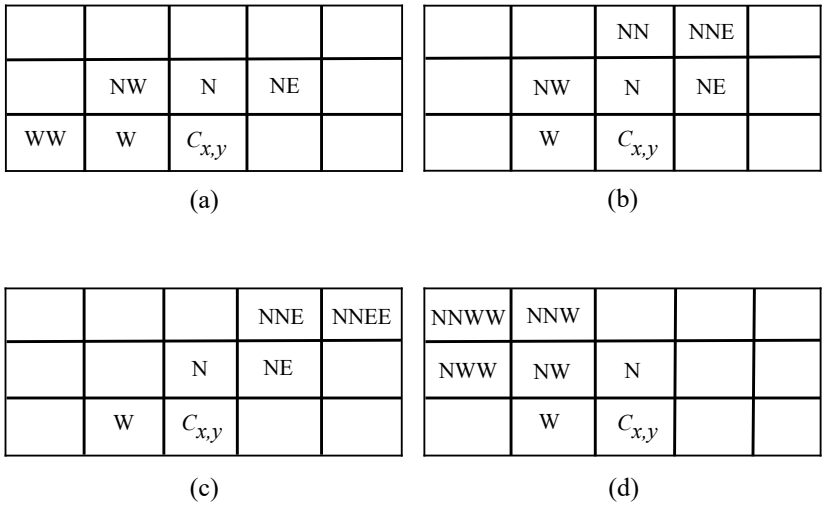


Figure 4.2: Template for gradient computation in the four viewing directions: (a) horizontal (b) vertical (c) diagonal (d) anti-diagonal

Now, the generated 4-bit binary patterns, broadly categorized as $pattern\,1$ and $pattern\,2$. While $pattern\,1$ depicts the presence of a pattern in a particular direction, $pattern\,2$ indicates the presence of a complex pattern. The prediction strategy is further strengthened by inclusion of the concept of gradients as described in (4.6)–(4.9) based on Fig. 4.2. The gradients derived in the four viewing directions are utilized to enhance the prediction strategy. Here, G_H, G_V, G_D and G_{AD} represents the gradient information in the horizontal, vertical, diagonal and anti-diagonal directions respectively.

$$G_H = \mid W - WW \mid + \mid N - NW \mid + \mid N - NE \mid \tag{4.6}$$

$$G_V = \mid W - NW \mid + \mid N - NN \mid + \mid NE - NNE \mid \tag{4.7}$$

$$G_D = \mid W - N \mid + \mid N - NNE \mid + \mid NE - NNEE \mid \tag{4.8}$$

$$G_{AD} = \mid W - NWW \mid + \mid NW - NNWW \mid + \mid N - NNW \mid \tag{4.9}$$

The GAP technique proposed by Wu and Memon (1997) is applied to generate the prediction $\hat{P}[x, y]$ for $pattern\,1$ as provided in Algorithm 3. Based on G_i and G_j values along with the predefined threshold, GAP categorizes the edges as normal or weak. Here, it is important to note that G_i and G_j represent the gradient values in the two viewing directions that are orthogonal to each other, while $C[i]$ and $C[j]$ correspond to the causal neighboring pixels from C in the specified direction.

Algorithm 3 Pseudocode for the prediction strategy in planar mode ($pattern\ 1$)

$\hat{P}[x,y] = ([(C[i] + C[j]) >> 1] + C_{diff}) >> 2$

if $(G_i - G_j > 32)$ **then**

$\hat{P}[x,y] = (\hat{P}[x,y] + C[j]) >> 1$

else if $(G_i - G_j > 8)$ **then**

$\hat{P}[x,y] = (3.\hat{P}[x,y] + C[j]) >> 2$

end if

In an illustration, consider a 4-bit binary pattern '1000' which implies that the gradient is significant in the horizontal direction. The threshold-based prediction process is carried out in comparison with the gradient in the vertical direction i.e. $G_i = G_H$ and $G_j = G_V$. Here, the subscript i specifies the viewing direction that has significant gradient information. The final prediction thus includes the causal pixels $C[i] = W$ and $C[j] = N$ in the horizontal and vertical viewing directions respectively, along with $C_{diff} = (NE - NW)$ that represents the difference between the causal pixels in the remaining two directions i.e. diagonal and anti-diagonal.

The remaining combinations of the binary pattern i.e. $pattern\ 2$ make use of the prediction estimation provided in (4.10). It is mainly based on the causal neighbors and weighting factor (α) that is determined using the gradient values G_{min1} and G_{min2} as provided in (4.11). C_{Gmin1} and C_{Gmin2} represents the relative causal pixel values corresponding to the two minimal gradients G_{min1} and G_{min2} respectively. Here, the implication is that the causal pixel relative to the first minimal directional gradient offers a predominant contribution to the prediction generation. This is due to the presence of the term G_{min1} in the numerator of (4.11) which results in a lower value of the weighing factor α corresponding to C_{Gmin2}.

$$\hat{P}[x,y] = \alpha.C_{Gmin2} + (1-\alpha).C_{Gmin1} \tag{4.10}$$

$$\alpha = \frac{G_{min1}}{G_{min1} + G_{min2}} \tag{4.11}$$

In an illustration for the prediction generation, if $G_{min1} = G_H$ and $G_{min2} = G_D$, then the corresponding causal pixel value will be $C_{Gmin1} = W$ and $C_{Gmin2} = NE$. The entire prediction strategy is well explained through the flowchart in Fig. 4.3.

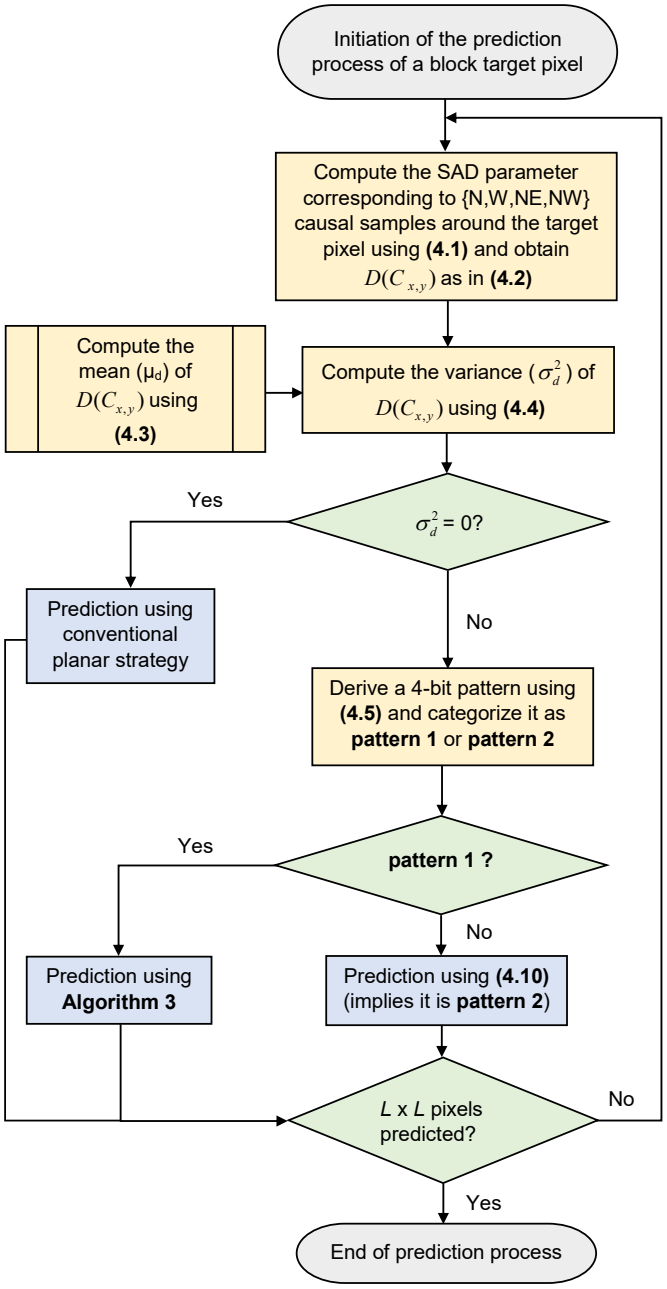


Figure 4.3: Flowchart for the prediction process in the DPP predictor

4.3 Combined Sample-based DC Prediction Strategy

The sample-based DC prediction (SDC) strategy intends to enhance the conventional DC prediction strategy in HEVC. It is achieved by performing the averaging of the block DC value (P_{DC}) and nearby reconstructed samples. For each target pixel $C_{x,y}$, a maximum of

three previously reconstructed samples located in the immediate neighborhood are required for the prediction generation i.e. one on the immediate left, top, and top-left of the target pixel as shown in the Fig. 4.4. All the pixels within the block are subjected to the same prediction mode and the pixel prediction process follows a horizontal scan pattern. To note that, the reference samples R_{above} and R_{left} are represented as $\hat{R}[x, y]$ in general.

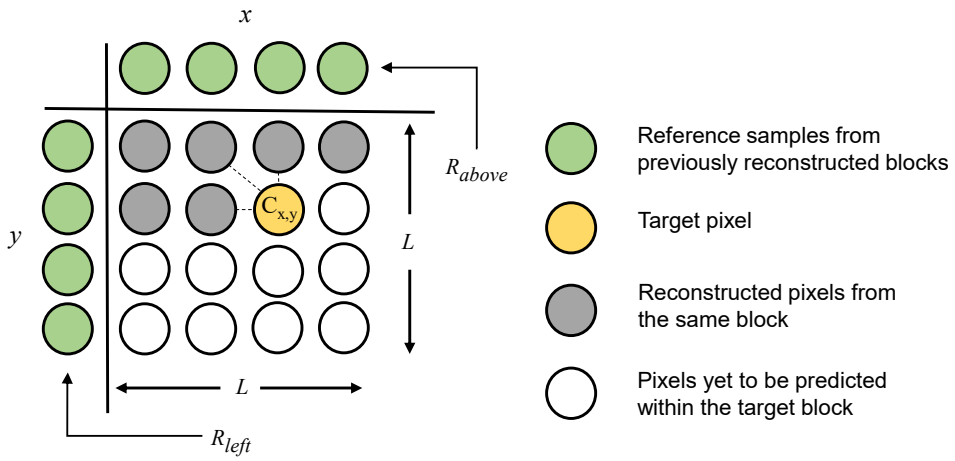


Figure 4.4: SDC prediction strategy

Algorithm 4 Pseudocode for SDC intra prediction

Initially compute P_{DC} value:

$$P_{DC} = \frac{1}{2L}(\sum_{x=0}^{L-1} \hat{R}[x, -1] + \sum_{y=0}^{L-1} \hat{R}[-1, y])$$

if (x = 0 && y = 0) **then**

$$\hat{P}[x, y] = P_{DC}$$

else if (x = 0) **then**

$$\hat{P}[x, y] = (P_{DC} + \hat{R}[0, y - 1]) >> 1$$

else if (y = 0) **then**

$$\hat{P}[x, y] = (P_{DC} + \hat{R}[x - 1, 0]) >> 1$$

else

$$\hat{P}[x, y] = (P_{DC} + \hat{R}[x, y - 1] + \hat{R}[x - 1, y] + \hat{R}[x - 1, y - 1]) >> 2$$

end if

The initial computation of P_{DC} essentially requires the $2L$ reference samples from the neighboring predicted blocks on the top and left to the current $L \times L$ block. The P_{DC} value which is computed at the block level is further fine-tuned for a better prediction accuracy as explained in Algorithm 4 based on the location of $C_{x,y}$. At the decoder side, the exact inverse process is carried out to reconstruct the input video sequence.

4.4 Contextual Prediction Strategy for Angular Mode

In the angular mode, the IAP predictor derives the context around the target pixel $C_{x,y}$ based on the variations in the local pixel intensity values. The mean (μ_{int}) and variance (σ^2_{int}) of the pixel intensities is defined in (4.12) and (4.13). Here, the variable a_i corresponds to the pixel intensity value of the i^{th} causal neighbor around the target pixel $C_{x,y}$. Figure 4.5 instantiates the $\tilde{R}$ causal pixels that are considered for the computation of the pixel intensity variation.

$$\mu_{int} = \frac{1}{\tilde{R}} \sum_{i=1}^{\tilde{R}} a_i \tag{4.12}$$

$$\sigma^2_{int} = \frac{1}{\tilde{R}} \sum_{i=1}^{\tilde{R}} (a_i - \mu_{int})^2 \tag{4.13}$$

NNWW	NNW	NN	NNE
NWW	NW	N	NE
WW	W	$C_{x,y}$	

Figure 4.5: Template for computation of variation in pixel intensity

When σ^2_{int} is zero, the median predictor will generate the prediction value using the four causal neighborhood pixels around the target pixel. In the case of non-zero σ^2_{int}, the prediction strategy is based on the gradient flow in the four viewing directions that are derived using (4.6)–(4.9). The snippet of pseudocode for the modified angular prediction strategy is provided in Algorithm 5.

Algorithm 5 Pseudocode for angular prediction strategy

if $(\sigma_{int}^2 > 0)$ **then**
  if $(G_i \neq 0 \mid_{i=s} \ \& \& \ \sum_{i=1}^{4} G_i = 0 \mid_{i \neq s})$ **then**
    if $(G_i > 80)$ **then**
      $\hat{P}[x,y] = C[j];\ where\, j \perp i$
    else
      $\hat{P}[x,y] = C[j] + C[i] - C[p]$
    end if
  else
    $\hat{P}[x,y] = \lfloor \frac{\sum\limits_{m \in K} wt_m \cdot h[m]}{\sum\limits_{m \in K} wt_m} \rfloor$
  end if
end if

To elaborate, consider a scenario wherein there is a single non-zero gradient value that is along the horizontal direction i.e. $G_i = G_H$. Here, subscript i denotes a directional orientation that is orthogonal to the direction specified by j, ranging between 1 and 4 corresponding to horizontal, vertical, diagonal and anti-diagonal directions respectively. When G_H is found to be greater than the threshold, the generated prediction value is simply $C[j] = N$. Here, the aim is to identify the sharp edges using the threshold value that is set to 80 based on the experimental trials. On the other hand, if the horizontal gradient G_H is less than the specified threshold value, then the generated prediction value $\hat{P}[x,y]$ would be based on the planar interpolation using the causal pixels $C[i] = W$, $C[j] = N$ and $C[p] = NW$, with the directional inclination of p at an orientation of 45^o from both i and j.

In scenarios where more than one gradient exists predominantly, the prediction process is carried out using the sample-based weighted prediction scheme (Wige et al. (2013)). This technique is mainly based on the weighted averaging of the neighboring samples (K:1$\rightarrow$4) around the target pixel to derive the prediction value as provided in (4.14).

$$\hat{P}[x,y] = \lfloor \frac{\sum\limits_{m \in K} wt_m \cdot h[m]}{\sum\limits_{m \in K} wt_m} \rfloor \tag{4.14}$$

Here, $h[m]$ comprises of the prior reconstructed neighborhood pixels $\{N, W, NE, NW\}$ in the immediate proximity to the target pixel $C_{x,y}$. The factor wt_m computed using (4.15)

indicates the weight of the reconstructed pixel $h[m]$. Section 3.3 gives a brief description on the significance of the included parameters.

$$wt_m = \lfloor a \cdot b^{-\frac{SAD(P_h^g[m], P_h^g[n])}{h_{dist}}} \rfloor \tag{4.15}$$

$$a = 2^{31-2-\beta}$$

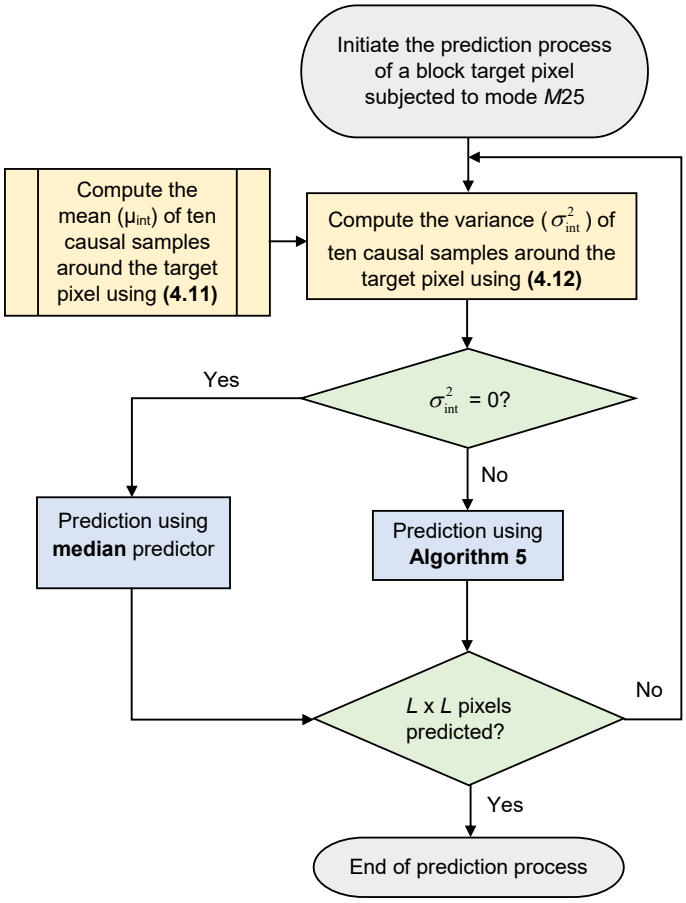


Figure 4.6: Flowchart for the prediction process in the IAP predictor

Now, the lossless mode based on IAP is embedded within the HEVC anchor by replacing one of the least likely used angular mode. This is done to minimize the modifications to the basic framework. From the detailed analysis regarding the usage of the 35 HEVC intra prediction modes provided in Guarda et al. (2017), it is evident that the angular modes 25 and 27 are the least frequently used modes. This serves as a basis for the replacement of mode 25 with the IAP prediction strategy, post validating the same. The entire algorithm is depicted in the flowchart presented in Fig. 4.6. Meanwhile, the remaining 32 directional

angular modes use the conventional SAP strategy (Zhou et al. (2012)) for the prediction as discussed in Section 3.3.1.

4.5 Simulation Results, Analysis and Discussions

Table 4.1 - Table 4.3 tabulates the relative bit-rate savings achieved using the standalone planar, DC and angular predictors, along with their combinations in comparison with the HEVC anchor. Here, it can be observed that incorporation of the DC predictor (SDC) with the combination of planar (DPP) and angular (IAP) predictors benefits only the Class A (UHD) and F (SC) video sequences subtly as they possess more of homogeneous regions. Apart from this fact, the net savings in bit-rate has not majorly affected the Class B-E sequences for any of the profile configurations. Therefore, in further discussions, the analysis is restricted to the combination of planar, DC and angular predictors termed as the CIP. The CIP predictor renders total savings in bit-rate of 9.13%, 2.06%, 2.40%, and 2.59% respectively, for the Main - AI, Main - LB, Main - LP, and Main - RA profile configurations, while the same offers 1.49%, 0.66%, and 0.87% savings in bit-rate for the Main_SCC - AI, Main_SCC - LB, and Main_SCC - RA respectively. The graphical representations in Fig. 4.7 and Fig. 4.8 clearly state that the combination of predictors for the planar, DC and angular modes outperform the standalone predictors.

Table 4.1: Bit-rate savings (%) for the Main - AI and Main - LB profile configurations using Class A-E sequences

Class	**AI**					**LB**				
	DPP[1]	**SDC**[2]	**IAP**[3]	**DPP+IAP**[4]	**CIP**[5]	**DPP**	**SDC**	**IAP**	**DPP+IAP**	**CIP**
A	9.15	2.11	11.75	12.20	12.26	2.35	0.53	3.13	3.40	3.44
B	4.96	1.56	7.34	7.52	7.51	0.98	0.34	2.06	2.14	2.12
C	3.82	1.04	5.59	5.77	5.77	1.21	0.24	1.71	1.80	1.79
D	5.57	1.21	8.63	8.92	8.94	1.00	0.18	1.48	1.57	1.57
E	6.62	1.38	10.84	11.17	11.15	0.44	0.08	1.32	1.36	1.36
Average (%)	**6.02**	**1.46**	**8.83**	**9.12**	**9.13**	**1.20**	**0.27**	**1.94**	**2.05**	**2.06**

$Note-$

[1] Difference-based planar prediction

[2] Sample-based DC prediction

[3] Intensity-variant angular prediction

[4] Combination of DPP and IAP

[5] Combined intra prediction (DPP+SDC+IAP)

Table 4.2: Bit-rate savings (%) for the Main - LP and Main - RA profile configurations using Class A-E sequences

Class	LP					RA				
	DPP	**SDC**	**IAP**	**DPP+IAP**	**CIP**	**DPP**	**SDC**	**IAP**	**DPP+IAP**	**CIP**
A	2.67	0.63	3.66	3.97	4.01	2.72	0.62	3.60	3.87	3.90
B	1.25	0.50	2.56	2.63	2.62	1.25	0.42	2.43	2.50	2.49
C	1.28	0.26	1.85	1.93	1.88	1.40	0.29	1.98	2.08	2.07
D	1.05	0.19	1.57	1.66	1.67	1.36	0.25	2.06	2.16	2.17
E	0.66	0.14	1.79	1.84	1.80	0.98	0.23	2.28	2.35	2.31
Average (%)	**1.38**	**0.34**	**2.29**	**2.41**	**2.40**	**1.54**	**0.36**	**2.47**	**2.59**	**2.59**

Table 4.3: Bit-rate savings (%) for the Main_SCC profile using Class F sequences

Class	AI					LB					RA				
	DPP	**SDC**	**IAP**	**DPP+IAP**	**CIP**	**DPP**	**SDC**	**IAP**	**DPP+IAP**	**CIP**	**DPP**	**SDC**	**IAP**	**DPP+IAP**	**CIP**
F	1.14	1.14	1.34	1.45	1.49	0.26	0.03	0.57	0.66	0.66	0.39	0.23	0.72	0.85	0.87

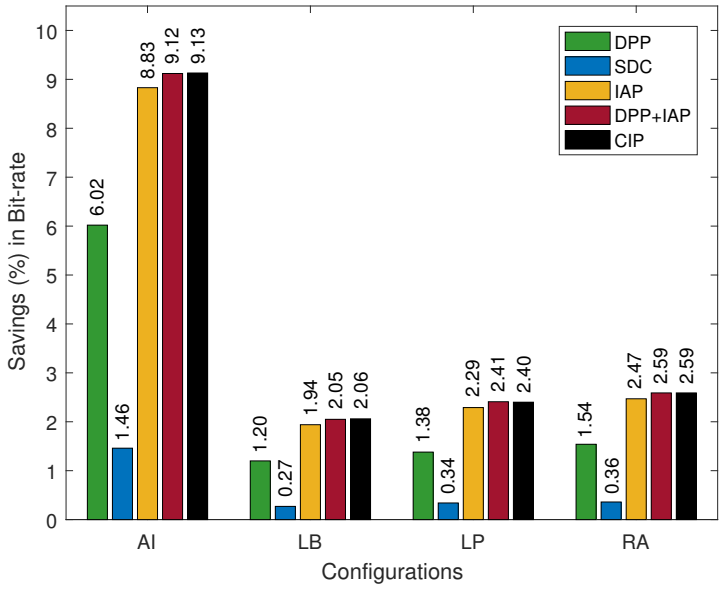


Figure 4.7: Graphical representation of the bit-rate savings in the standalone predictors and their combinations for the Main profile using Class A-E sequences

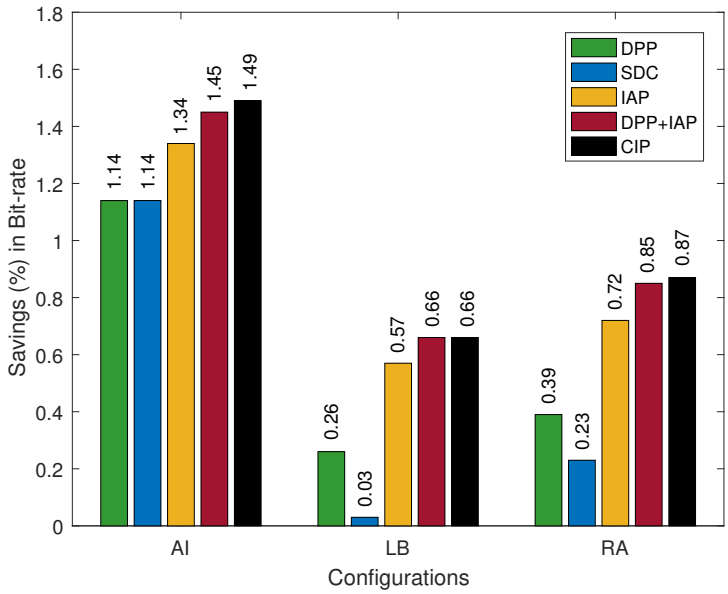


Figure 4.8: Graphical representation of the bit-rate savings in the standalone predictors and their combinations for the Main_SCC profile using Class F sequences

Table 4.4 tabulates the percentage of blocks predicted using the planar, DC and angular prediction modes in the HEVC anchor and proposed CIP predictor for the first intra frame. On the whole, Classes A, B, C, and E have responded well to the planar predictor, while Classes D and F responded moderately to the predictor. The DC predictor benefitted the Classes A and F sequences, while Class D responded moderately to this predictor. The results reveal that all test data classes are at an advantage due to the modified mode $M25$. In general, it can be claimed that each type of predictor has helped the blocks to be rightly predicted, resulting in enhancement of the prediction accuracy.

Table 4.5 depicts the number of blocks coded using the various PU sizes ranging between 4×4 and 32×32, considering the first frame of the video sequences. On comparing with the HEVC anchor, it is observed that more number of blocks are coded using the PU size 32×32 in the CIP based prediction. In other words, there is a substantial drop in the number of blocks coded using the minimum PU size 4×4, resulting in reduction of the total coded blocks within a given frame. This helps to reduce the side information, thereby achieving additional compression and reduction in run-time. Figure 4.9 illustrates the same graphically in terms of pixel distribution for the various PU sizes in the Kimono (Class B) video sequence. Due to the inclusion of the new set of coding tools within the SCM, Class F sequences are excluded from the analysis in Table 4.5.

Table 4.4: Intra prediction mode distribution (%) in the HEVC anchor and CIP predictor

Class	Video Sequence	Anchor				CIP			
		Planar (M0)	DC (M1)	Angular (M2-M34)		Planar (M0)	DC (M1)	Angular (M2-M34)	
				M25	Others			M25	Others
A	Traffic	12.8	6.1	1.7	79.4	24.8	8.1	8.3	58.8
	People on Street	12.1	7.2	2.1	78.6	32.7	8.2	7.7	51.4
B	Kimono	21.1	8.7	1.4	68.8	20.5	7.1	9.6	62.8
	Park Scene	15.7	11.1	1.8	71.4	27.1	9.3	6.6	57.0
	BQ Terrace	8.6	6.5	15.3	69.6	12.4	5.3	6.2	76.1
C	Race Horses	15.8	7.9	2.0	74.3	25.4	7.6	7.0	60.0
	BQ Mall	11.9	7.8	5.9	74.4	16.6	6.9	9.0	67.5
	Party Scene	9.7	6.7	2.9	80.7	14.1	5.2	10.5	70.2
	Basketball Drill	7.3	5.6	0.8	86.3	10.7	5.0	2.2	82.1
D	Blowing Bubbles	11.4	5.8	4.4	78.4	9.3	4.9	4.5	81.3
	Race Horses	10.7	6.1	2.2	81.0	18.5	8.0	8.7	64.8
	BQ Square	7.4	5.9	2.8	83.9	7.6	6.2	5.7	80.5
	Basketball Pass	8.4	4.2	2.0	85.4	10.6	1.8	12.4	75.2
E	Vidyo1	15.1	8.7	5.3	70.9	19.4	3.4	5.3	71.9
	Vidyo3	14.2	10.9	4.1	70.8	23.4	5.3	7.3	64.0
	Vidyo4	16.0	9.4	3.6	71.0	28.7	4.1	4.1	63.1
	Johnny	15.2	13.7	4.1	67.0	16.9	4.2	4.2	74.7
F	Basketball Drill Text	7.4	4.7	0.5	87.4	6.9	7.6	3.2	82.3
	China Speed	8.7	2.4	0.6	88.3	8.8	4.0	6.8	80.4
	Slide Show	8.5	3.0	0.8	87.7	17.8	5.0	9.4	67.8

Table 4.5: Comparison of the total number of blocks coded using the various PU block sizes in the HEVC anchor and CIP predictor

Class	Video Sequence	Method	Block Count				
			4x4	8x8	16x16	32x32	Total
A	Traffic	Anchor	233240	5458	54	1	238753
		CIP	114116	22079	2304	261	138760
B	Kimono	Anchor	110124	4565	72	1	114762
		CIP	19992	12694	1873	451	35010
C	BQ Mall	Anchor	20236	1025	35	1	21297
		CIP	15716	1391	170	15	17292
D	Race Horses	Anchor	6040	50	0	0	6090
		CIP	3236	487	54	3	3780
E	Vidyo1	Anchor	51200	1440	36	1	52677
		CIP	22836	2887	827	156	26706

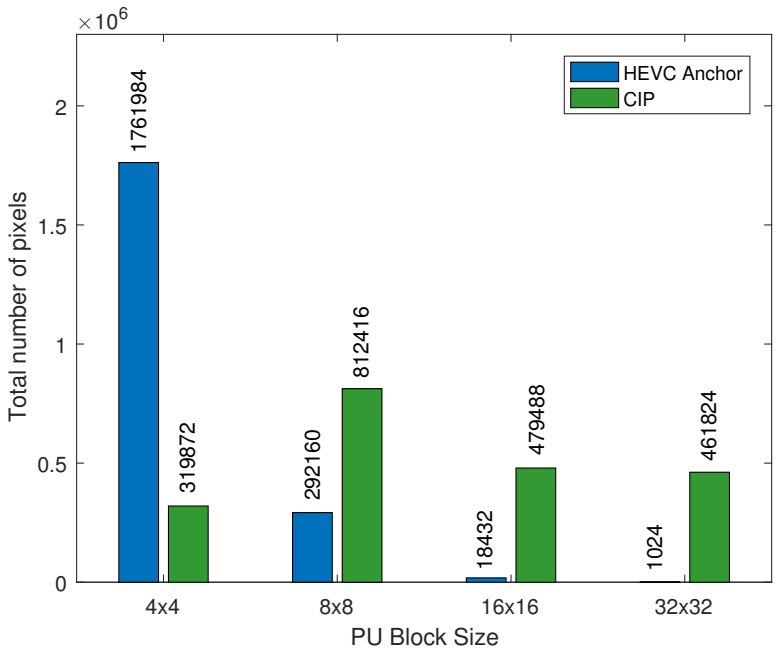


Figure 4.9: Graphical representation of the pixel distribution for various PU block sizes in the Kimono sequence

From Table 4.6, it is quite evident that the sum of residual error/frame is significantly lower in the case of the sample-based CIP predictor in comparison with the conventional block-based HEVC anchor. This implicitly states that there is a reduction in the residual energy. Now, the resultant error minimization is an indication of the fact that the magnitude of the generated residual values are lesser than in the HEVC anchor, thus reducing the time consumed for the entropy coding.

Table 4.6: Prediction error analysis in the HEVC anchor and CIP predictor

Class	**Video Sequence**	**Sum of absolute error/frame**		**Average absolute error/sample**	
		Anchor	**CIP**	**Anchor**	**CIP**
A	Traffic	13508500	11847276	3.30	2.89
B	Kimono	4696252	4513268	2.26	2.18
C	BQ Mall	1887416	1818086	4.73	4.55
D	Race Horses	605768	598435	6.07	5.99
E	Vidyo1	1778940	1749592	1.93	1.90
F	China Speed	3690580	2068284	4.69	2.63

The run-time savings (%) in the CIP predictor over the HEVC anchor has been recorded in Table 4.7. Improvement in the prediction accuracy has led to lesser encoding and decoding time. In short, the savings accomplished in entropy coding time compensates for any increase in the computation time, if exists, due to the computational complexities.

Table 4.7: Run-time savings (%) in the proposed CIP predictor

Class	AI		LB		LP		RA	
	ET†	DT‡	ET	DT	ET	DT	ET	DT
A	0.11	0.43	0.15	0.28	0.12	0.10	-0.08	0.14
B	0.25	0.20	0.08	0.30	0.14	0.08	0.07	0.15
C	0.84	0.20	0.52	0.12	0.22	0.18	0.22	0.02
D	0.44	0.32	0.44	0.16	0.18	0.12	0.06	0.18
E	0.48	0.24	0.38	0.28	-0.04	0.05	0.15	0.10
F	0.15	0.12	-0.02	0.05	–	–	-0.04	0.07

$Note-$ † Encoding time, ‡ Decoding time

4.6 Summary

The proposed CIP predictor enhances the prediction accuracy of the HEVC intra modes, thereby attaining better compression efficiency by suitably modifying the intra prediction strategies of HEVC. The proposed predictor utilizes the local image characteristics to clearly distinguish between the gradually varying surfaces, uniform regions, and regions with edges. The prediction strategy aims to rightly code the blocks using the suitable prediction mode at the pixel level. The comparative simulation results state that the proposed sample-based CIP predictor outperforms the HEVC anchor considerably in the lossless coding mode. The predictor validated using the Class A-E sequences has established a significant improvement in the bit-rate savings by 9.13%, 2.06%, 2.40%, and 2.59% over the HEVC anchor for the Main - AI, Main - LB, Main - LP, and Main - RA profile configurations respectively. It also renders prominent savings in bit-rate of 1.49%, 0.66%, and 0.87% over the HEVC anchor using Class F sequences for the Main_SCC - AI, Main_SCC - LB, and Main_SCC - RA profile configurations respectively. As the proposed work reduces the total number of blocks quite significantly, the coding overhead also decreases thereby bringing improvements in the run-time. The results and analysis thus reveal that the CIP strategy serves as a reliable predictor in the candidate list formed to process the intra mode selection for lossless compression.

Chapter 5

PIXELWISE IMPROVISED BLEND OF PREDICTORS FOR PLANAR AND ANGULAR MODES

5.1 Introduction

The chapter presents a pixelwise fusion of sub-predictors for the improvisation of the planar and angular modes. In general, most of the available video content is heterogeneous in nature, comprising of several edges and textural structures. The DPCM based prediction in its simplest form cannot simultaneously deal with the pattern that includes smooth regions, regions with discontinuities, etc. within an image block. To handle the vividly diverse nature of the video sequences, a heuristic blend of sub-predictors based on the penalizing factors is used to derive the optimum prediction mode for a target block. The mechanism of blending the sub-predictors has proven to benefit the process of identifying the predominant nature within the current block to choose the best-suited predictor that majorly contributes to the final prediction value.

5.2 Blend of Predictors for Planar and Angular Modes

This section elaborates on the incorporation of a blending strategy using a set of sub-predictors to enhance the intra prediction process of HEVC. The scheme referred to as the improvised blend of predictors (IBP) is mainly based on the past of the nearby causal

samples and embedded into the planar and angular modes of the HEVC framework. It is designed to locally adapt to the regions with smooth gradients and edges. As a pre-requisite prior to the process of blending, penalizing factors corresponding to each of the pre-selected sub-predictors are derived to serve as the weighted coefficients during the prediction generation. A lower value of the penalizing factor signifies that the corresponding sub-predictor contributes more to the prediction process. Therefore, this factor intuitively serves as a local measure to suitably penalize the sub-predictor that results in larger residuals.

A total of six simple sub-predictors that share a dependency on the pixels in immediate proximity are considered for the prediction generation. Each of the sub-predictor depends solely on one or a combination of the four causal neighborhood pixels. The selection of these sub-predictors relies on the correlation between the target pixel and its neighborhood samples. They are meant to generate the prediction value based on their local adaptation. The categorization of these sub-predictors mainly intends to adhere to the regions that possess discontinuities and smooth gradients. They are grouped as Set 1: [N, W, NE, NW] and Set 2: [$(N+W-NW)$, $(W+NE-N)$] predictors to identify the patterns with edges and smooth gradients respectively. Hence, Set 1 and Set 2 group of predictors comprising of four (S = 4) and two (S = 2) sub-predictors are chosen to efficiently predict the angular and planar modes respectively. Here, N, W, NE and NW represent the causal neighbors to the target pixel $C_{x,y}$ at the specified directional orientation. Once the sub-predictors are assigned based on the prediction mode, then the rest of the computations that involve the derivation of the penalizing factors ($G_{pf,i}$) and correction term ($\hat{E}[x,y]$) towards the generation of the final prediction value are common to both the modes. Set 1 and Set 2 predictors generate four and two penalizing factors corresponding to the angular and planar modes respectively.

For the ease of equation formulation, let $a(n)$ series denote the generalized pixel notation for the target pixel $C_{x,y}$ and its causal neighborhood pixels as shown in Fig. 5.1. To be precise, here $a(n)$ is the target pixel which is to be predicted and $a(n-m)$ $\{m$:1 to $K\}$ represents the surrounding four causal pixels shaded in gray. Furthermore, $p_i(n-m)$ $\{i$:1 to $S\}$ represents the prediction estimate corresponding to the i^{th} sub-predictor for the causal neighbor $a(n-m)$. Figure 5.2 elaborates the pixels (shaded in green) that are involved in the derivation of $p_i(n-m)$ corresponding to the six sub-predictors. The penalizing factor $G_{pf,i}$ corresponding to each of the chosen sub-predictors are derived as given in (5.1). $G_{pf,i}$ for the i^{th} sub-predictor is determined at the pixel level based on the cumulative

residuals of the causal neighbors around $C_{x,y}$.

$$G_{pf,i} = \sum_{m=1}^{K} |a(n-m) - p_i(n-m)| \tag{5.1}$$

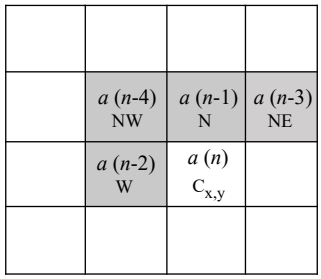


Figure 5.1: Template of the four causal neighbors $a(n-m)$ with the directional notation

In an illustration for angular mode, consider deriving the penalizing factor $G_{pf,1}$ for the sub-predictor N from the group Set 1. The computation of $G_{pf,1}$ includes four absolute differential terms $(N-NN)$, $(W-NW)$, $(NE-NNE)$ and $(NW-NNW)$ corresponding to the four causal neighbors as depicted in Fig. 5.2(a).

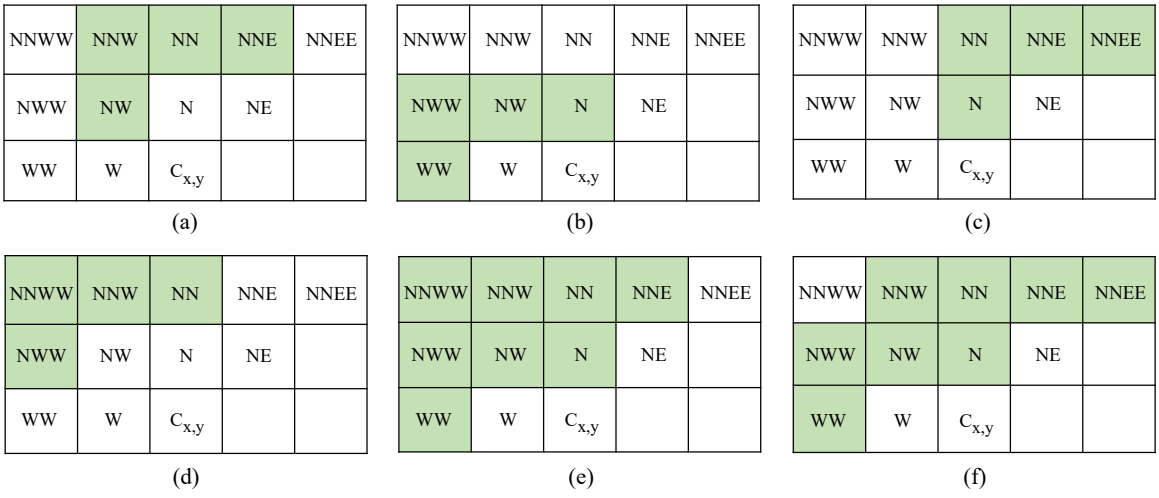


Figure 5.2: Template of the causal neighbors essential in deriving $p_i(n-m)$ corresponding to the sub-predictors: (a) N (b) W (c) NE (d) NW (e) $N+W-NW$ (f) $W+NE-N$

Once the S penalizing factors are available, the prediction value $\tilde{P}[x,y]$ is computed as given in (5.2). The term P_f in the denominator of (5.2) is the normalization factor that

makes the sum of the weighted coefficients as unity.

$$\tilde{P}[x,y] = \frac{1}{P_f} \sum_{i=1}^{S} \frac{p_i(n)}{G_{pf,i}} \tag{5.2}$$

$$P_f = \sum_{i=1}^{S} \frac{1}{G_{pf,i}}$$

The prediction value $\tilde{P}[x,y]$ which is derived using (5.2) is made up of S predictor estimates for the target pixel $C_{x,y}$, with the inverse of the relevant penalizing factors $(1/G_{pf,i})$ serving as the weighted coefficients. This implies that if the sub-predictor predicts well, then the corresponding weighted coefficient will be substantially higher than the rest. The blending process entirely depends on the value of $G_{pf,i}$. While all the $G_{pf,i}$ values are non-zero, the contribution of all the sub-predictors is vital for the prediction generation. In situations when exactly one of the $G_{pf,i}$ value is zero, the implication is that the predictor estimate $p_i(n)$ corresponding to the sub-predictor is used to predict the neighborhood pixels as well. Hence, $p_i(n)$ is chosen as the prediction value in such scenarios. Meanwhile, when more than one penalty factor is zero, then arbitrarily any of the corresponding sub-predictors can be chosen as the prediction value since it does not affect the entropy.

The generated prediction value $\tilde{P}[x,y]$ is further fine-tuned to capture the influence of the blending predictor on the causal neighborhood around the target pixel $C_{x,y}$. This is achieved by determining the correction term $\hat{E}[x,y]$ as given in (5.3). The term represents the average cumulative residuals of the neighborhood causal pixels around $C_{x,y}$. It is then added to $\tilde{P}[x,y]$ for obtaining the final prediction value $\hat{P}[x,y]$ as provided in (5.4). This helps to detect and adapt to the dominant characteristics in the local neighborhood region which may include edges or smooth variations in the intensity values.

$$\hat{E}[x,y] = \frac{1}{K} \sum_{m=1}^{K} (a(n-m) - p(n-m)) \tag{5.3}$$

$$\hat{P}[x,y] = \tilde{P}[x,y] + \hat{E}[x,y] \tag{5.4}$$

At instances when the previously reconstructed samples are unavailable for the prediction i.e. pixels along the block boundary, then the closest available reconstructed pixel is extended to serve as a reference sample for the subsequent encoding process.

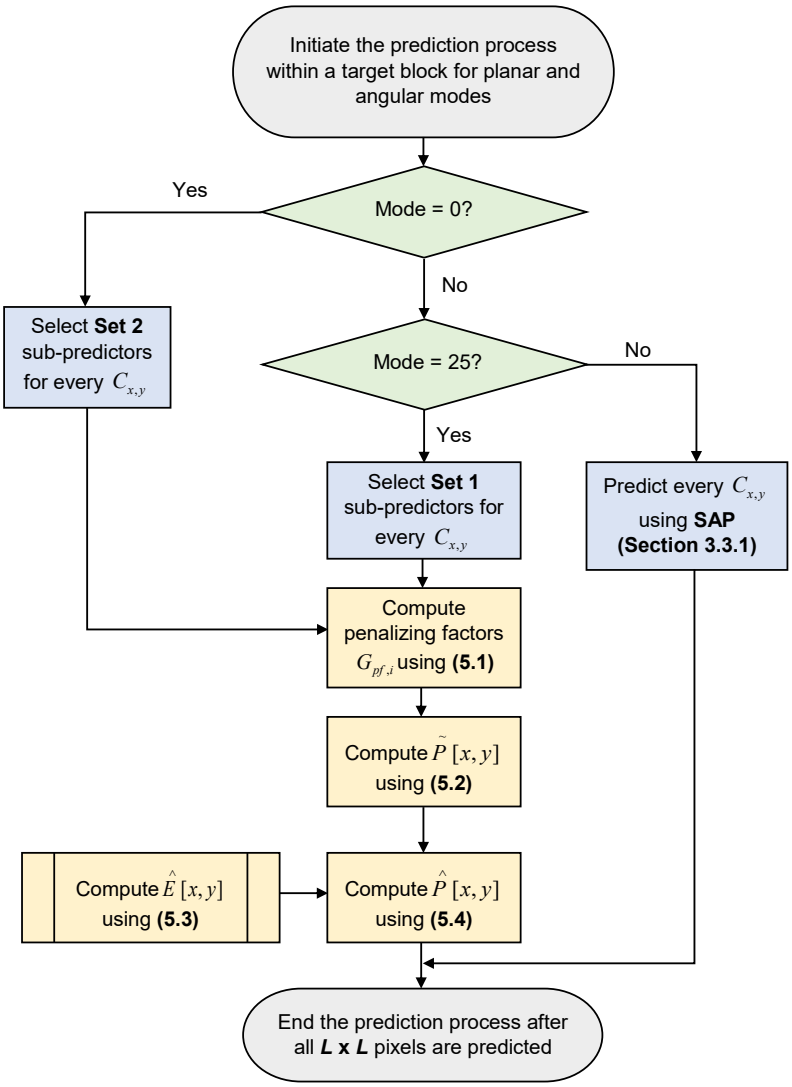


Figure 5.3: Flowchart for IBP prediction scheme

Based on the statistical analysis and several sets of trials performed on the test video sequences, mode $M25$ has emerged as the least frequently used prediction mode out of the 33 angular modes. Hence, mode $M25$ is reserved for the IBP based prediction process, thereby instilling minimal modifications to the HEVC framework as discussed in Section 4.4. This prediction mode is at an angular displacement of -2 from the vertical mode ($M26$) aligned along the vertical axis. Meanwhile, the remaining 32 angular modes are predicted using the SAP prediction scheme described in Section 3.3.1. Figure 5.3 depicts the entire IBP prediction mechanism embedded into the HEVC framework to improve the prediction accuracy of the planar and angular intra prediction modes.

In general, the concept of the blending of sub-predictors restricts the amount of contribution by the non-dominant characteristic of the region around the pixel under consideration. This is made possible by suitably selecting a sub-predictor with a minimal penalizing factor that implicitly helps to generate an optimum prediction.

5.3 Simulation Results, Analysis and Discussions

The subset of predictors i.e. Set 1 and Set 2 are individually applied to $M25$ of angular and planar modes respectively. The standalone predictors are termed as IBP for angular (IBP-A) and IBP for planar (IBP-P) in the aforementioned discussion. The combined behavior of the Set 1 and Set 2 sub-predictors for the angular and planar modes of HEVC is referred to as the IBP predictor. Table 5.1 and Table 5.2 gives an insight into the savings in bit-rate achieved using the IBP-P, IBP-A, and IBP predictors for the Main and Main_SCC profiles respectively.

Table 5.1: Bit-rate savings (%) in the IBP-P, IBP-A and IBP predictors for the Main profile

Class	**AI**			**LB**			**LP**			**RA**		
	IBP-P	**IBP-A**	**IBP**	**IBP-P**	**IBP-A**	**IBP**	**IBP-P**	**IBP-A**	**IBP**	**IBP-P**	**IBP-A**	**IBP**
A	10.53	13.35	13.75	2.82	3.85	4.10	3.14	4.42	4.68	3.24	4.35	4.60
B	4.12	7.69	7.79	0.74	2.14	2.16	0.81	2.64	2.64	0.98	2.53	2.54
C	2.66	5.36	5.46	0.86	1.72	1.79	0.90	1.85	1.91	1.01	1.98	2.05
D	4.45	8.73	8.97	0.79	1.52	1.62	0.83	1.61	1.70	1.11	2.11	2.21
E	5.30	11.08	11.55	0.13	1.34	1.35	0.19	1.80	1.83	0.41	2.30	2.34
Average (%)	**5.41**	**9.24**	**9.50**	**1.07**	**2.11**	**2.20**	**1.17**	**2.46**	**2.55**	**1.35**	**2.65**	**2.75**

Table 5.2: Bit-rate savings (%) in IBP-P, IBP-A and IBP predictors for the Main_SCC profile

Class	**AI**			**LB**			**RA**		
	IBP-P	**IBP-A**	**IBP**	**IBP-P**	**IBP-A**	**IBP**	**IBP-P**	**IBP-A**	**IBP**
F	0.93	1.04	1.29	0.10	0.61	0.65	0.56	0.91	0.99

The tables reveal that the IBP prediction technique performs exceptionally well when the standalone predictors perform jointly on the regions with smooth gradients and edges. The graphical representation in Fig. 5.4 interprets the same and thus the further analysis is conducted using the IBP predictor alone. In case of the Main profile, it can be observed that

an overall savings of 9.50%, 2.20%, 2.55%, and 2.75% are achieved in terms of bit-rate for the AI, LB, LP, and RA configurations respectively, validated using the sequences belonging to Class A-E. Meanwhile, bit-rate savings of 1.29%, 0.65%, and 0.99% are obtained using the proposed method for the Main_SCC - AI, Main_SCC - LB, and Main_SCC - RA profile configurations respectively, validated using the Class F sequences.

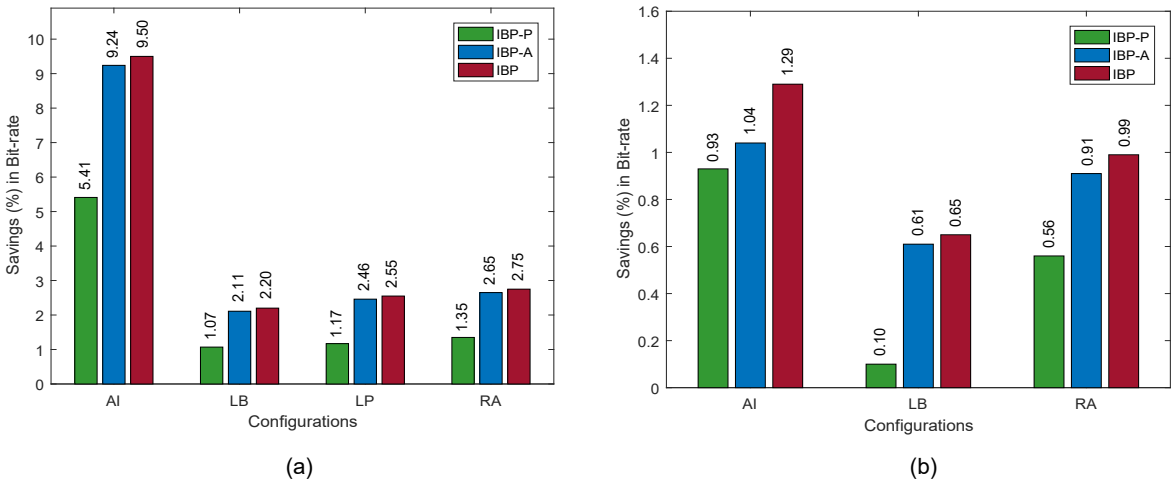


Figure 5.4: Graphical representation of the bit-rate savings in the standalone predictors and their combinations for the (a) Main profile using Class A-E sequences (b) Main_SCC profile using Class F sequences

Detailed classwise summary on the savings (%) in bit-rate is tabulated in Table 5.3 - Table 5.7 which has been derived in comparison with the HEVC anchor for the various state-of-the-art prediction strategies and the proposed methods i.e. GDP, CIP, and IBP. To strengthen the validation of the proposed methods, Class 4K which is a general test dataset consisting of natural UHD video sequences is also considered and the corresponding results are tabulated in Table 5.8. From these tables, it is clear that the proposed IBP predictor outperforms the proposed methods of GDP and CIP in addition to the state-of-the-art schemes, using Class A-E sequences. On the other hand, this predictor performs moderately on the Class F and 4K sequences in comparison with the CIP predictor. The possible reason for such occurrences could be the presence of more homogeneous regions or edginess within the data blocks which is the characteristic trait of this class of sequences, thus favoring DC or angular prediction. Table 5.9 supports this claim by providing the information on the mode distribution. The graphical representations in Fig. 5.5 - Fig. 5.7 provide better behavioral visualization of the proposed predictors and existing state-of-the-art prediction

techniques in comparison to the HEVC anchor.

Table 5.3: Bit-rate savings (%) for the Main - AI profile configuration

Class	RDPCM[1]	SAP[2]	MSAP[3]	SWP[4]	DTM[5]	CIG[6]	GDP[7]	CIP[8]	IBP[9]
A	9.55	10.69	10.75	8.39	6.74	11.10	11.27	12.26	13.75
B	5.35	6.80	6.85	4.85	3.25	7.12	7.17	7.51	7.79
C	4.29	4.79	4.90	4.29	2.88	5.12	5.37	5.77	5.46
D	7.52	7.76	7.80	5.71	5.19	8.16	8.19	8.94	8.97
E	9.24	10.03	10.10	5.18	5.05	10.40	10.45	11.15	11.55
Average (%)	**7.19**	**8.01**	**8.08**	**5.68**	**4.62**	**8.38**	**8.49**	**9.13**	**9.50**

$Note-$

[1] Residual DPCM (Lee et al. (2013))

[2] Sample-based angular prediction (Zhou et al. (2012))

[3] Modified sample-based angular prediction (Antony and Sreelekha (2015))

[4] Sample-based weighted prediction (Wige et al. (2013))

[5] Directional template matching (Wige et al. (2013))

[6] Combination of ISAP and GASP (Antony and Sreelekha (2017))

[7] Gradient-oriented directional prediction (TGAPP+GSSWAP)

[8] Combined intra prediction (DPP+SDC+IAP)

[9] Improvised blend of predictors

Table 5.4: Bit-rate savings (%) for the Main - LB profile configuration

Class	RDPCM	SAP	MSAP	SWP	DTM	CIG	GDP	CIP	IBP
A	2.38	2.72	2.80	1.84	1.37	2.82	3.01	3.44	4.10
B	1.40	1.90	1.94	1.02	0.75	1.98	2.01	2.12	2.16
C	1.18	1.54	1.58	1.03	0.67	1.60	1.64	1.79	1.79
D	1.19	1.30	1.32	0.92	0.69	1.35	1.39	1.57	1.62
E	0.96	1.20	1.22	0.26	0.28	1.25	1.25	1.36	1.35
Average (%)	**1.42**	**1.73**	**1.77**	**1.01**	**0.75**	**1.80**	**1.86**	**2.06**	**2.20**

Table 5.5: Bit-rate savings (%) for the Main - LP profile configuration

Class	RDPCM	SAP	MSAP	SWP	DTM	CIG	GDP	CIP	IBP
A	2.82	3.20	3.25	2.10	1.61	3.32	3.53	4.01	4.68
B	1.67	2.36	2.36	1.32	0.84	2.38	2.50	2.62	2.64
C	1.27	1.69	1.70	1.12	0.71	1.72	1.79	1.88	1.91
D	1.27	1.42	1.42	0.97	0.72	1.45	1.48	1.67	1.70
E	1.35	1.23	1.25	0.43	0.41	1.26	1.38	1.80	1.83
Average (%)	**1.68**	**1.98**	**2.00**	**1.19**	**0.86**	**2.03**	**2.14**	**2.40**	**2.55**

Table 5.6: Bit-rate savings (%) for the Main - RA profile configuration

Class	RDPCM	SAP	MSAP	SWP	DTM	CIG	GDP	CIP	IBP
A	2.77	3.15	3.18	2.20	1.68	3.26	3.47	3.90	4.60
B	1.68	2.24	2.28	1.27	0.95	2.30	2.37	2.49	2.54
C	1.40	1.78	1.79	1.27	0.83	1.85	1.92	2.07	2.05
D	1.71	1.82	1.83	1.31	1.06	1.86	1.94	2.17	2.21
E	1.71	2.08	2.12	0.63	0.60	2.16	2.20	2.31	2.34
Average (%)	**1.85**	**2.21**	**2.24**	**1.34**	**1.02**	**2.29**	**2.38**	**2.59**	**2.75**

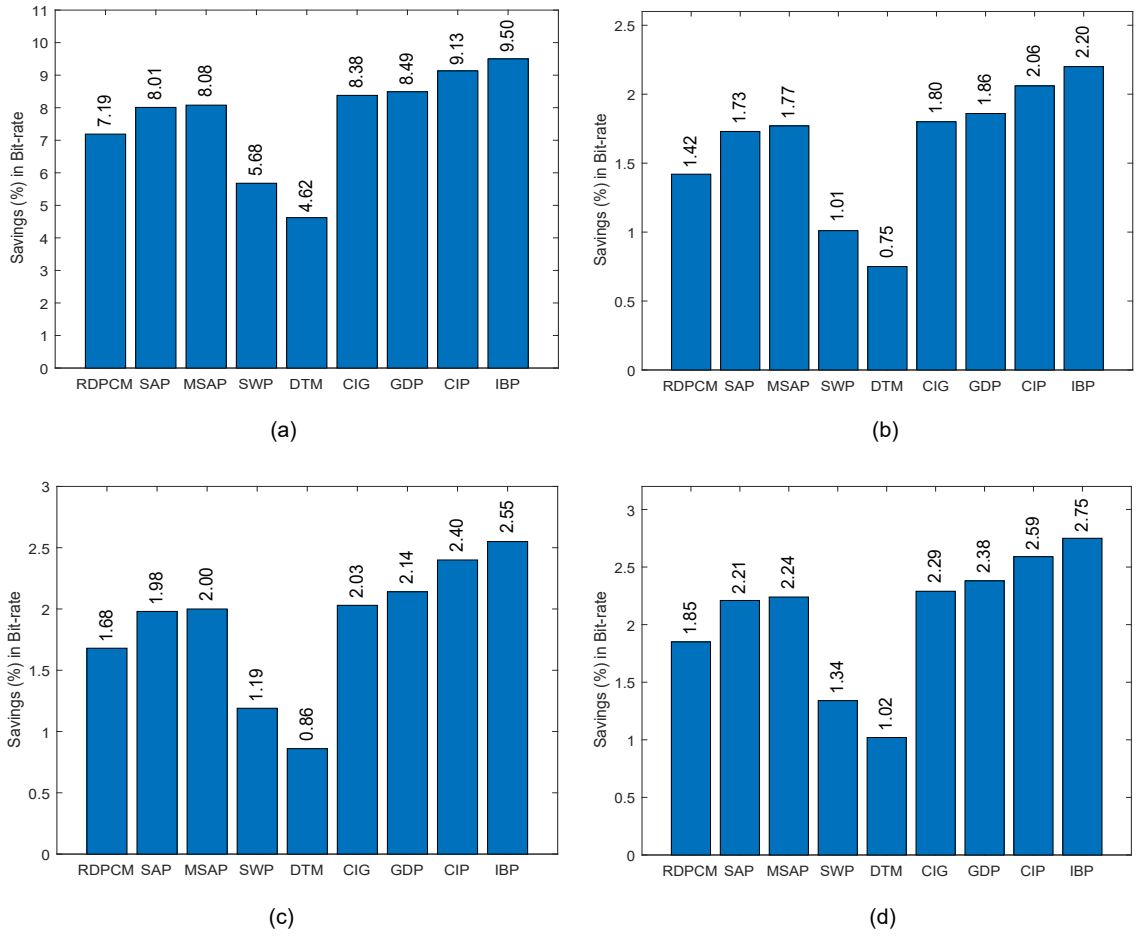


Figure 5.5: Graphical representation of bit-rate savings in the various prediction techniques using Class A-E sequences for the profile configurations: (a) Main - AI (b) Main - LB (c) Main - LP (d) Main - RA

Table 5.7: Bit-rate savings (%) for the Main_SCC profile

Configurations	RDPCM	SAP	MSAP	SWP	DTM	CIG	CIP	IBP
AI	0.65	0.59	0.64	1.06	1.42	0.96	1.49	1.29
LB	0.40	0.37	0.40	0.17	0.28	0.45	0.66	0.65
RA	0.48	0.40	0.42	0.33	0.41	0.46	0.87	0.99

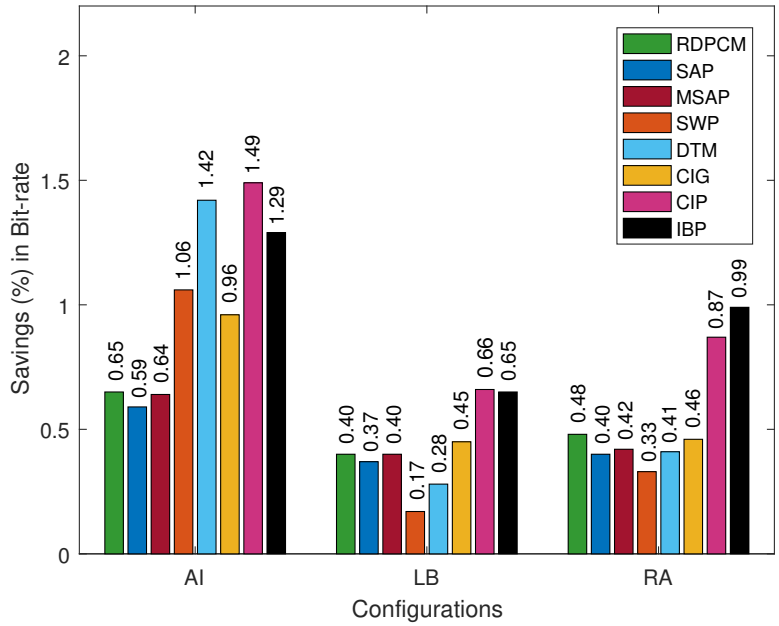


Figure 5.6: Graphical representation of the bit-rate savings in the various prediction techniques for the Main_SCC profile using Class F sequences

Table 5.8: Bit-rate savings (%) for the Main profile using Class 4K sequences

Configurations	RDPCM	SAP	MSAP	SWP	DTM	CIG	GDP	CIP	IBP
AI	2.62	4.26	4.32	4.22	3.10	4.38	4.42	4.68	4.54
LB	1.25	2.39	2.45	2.47	1.42	2.50	2.46	2.67	2.52
LP	1.46	2.81	2.78	2.81	1.75	2.82	2.85	3.20	2.87
RA	1.32	2.50	2.49	2.53	1.62	2.53	2.52	2.78	2.55

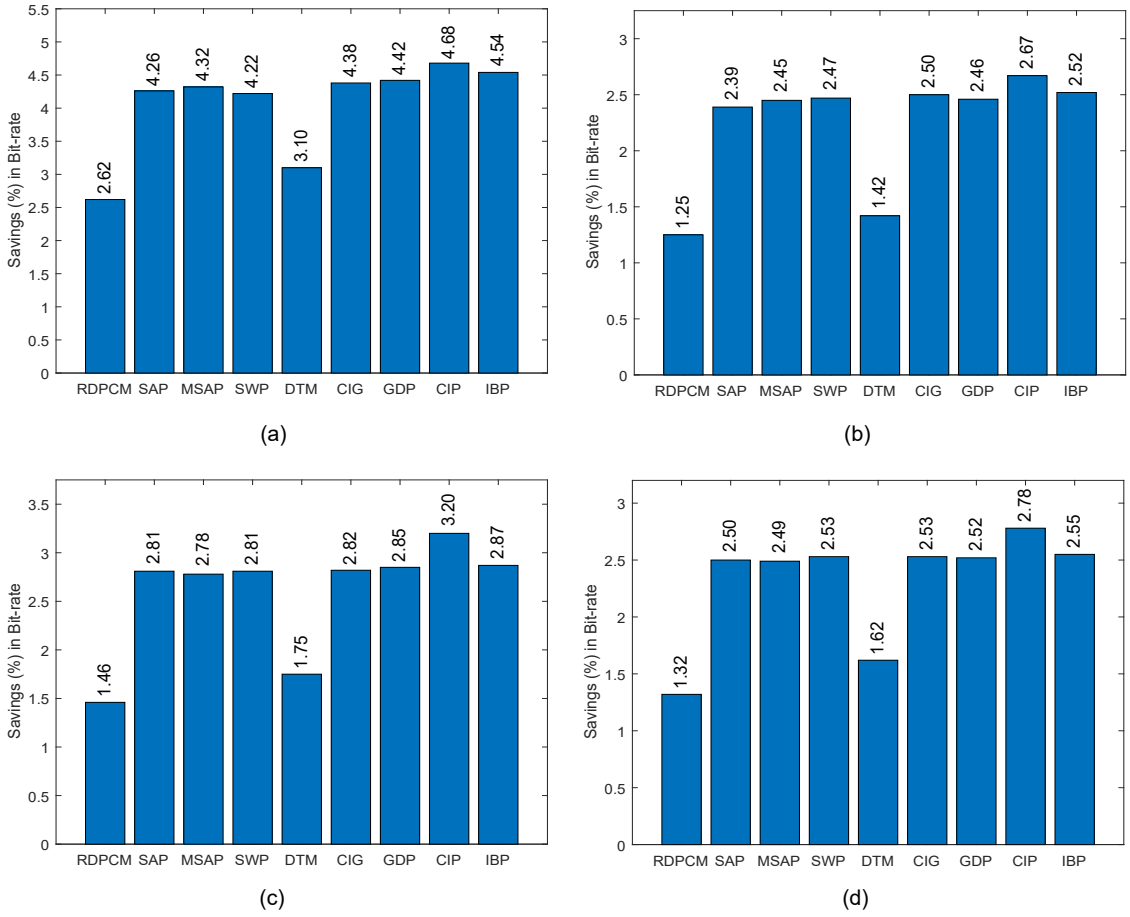


Figure 5.7: Graphical representation of bit-rate savings in the various prediction techniques using Class 4K sequences for the profile configurations: (a) Main - AI (b) Main - LB (c) Main - LP (d) Main - RA

Table 5.9 depicts that embedding the proposed algorithm in the $M0$ and $M25$ modes of HEVC has favored the most number of pixels to be predicted using the IBP prediction strategy. This indicates that there is a prominent increase in the total number of blocks being predicted using the modified $M0$ and $M25$ modes, thus resulting in increased prediction accuracy. This has been possible due to the selection of the predictor type, that targets capturing the dominant characteristic in the causal neighborhood around the target pixel. The table also reveals that a small percentage of blocks are coded using the prediction mode $M25$ in the HEVC anchor. The coding gain that is achieved using the proposed IBP prediction compensates for the subtle losses incurred if any, on the usage of the angular mode $M25$ as a specific prediction mode.

Table 5.9: Intra prediction mode distribution (%) in the HEVC anchor and IBP predictor

Class	Video Sequence	Anchor				IBP			
		Planar (M0)	DC (M1)	Angular (M2-M34)		Planar (M0)	DC (M1)	Angular (M2-M34)	
				M25	Others			M25	Others
A	Traffic	12.8	6.1	1.7	79.4	19.2	1.5	18.8	60.5
	People on Street	12.1	7.2	2.1	78.6	26.8	2.4	28.8	42.0
B	Kimono	21.1	8.7	1.4	68.8	12.1	9.2	11.6	67.1
	Park Scene	15.7	11.1	1.8	71.4	16.5	11.1	18.0	54.4
	BQ Terrace	8.6	6.5	15.3	69.6	7.2	8.8	5.0	79.0
C	Race Horses	15.8	7.9	2.0	74.3	17.9	3.9	12.5	65.7
	BQ Mall	11.9	7.8	5.9	74.4	5.1	12.3	3.7	78.9
	Party Scene	9.7	6.7	2.9	80.7	5.8	7.0	4.1	83.1
	Basketball Drill	7.3	5.6	0.8	86.3	1.8	2.3	1.6	94.3
D	Blowing Bubbles	11.4	5.8	4.4	78.4	3.8	2.8	3.4	90.0
	Race Horses	10.7	6.1	2.2	81.0	12.0	2.5	18.4	67.1
	BQ Square	7.4	5.9	2.8	83.9	2.5	6.8	1.9	88.8
	Basketball Pass	8.4	4.2	2.0	85.4	14.6	1.3	4.2	79.9
E	Vidyo1	15.1	8.7	5.3	70.9	16.6	1.3	7.6	74.5
	Vidyo3	14.2	10.9	4.1	70.8	17.2	2.3	8.7	71.8
	Vidyo4	16.0	9.4	3.6	71.0	21.2	1.9	10.9	66.0
	Johnny	15.2	13.7	4.1	67.0	12.8	5.5	3.7	78.0
F	Basketball Drill Text	7.4	4.7	0.5	87.4	4.3	3.4	1.5	90.8
	China Speed	8.7	2.4	0.6	88.3	5.1	3.1	1.8	90.0
	Slide Show	8.5	3.0	0.8	87.7	19.5	2.5	7.1	70.9
4K	Beauty	12.8	9.0	3.4	74.8	0.7	19.1	1.4	78.8
	Bosphorus	17.1	8.1	2.3	72.5	3.8	14.0	3.5	78.7
	Honey Bee	20.3	8.3	3.1	68.3	1.0	12.4	1.8	84.8
	Jockey	20.2	9.7	3.1	67.0	1.1	15.5	1.3	82.1
	Ready Set Go	12.7	6.5	7.3	73.5	5.0	12.4	9.0	73.6
	Shake & Dry	19.6	6.3	2.1	72.0	2.0	4.9	6.0	87.1
	Yacht Ride	17.7	8.8	2.3	71.2	4.1	17.0	1.7	77.2

Table 5.10 elaborates the distribution of the coded blocks for the various PU sizes. From the table, it is evident that in case of HEVC anchor, more number of blocks are coded using PU size of 4×4 followed by 8×8, 16×16 and 32×32. The incorporation of the IBP prediction technique has resulted in a decrease in the number of blocks coded using 4×4 and a subsequent rise in the coded blocks using 32×32. Figure 5.8 illustrates the distribution of the pixels for the various PU sizes in the Vidyo1 (Class E) video sequence. The illustration clearly states that more number of pixels are coded using larger PU sizes. Additionally,

there is a reduction in the total number of coded blocks as compared to the HEVC anchor, which helps to achieve savings in the run-time.

Table 5.10: Comparison of the total number of blocks coded using the various PU block sizes in the HEVC anchor and proposed IBP predictor

Class	Video Sequence	Method	Block Count				
			4x4	8x8	16x16	32x32	Total
A	Traffic	Anchor	233240	5458	54	1	238753
		IBP	93732	19823	3066	530	117151
B	Kimono	Anchor	110124	4565	72	1	114762
		IBP	11956	3903	957	1355	18171
C	BQ Mall	Anchor	20236	1025	35	1	21297
		IBP	11268	1039	192	101	12600
D	Race Horses	Anchor	6040	50	0	0	6090
		IBP	2772	451	52	13	3288
E	Vidyo1	Anchor	51200	1440	36	1	52677
		IBP	20296	2870	942	168	24276
4K	Yacht Ride	Anchor	415552	23756	457	8	439773
		IBP	176168	55702	5464	500	237834

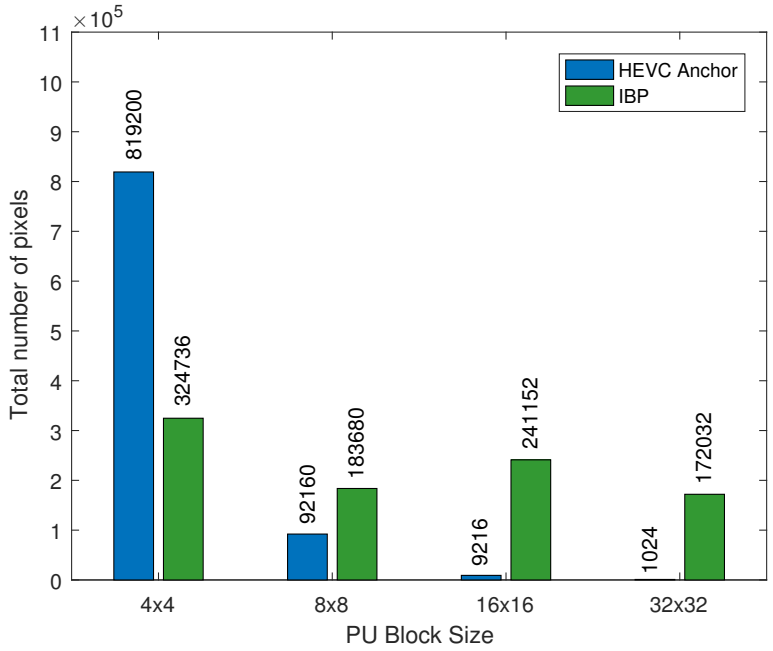


Figure 5.8: Graphical representation of the pixel distribution for the various PU sizes in the Vidyo1 sequence

Table 5.11 highlights the fact that the efficiency of the prediction strategy has resulted in the minimization of the total number of residuals, thereby boosting the coding gain. It indicates that there will be a substantial drop in the prediction error that is needed to represent a pixel on an average. This is because the smaller magnitude of the residuals requires fewer bits during entropy coding.

Table 5.11: Prediction error analysis in the HEVC anchor and IBP predictor

Class	**Video Sequence**	**Sum of absolute error/frame**		**Average absolute error/sample**	
		Anchor	**IBP**	**Anchor**	**IBP**
A	Traffic	13508500	11817152	3.30	2.89
B	Kimono	4696252	4475056	2.26	2.16
C	BQ Mall	1887416	1800572	4.73	4.51
D	Race Horses	605768	594960	6.07	5.96
E	Vidyo1	1778940	1761016	1.93	1.91
F	China Speed	3690580	2125264	4.69	2.70
4K	Yacht Ride	16588744	16132384	2.00	1.94

The details on the run-time savings (%) is presented in Table 5.12. Less complex computations and algorithmic simplicity have resulted in an at par latency at the encoder and decoder, in comparison with the HEVC anchor. In short, an increase in compression efficiency is achieved without any significant rise in the run-time.

Table 5.12: Run-time savings (%) in the proposed IBP predictor

Class	**AI**		**LB**		**LP**		**RA**	
	ET†	**DT**‡	**ET**	**DT**	**ET**	**DT**	**ET**	**DT**
A	1.12	- 0.08	0.45	0.12	0.54	0.06	0.20	0.25
B	0.25	0.32	0.28	- 0.06	0.38	0.12	0.16	- 0.10
C	1.15	0.16	0.66	0.24	1.02	0.15	0.24	0.18
D	1.24	0.14	0.56	0.20	0.82	0.10	1.08	0.15
E	0.54	0.10	0.45	0.24	0.60	0.14	0.15	0.32
F	0.82	- 0.12	0.85	0.26	–	–	0.18	0.30
4K	0.96	0.22	0.72	0.18	0.56	0.09	0.14	0.12

$Note-$ † Encoding time, ‡ Decoding time

5.4 Simulation Results - Whole Slide Images in Pathology

To assess the performance of the proposed methods i.e. GDP, CIP, and IBP for a medical application, they are applied on the whole slide images (WSI) of the tissue samples (WSI images) that are high-resolution images with the finest details as presented in Appendix I. The WSIs of the biopsy tissues usually portray a great variety of cellular structures and tissues, which result in a combination of smooth regions and regions depicting edges in various directions. The smooth regions in the WSIs usually resemble many smooth textures commonly found in natural imagery. However, differently from the natural imagery, WSIs features a very limited number of directional patterns as well.

Table 5.13: Savings (%) in bit-rate and run-time for the Main - AI profile configuration using the WSIs

Pathology Images	**Bit-rate Savings (%)**			**Run-time Savings (%)**					
	GDP	**CIP**	**IBP**	**GDP**		**CIP**		**IBP**	
				ET	**DT**	**ET**	**DT**	**ET**	**DT**
WSI 1	13.46	14.62	16.22	0.07	0.02	0.10	0.02	0.11	0.04
WSI 2	12.57	13.69	15.82	0.05	0.03	0.13	0.05	0.12	0.05
WSI 3	11.46	12.53	14.36	0.05	-0.01	0.12	0.01	0.08	0.02
Average (%)	**12.50**	**13.61**	**15.46**	**0.06**	**0.01**	**0.12**	**0.03**	**0.10**	**0.04**

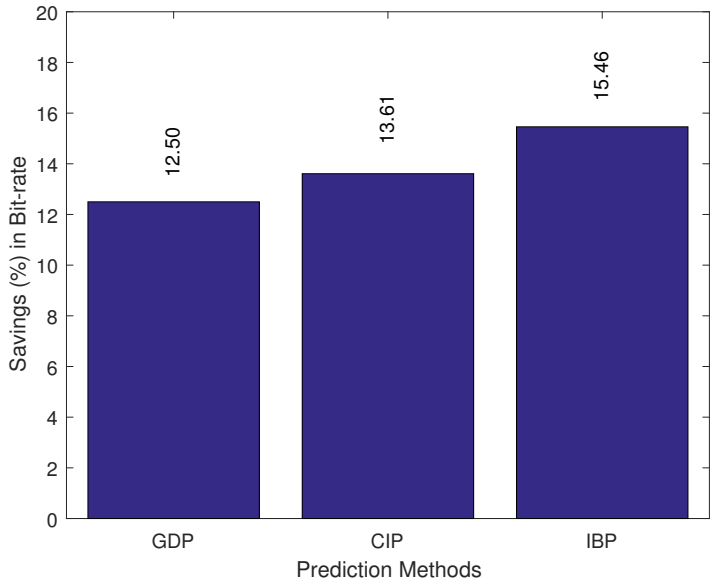


Figure 5.9: Graphical representation of bit-rate savings in the proposed prediction techniques using WSIs for the Main - AI profile configuration

The evaluation is performed using HM-16.12 in 4:2:0 lossless mode for the Main - AI profile configuration. Table 5.13 presents the percentage savings attained in terms of bit-rate and run-time. The table reveals that a promising performance is observed, with IBP predictor outperforming the GDP and CIP predictors as shown in Fig. 5.9. Also, the percentage run-time savings achieved using the three predictors is at par with the HEVC anchor.

5.5 Summary

The proposed IBP predictor, a pixelwise prediction scheme is an efficient mechanism to code the HEVC test video sequences in a lossless manner. This is achieved by suitably modifying the planar and angular modes of the HEVC anchor to embed the simple combination of sub-predictors that adapts to the causal neighborhood with smooth gradients and edges based on the derived penalty factors. Optimal prediction is achieved using this heuristic blend of predictors over the conventional block-based planar and angular prediction strategies. The incorporated mechanism helps to select the best prediction mode for a target block, based on the suitable regional adaptation. Prominent gains are achieved in terms of percentage savings in bit-rate, without any increase in run-time for the lossless coding of the natural and SC sequences. The predictor validation performed on the Class A-E sequences has established a significant improvement in terms of bit-rate savings by 9.50%, 2.20%, 2.55%, and 2.75% over HEVC anchor for the Main - AI, Main - LB, Main - LP, and Main - RA profile configurations respectively. Bit-rate savings of 4.54%, 2.52%, 2.87%, and 2.55% have been accomplished for the aforementioned profile configurations using the Class 4K video sequences as well. It also renders prominent savings in bit-rate of 1.29%, 0.65%, and 0.99% over HEVC anchor using the Class F sequences for the Main_SCC - AI, Main_SCC - LB, and Main_SCC - RA profile configurations respectively. The predictor also outperforms the state-of-the-art prediction techniques present in the literature. Additionally, when tested using WSI pathology images, desirable results are obtained and IBP predictor outperforms the GDP and CIP predictor. In short, the proposed IBP predictor designed to tackle smooth and edgy regions is a promising candidate for lossless compression of the natural and SC video sequences, in addition to the whole slide pathology images.

Chapter 6

CONCLUSION AND FUTURE WORKS

6.1 Concluding Summary

With the increase in the resolution of the video sequences, larger data blocks with uniform regions and smooth variations in addition to regions with drastic intensity variations become quite prominent. The block-based prediction mechanism adopted in HEVC mainly relies on the reference samples belonging to the previously reconstructed neighboring data blocks i.e. blocks on the top, left and top-left. Now, for efficient compression, the chosen prediction technique must be capable to fine-tune the entire prediction process based on the local region characteristics. The sample-based prediction methods such as SWP and DTM predictors accomplish the same but do not cater to the extraction of the finest edge details. Additionally, in cases where a diagonal edge exists, it could result in the propagation of the prediction errors while traversing through the various stages of the encoding process. Such scenarios can be well addressed through the exploitation of the picture characteristics for efficient lossless compression of the video content through suitable modifications in the prediction mechanisms of the HEVC intra prediction modes.

As an outcome of the current research work, prediction schemes such as GDP, CIP, and IBP are proposed that meet these requirements. The prediction process in the GDP algorithm is based on the gradient information around the target pixel in the four directional orientations. Meanwhile, in DPP, IAP, and IBP prediction strategies, it has been possible through the efficient exploitation of the local image characteristics within the data blocks, resulting in appropriate detection of the image traits to smoothly carrying out the prediction

process based on the neighborhood context or the derived penalty on the sub-predictors. On the other hand, the proposed SDC algorithm is a prediction scheme that fine-tunes the DC prediction mechanism specified for the homogeneous regions based on the reference samples of the neighboring blocks in addition to the samples from the target block as well. This work involves lossless mode and hence distortion analysis is not necessary. However, certain other parameters like savings in bit-rate, savings in run-time, distribution of the prediction modes, coded block distribution and residual analysis are used to evaluate the algorithmic performance.

The simulation result analysis reveals that the enhancement in compression efficiency is achieved without any increase in the run-time. It can be also observed that as the prediction accuracy of the proposed sample-based prediction methods is quite superior to the block-based HEVC anchor, there are more zero residuals (i.e. residuals whose magnitude is zero) which are directly fed to the entropy coding stage, that follows the prediction stage. In general, entropy coding accounts for a major portion of the encoding time in comparison with the time taken for the prediction process. But, in the case of the proposed methods, since there is more number of zero residuals, lesser entropy coding time will be required to code them and a reduction in the quantum of the load imposed on the CABAC has been observed. This will implicitly add to the improvement in the run-time savings using the proposed methods. However, from the proposed prediction strategies at an algorithmic level, it is evident that there are several computations involved like gradient computation, the formation of the binary patterns etc., leading to an increase in the encoding time. The savings achieved in entropy coding time will now help to compensate for the undesirable increase in the computational time, thus resulting in an at par latency at encoder and decoder as well. Furthermore, the distribution of the intra prediction modes helps to analyze that more number of pixels have favored the proposed strategies. Additionally, a huge downfall in the total number of coded blocks is observed as a result of more number of pixels being coded using larger PU sizes. This results in the minimized transmission of the control and header information associated with each coding block. From the prediction error analysis, it can be concluded that there is a net reduction in the magnitude of the generated residuals based on the derived sum of absolute error/frame, which is lesser in comparison to the HEVC anchor.

The gradient-oriented GDP prediction strategy for the planar and angular modes can cut down the bit-rates by 8.49%, 1.86%, 2.14%, and 2.38% for the Main - AI, Main - LB, Main - LP, and Main - RA profile configurations respectively, in comparison with the HEVC

anchor using Class A-E sequences. As the Class F sequences are computer-generated/screen content coding sequences, they do not perform as desirable to the GDP predictor, owing to the peculiarities that they possess. The proposed context-based CIP predictor for the planar, DC and angular modes validated using the Class A-E sequences has established a significant improvement in the bit-rate savings by 9.13%, 2.06%, 2.40%, and 2.59% over HEVC anchor for the Main - AI, Main - LB, Main - LP, and Main - RA profile configurations respectively. It also renders prominent savings in bit-rate of 1.49%, 0.66%, and 0.87% using the Class F sequences for the Main_SCC - AI, Main_SCC - LB, and Main_SCC - RA profile configurations respectively. The approach of IBP prediction for the planar and angular modes which is based on the penalizing factors has contributed a significant improvement in the bit-rate savings by 9.50%, 2.20%, 2.55%, and 2.75% over HEVC anchor for the Main - AI, Main - LB, Main - LP, and Main - RA profile configurations respectively, using the Class A-E sequences. It has also accomplished prominent savings in bit-rate of 1.29%, 0.65%, and 0.99% using the Class F sequences for Main_SCC - AI, Main_SCC - LB, and Main_SCC - RA profile configurations respectively. The simulation results also reveal that the proposed prediction algorithms have provided commendable results when validated using the Class 4K video sequences. Considering the Main - AI profile configuration, bit-rate savings of approximately 4.42%, 4.68%, and 4.54% are attained using the proposed GDP, CIP, and IBP predictors respectively. The savings in bit-rate is also observed with the remaining configurations as well. The proposed predictors when applied on the high-resolution whole slide images of the tissues for the Main - AI profile configuration render promising results in terms of savings in bit-rate and run-time.

The simulation results reveal that the proposed sample-based prediction schemes have outperformed the conventional in-built block-based intra prediction mechanisms of HEVC, in addition to state-of-the-art techniques which include RDPCM, SAP, MSAP, SWP, DTM, and CIG. The IBP algorithm performs well when validated using the Class A-E sequences, followed by CIP and GDP schemes. Meanwhile, the CIP predictor outperforms the IBP and GDP prediction mechanisms in the case of Class F video sequences. On the whole, IBP and CIP predictors emerge as effective prediction strategies in comparison with the GDP based prediction. These algorithms have also been tested using the complex natural sequences referred to as Class 4K, which have more uniform regions within a frame, thus possessing more chances of subtle variations within a large coding block. The results have proven that the proposed methods have performed exceptionally well on these sequences as well in the order of CIP, IBP, and GDP. Additionally, these predictors perform exceptionally well when

applied on the whole slide pathology images which possess the intensive finest detail. In short, it can be concluded that the sample-based intra prediction techniques proposed in this thesis intend to aid applications that demand lossless compression by enhancing the prediction accuracy to attain higher coding gains, with no increase in run-time.

6.2 Future Scope

The thesis mainly deals with sample-based prediction strategies for enhancing the prediction accuracy and overall compression efficiency, while in lossless mode. As a future scope in line with the research objectives associated with the work contained in this thesis,

- The prediction strategy can be further improved to facilitate its usage in the remaining angular prediction modes as well.
- The proposed methods in the thesis can be further extended and enhanced to emerge as the best-suited intra prediction mode in the candidate list of modes for lossy compression.

Appendix I
TEST DATASET

A-1 HEVC Test Video Sequences

Any scene when sampled at a point in time produces a video frame and if sampled at regular intervals provides the user an illusion of motion in the captured video. A typical natural video scene comprises several objects each possessing varying shape, depth, texture, and illumination. On the other hand, the SC video sequences possess repeating textural patterns and abrupt edges. In general, the features of a video scene that are relevant for video compression include variation in texture within the scene, number and shape of objects, hue, object motion, movement of the camera, changes in the viewpoint etc. The chosen test video sequences are categorized broadly into two categories: natural (Class A-E and 4K) or computer-generated/SC (Class F). The first video frame of the test data sequences is presented in Fig. A.1 - Fig. A.7.

(a) (b)

Figure A.1: First still of Class A video sequences: (a) Traffic (b) People on Street

(a) (b) (c)

Figure A.2: First still of Class B video sequences: (a) Kimono (b) Park Scene (c) BQ Terrace

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

Figure A.3: First still of Class C video sequences: (a) Race Horses (b) BQ Mall (c) Party Scene (d) Basketball Drill

(a) (b) (c)

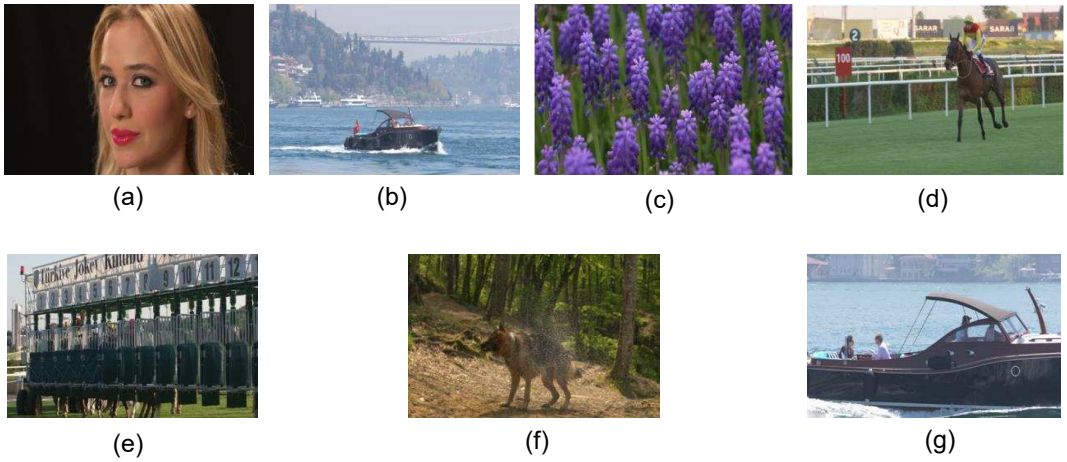
Figure A.4: First still of Class D video sequences: (a) Blowing Bubbles (b) BQ Square (c) Basketball Pass

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

Figure A.5: First still of Class E video sequences: (a) Vidyo1 (b) Vidyo3 (c) Vidyo4 (d) Johnny

(a) (b) (c)

Figure A.6: First still of Class F video sequences: (a) China Speed (b) Basketball Drill Text (c) Slide Show



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

(e) (f) (g)

Figure A.7: First still of Class 4K video sequences: (a) Beauty (b) Bosphorus (c) Honey Bee (d) Jockey (e) Ready Steady Go (f) Shake & Dry (g) Yacht Ride

A-2 Whole Slide Images in Pathology

The whole slide imaging refers to the scanning of the conventional glass slides with Aperio medical equipment such as a microscope or scanscope to produce digital slides. The acquired WSI images are RGB in the scanscope virtual slide (SVS) format. They are converted to YUV format to make it compatible with the HM software. Though SVS format is similar to the TIFF format in terms of features, it has lesser compatibility with the regular image viewers and also tends to have a very large file size. Each SVS file contains multiple images in a predefined order that includes a baseline tiled image, a thumbnail, intermediate pyramid images (made up of tiles), an optional slide label image etc. When it is viewed using a suitable image viewer, a sample area from the image appears as shown in Fig. A.8 - Fig. A.10.

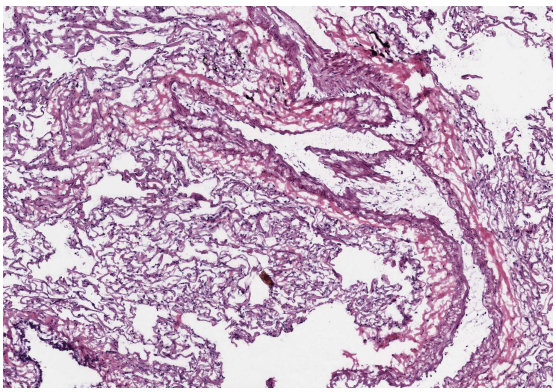


Figure A.8: Sample region of WSI 1 (Resolution: 5995 x 6491)

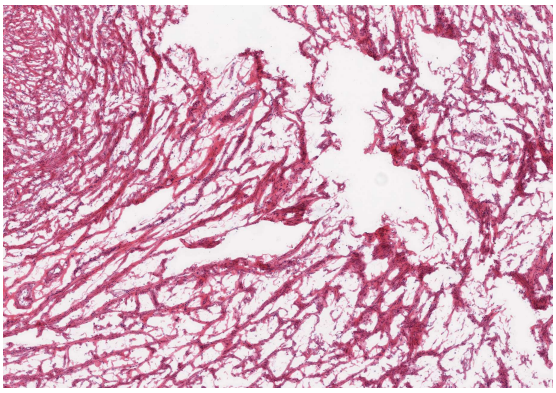


Figure A.9: Sample region of WSI 2 (Resolution: 4500 x 5808)

Figure A.10: Sample region of WSI 3 (Resolution: 4500 x 4504)

Bibliography

Akramullah, S. (2014). *Digital video concepts, methods, and metrics: quality, compression, performance, and power trade-off analysis*, Apress.

Antony, A. and Sreelekha, G. (2015). "Highly efficient near lossless video compression using selective intra prediction for HEVC lossless mode." *$AE\ddot{U}$-International Journal of Electronics and Communications*, 69(11), 1650–1658.

Antony, A. and Sreelekha, G. (2017). "HEVC-based lossless intra coding for efficient still image compression." *Multimedia Tools and Applications*, 76(2), 1639–1658.

Antony, A. and Sreelekha, G. (2017). "Performance enhancement of HEVC lossless mode using sample-based angular and planar predictions." *Signal, Image and Video Processing*, 11(6), 1057–1064.

Bossen, F. (2012). *Common Test Conditions and Software Reference Configurations*, JCTVC-H1100, San Jose.

Bossen, F., Bross, B., Suhring, K. and Flynn, D. (2012). "HEVC complexity and implementation analysis." *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12), 1685–1696.

Chen, H., Saxena, A. and Fernandes, F. (2014). "Nearest-neighbor intra prediction for screen content video coding." *In: Proceedings of IEEE International Conference on Image Processing*, 3151–3155.

Chen, J. and Lee, T. (2011). *Planar Intra Prediction Improvement*, JCTVC-F483, Torino.

Chernyak, R.I. (2014). "Analysis of the intra predictions in H.265/HEVC." *In: Applied Mathematical Sciences*, 8(148), 7389–7408.

da Silva, T.L., Agostini, L.V. and da Silva Cruz, L.A. (2014). "HEVC intra prediction acceleration based on texture direction and prediction unit modes reuse." *APSIPA Transactions on Signal and Information Processing*, 3.

Fraunhofer. (2016). *https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/.*

Guarda, A.F., Santos, J.M., da Silva Cruz, L.A., Assuncao, P.A., Rodrigues, N.M. and de Faria, S.M. (2017). "A method to improve HEVC lossless coding of volumetric medical images." *Signal Processing: Image Communication*, 59, 96–104.

Hang, H.M., Peng, W.H., Chan, C.H. and Chen, C.C. (2010). "Towards the next video standard: high efficiency video coding." *In: Proceedings of APSIPA Annual Summit and Conference*, 609–618.

Haoping, Y., Robert, C., Krishna, R. and Jizheng, X. (2015). *Common Test Conditions for Screen Content Coding*, JCTVC-U1015-r2, Warsaw.

Jeon, G., Kim, K. and Jeong, J. (2014). "Improved residual DPCM for HEVC lossless coding." *SIBGRAPI Conference on Graphics, Patterns and Images*, 95–102.

Juurlink, B., Alvarez-Mesa, M., Chi, C.C., Azevedo, A., Meenderinck, C. and Ramirez, A. (2012). *Scalable parallel programming applied to H.264/AVC decoding*, Springer Science & Business Media.

Kim, K., Jeon, G. and Jeong, J. (2014). "Improvement of implicit residual DPCM for HEVC." *International Conference on Signal-Image Technology and Internet-Based Systems*, 652–658.

Knezovic, J. and Kovac, M. (2003). "Gradient based selective weighting of neighboring pixels for predictive lossless image coding." *In: Proceedings of 25th IEEE International Conference on Information Technology Interfaces*, 483–488.

Kwon, M., Kim, H.J., Lee, C.W. and Lee, S.U. (1999). "A lossless image coder with context-based minimizing MSE prediction and entropy coding." *In: Proceedings of the 1999 IEEE International Symposium on Circuits and Systems VLSI (Cat. No. 99CH36349)*, 4, 479–482.

Lainema, J., Bossen, F., Han, W.J., Min, J. and Ugur, K. (2012). "Intra coding of the HEVC standard." *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12), 1792–1801.

Lee, S., Kim, I.K. and Kim, C. (2013). *RCE2: Test 1 residual DPCM for HEVC lossless coding*, JCTVC-M0079, Korea.

Ohm, J.R., Sullivan, G.J., Schwarz, H., Tan, T.K. and Wiegand, T. (2012). "Comparison of the coding efficiency of video coding standards - including high efficiency video coding (HEVC)." *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12), 1669–1684.

Pourazad, M.T., Doutre, C., Azimi, M. and Nasiopoulos, P. (2012). "HEVC: The new gold standard for video compression: How does HEVC compare with H.264/AVC?." *IEEE Consumer Electronics Magazine*, 1(3), 36–46.

Richardson, I.E. (2002). *Video Codec Design*, John Wiley.

Richardson, I.E. (2010). *The H.264 Advanced Video Compression Standard*, John Wiley.

Rosewarne, C., Bross, B., Naccari, M., Sharman, K. and Sullivan, G. (2016). *High Efficiency Video Coding (HEVC) Test Model 16 (HM 16) Improved Encoder Description Update 6*, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Geneva.

Saloman, D. (2007). *Data Compression*, Springer.

Sanchez, V. (2015). "Lossless screen content coding in HEVC based on sample-wise median and edge prediction." *IEEE International Conference on Image Processing*, 4604–4608.

Sanchez, V., Auli-Llinàs, F., Bartrina-Rapesta, J. and Serra-Sagrista, J. (2014). "HEVC-based lossless compression of whole slide pathology images." *IEEE Global Conference on Signal and Information Processing*, 297–301.

Sanchez, V., Auli-Llinàs, F. and Serra-Sagrista, J. (2016). "Piecewise mapping in HEVC lossless intra-prediction coding." *IEEE Transactions on Image Processing*, 25(9), 4004–4017.

Sanchez, V., Hernandez-Cabronero, M., Auli-Llinàs, F. and Serra-Sagrista, J. (2016). "Fast lossless compression of whole slide pathology images using HEVC intra-prediction." *IEEE International Conference on Acoustics, Speech and Signal Processing*, 1456–1460.

Sayood, K. (2002). *Lossless Compression Handbook*, Academic Press.

Seemann, T. and Tisher, P. (1997). "Generalized locally adaptive DPCM." *Department of Computer Science Technical Report CS97/301*, 1–15.

Stewart, R.J., Lure, Y.M. and Liou, C.S. (1994). "An adaptive technique to maximize lossless image data compression of satellite image."

Sullivan, G.J. and Baker, R.L. (1994). "Efficient quadtree coding of images and video." *IEEE Transactions on Image Processing*, 3(3), 327–331.

Sullivan, G.J., Ohm, J.R., Han, W.J. and Wiegand, T. (2012). "Overview of the high efficiency video coding (HEVC) standard." *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12), 1649–1667.

Sze, V., Budagavi, M. and Sullivan, G.J. (2014). *High Efficiency Video Coding (HEVC): Algorithms and Architectures*, Springer.

Test Sequences. (2012). *ftp://ftp.tnt.uni-hannover.de/*.

Wang, Y., Ostermann, J. and Zhang, Y.-Q. (2002). *Video Processing and Communications*, Prentice Hall.

Wein, M. (2015). *High Efficiency Video Coding: Coding Tools and Specification*, Springer.

Weinberger, M.J., Seroussi, G. and Sapiro, G. (1996). "LOCO-I: A low complexity, context-based, lossless image compression algorithm." *In: Proceedings of IEEE Data Compression Conference*, 140–149.

Wiegand, T., Sullivan, G.J., Bjøntegaard, G. and Luthra, A. (2003). "Overview of the H.264/AVC video coding standard." *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7), 560–576.

Wige, E., Yammine, G., Amon, P., Hutter, A. and Kaup, A. (2013). "Sample-based weighted prediction with directional template matching for HEVC lossless coding." *In: Proceedings of Picture Coding Symposium*, 305–308.

WSI Pathology Images. *https://digitalpathologyassociation.org/*.

Wu, X. and Memon, N. (1997). "Context-based, adaptive, lossless image coding." *IEEE Transactions on Communications*, 45(4), 437–444.

Wu, X. and Memon, N. (1997). "Recent developments in context-based predictive techniques for lossless image compression." *The Computer Journal*, 40(2) and (3), 127–136.

Xu, J., Toshi, R. and Cohen, R.A. (2015). "Overview of the emerging HEVC screen content coding extension." *IEEE Transactions on Circuits and Systems for Video Technology*, 26(1), 50–62.

Zhou, M., Gao, W., Jiang, M. and Yu, H. (2012). "HEVC lossless coding and improvements." *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12), 1839–1843.

List of Publications

Refereed International Journals

- S. Shilpa Kamath, P. Aparna and Abhilash Antony (2018), "Gradient-oriented directional predictor for HEVC planar and angular intra prediction modes to enhance lossless compression", *International Journal of Electronics and Communications (AEÜ)*, Vol. 95, pp. 73–81.
  DOI: 10.1016/j.aeue.2018.07.037

- S. Shilpa Kamath, P. Aparna and Abhilash Antony (2020), "Performance enhancement of HEVC lossless mode using context-based angular and planar intra predictions", *Multimedia Tools and Applications*, Vol. 79 No. 17, pp. 11375–11397.
  DOI: 10.1007/s11042-019-08466-4

- S. Shilpa Kamath, P. Aparna and Abhilash Antony (2020), "Pixelwise improvised blend of predictors in HEVC lossless mode", *International Journal of Electronics and Communications (AEÜ)*, Vol. 114.
  DOI: 10.1016/j.aeue.2019.153000

Refereed International Conferences

- S. Shilpa Kamath, P. Aparna and Abhilash Antony (2017), "Sample-based DC prediction strategy for HEVC lossless intra prediction mode", *IEEE International Conference on Image Information Processing*, pp. 656–660.
  DOI: 10.1109/ICIIP.2017.8313797

Bio-data

Name	: SHILPA KAMATH S.
Address	: Shiva Niwas, Mannagudda, Mangaluru
Email	: shilpa.1107@yahoo.co.in
Mobile	: +91 9482 440940
Qualifications	: B.E. in Telecommunication Engineering, PACE, Mangaluru M.Tech. in Microelectronics, MIT, Manipal
Research Interests	: Video compression, medical image processing