PUCRS

ESCOLA POLITÉCNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
DOUTORADO EM CIÊNCIA DA COMPUTAÇÃO

RAMON COSTI FERNANDES

EXPLORING CURVE-BASED PREDICTION MODELS FOR INTRA-FRAME PREDICTION

Porto Alegre
2021

Pontifical Catholic University of Rio Grande do Sul
School of Technology
Computer Science Graduate Program

EXPLORING CURVE-BASED PREDICTION MODELS FOR INTRA-FRAME PREDICTION

RAMON COSTI FERNANDES

Thesis submitted to the Pontifical Catholic University of Rio Grande do Sul in partial fulfillment of the requirements for the degree of Ph. D. in Computer Science.

Advisor: Prof. Ph.D. César Augusto Missio Marcon
Co-Advisor: Prof. Ph.D. Luciano Volcan Agostini

Porto Alegre
2021

Ficha Catalográfica

F363e Fernandes, Ramon Costi

Exploring Curve-Based Prediction Models for Intra-Frame Prediction / Ramon Costi Fernandes. – 2021.
132.
Tese (Doutorado) – Programa de Pós-Graduação em Ciência da Computação, PUCRS.

Orientador: Prof. Dr. César Augusto Missio Marcon.
Co-orientador: Prof. Dr. Luciano Volcan Agostini.

1. Intra-Frame Prediction. 2. Video Coding. 3. Predictive Coding. I. Marcon, César Augusto Missio. II. Agostini, Luciano Volcan. III. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da PUCRS
com os dados fornecidos pelo(a) autor(a).
Bibliotecária responsável: Clarissa Jesinska Selbach CRB-10/2051

Ramon Costi Fernandes

EXPLORING CURVE-BASED PREDICTION MODELS FOR INTRA-FRAME PREDICTION

This Thesis has been submitted in partial fulfillment of the requirements for the degree of Doctor of Computer Science, of the Graduate Program in Computer Science, School of Technology of the Pontifícia Universidade Católica do Rio Grande do Sul.

Sanctioned on March 23, 2021.

COMMITTEE MEMBERS:

Prof. Dr. Fabiano Passuelo Hessel (PPGCC/PUCRS)

Prof. Dr. Bruno Zatt (PGCC/UFPel)

Prof. Dr. Mateus Grellert da Silva (UFSC)

Prof. Dr. César Augusto Missio Marcon (PPGCC/PUCRS - Advisor)

Prof. Dr. Luciano Volcan Agostini (PGCC/UFPel – Co-Advisor)

EXPLORANDO MODELOS DE PREDIÇÃO BASEADOS EM CURVAS PARA A PREDIÇÃO INTRA-QUADROS

RESUMO

Esta Tese apresenta dois novos modelos de predição intra-quadros: LSAS (superfícies de aproximação baseadas em mínimos quadrados, em inglês Least-Squares Approximation Surfaces) e modelos de predição baseados em curvas. Ambas as abordagens aprimoram a predição intra-quadros, oferecendo uma melhor aproximação do conteúdo de bloco codificado. Embora o LSAS não seja prático devido aos requisitos proibitivos de taxa de bits, os modelos de predição baseados em curvas mostram resultados promissores.

As avaliações dos novos modelos de predição intra-quadros adotam o padrão HEVC (codificação de vídeo de alta eficiência, em inglês High Efficiency Video Coding), que culmina anos de avanços em tecnologias de codificação de vídeo. Em comparação com seu antecessor H.264, o HEVC alcança até 50% de melhorias na eficiência de codificação. Isso se traduz em metade do tamanho de um vídeo codificado, mantendo a mesma qualidade visual. Entre as muitas melhorias do HEVC, o seu preditor intra-quadros vem com mais modos de predição, capazes de modelar mais texturas em blocos de predição do que seus antecessores. Melhorar a predição intra-quadros é um aspecto importante do fluxo de codificação, pois uma melhor predição se traduz em redução da energia residual, consequentemente melhorando a eficiência da codificação.

Todos os trinta e três modos angulares do HEVC receberam um cálculo de deslocamento de curva para cada amostra predita para que a predição de blocos modele regiões de imagem com texturas curvas. A proposta inclui um pequeno custo adicional nos elementos de sintaxe do vídeo codificado, para incorporar o valor de deslocamento da curva, e um aumento no tempo de codificação. No entanto, a predição aprimorada compensa esse custo com uma maior eficiência de codificação. Utilizando os modelos baseados em curvas Centerline e Radial, para aplicar o cálculo de deslocamento, os resultados experimentais demonstram maior precisão da predição com menor energia residual, alcançando uma redução média na taxa de bits Bjøntegaard-Delta (BD-Rate) de 2% e 3% para as sequências de teste HEVC na configuração All-Intra-8, para os modelos Centerline e Radial, respectivamente, porém com um aumento no tempo de codificação. Uma maior eficiência de codificação é alcançável com a combinação de ambos os modelos de curvas, chegando a uma redução média de 4% BD-Rate na configuração All-Intra-8.

Palavras-Chave: Predição Intra-quadros, Codificação de Vídeos, Codificação Preditiva.

EXPLORING CURVE-BASED PREDICTION MODELS FOR INTRA-FRAME PREDICTION

ABSTRACT

This Thesis presents two novel intra-frame prediction models for video coding: Least-Squares Approximation Surfaces (LSAS) and curve-based prediction models. Both approaches enhance intra-frame prediction capabilities by offering a better approximation of encoded block contents. While the LSAS implementation is currently not practical due to its prohibitive bitrate requirements, the curve-based prediction models show promising results.

The evaluations for the novel intra-frame prediction models adopt the High Efficiency Video Coding (HEVC) standard, which culminates years of advancements in video coding technologies. Compared to its predecessor, H.264, HEVC achieves up to 50% coding efficiency improvements. This efficiency gain translates into half the encoded video size while keeping the same visual quality. Among the many improvements of HEVC, its intra-frame predictor comes with more prediction modes, capable of modeling more textures in prediction blocks than its predecessors. Improving intra-frame prediction is an essential aspect of the encoding flow, as a better prediction translates into reduced residual energy, consequently improving coding efficiency.

All thirty-three angular modes in HEVC received a curve displacement calculation to each predicted sample so that the resulting prediction block models image regions with curved textures. The proposal includes a small overhead in the bitstream syntax elements to transmit the curve displacement value and increased encoding times. However, the enhanced prediction offsets this overhead with improved coding efficiency. The experimental results demonstrate increased prediction accuracy with lower residual energy when applying the Centerline and Radial curve-based models for the displacement calculations. These models achieve an average reduction of the Bjøntegaard-Delta bitrate (BD-Rate) of 2% and 3% for the HEVC test sequences using the All-Intra-8 configuration, for the Centerline and Radial models, respectively, albeit at an increase in encoding times. Higher encoding efficiency is achievable with the combination of both curve models, reaching an average 4% BD-Rate reduction with the All-Intra-8 configuration.

Keywords: Intra-frame prediction, Video Coding, Predictive Coding.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AOM – Alliance for Open Media

AV1 – AOMedia Video 1

AVC – Advanced Video Coding

BD-Rate – Bjøntegaard Delta Bitrate

CABAC – Context-Adaptive Binary Arithmetic Coding

CAVLC – Context-Adaptive Variable-Length Coding

CBs – Coding Blocks

CDW – Curve Displacement Weight

CTB – Coding Tree Block

CTC – Common Test Conditions

CTU – Coding Tree Unit

CU – Coding Unit

DC – Direct Component

DCT – Discrete Cosine Transform

DCT-IF – Discrete Cosine Transform Interpolation Filter

FHD – Full High Definition

FPS – Frames per Second

GOP – Group of Pictures

GSE – Grupo de Sistemas Embarcados

HD – High Definition

HEVC – High Efficiency Video Coding

HEVC-SCC – HEVC Screen Content

HM – HEVC test Model

HVS – Human Visual System

IDCT – Inverse Discrete Cosine Transform

IEC – International Electrotechnical Commission

ISO – International Organization for Standardization

ITU – International Telecommunication Union

ITU-R – ITU for Radiocommunication

ITU-T – ITU Telecommunications Standardization Sector

JCT-VC – Joint-Collaborative Team for Video Coding

JVET – Joint Video Exploration Team

Kbps – Kilobits per second

LSAS – Least-Squares Approximation Surfaces

Mbps – Megabits per second

MCP – Motion Compensated Prediction

MPEG – Moving Picture Experts Group

MPM – Most Probable Mode

MRL – Multi-Reference Line

MSE – Mean Squared Error

MTT – Multi-Type Tree

PB – Prediction Block

PDPC – Position Dependent Prediction Combination

PSNR – Peak Signal-to-Noise Ratio

PU – Prediction Unit

QCIF – Quarter Common Intermediate Format

QHD – Quad High Definition

QP – Quantization Parameter

RD-cost – Rate-Distortion cost

RD-list – Rate-Distortion list

RDO – Rate-Distortion Optimization

RGB – Red Green Blue

RMD – Rough Mode Decision

RQT – Residual Quadtree

SAD – Sum of Absolute Differences

SATD – Sum of Absolute Transformed Differences

SCC – Screen Content Coding

SLSP – Sparse Least Squares Prediction

TB – Transform Block

TU – Transform Unit

UHD – Ultra High Definition

VCEG – Video Coding Experts Group

VVC – Versatile Video Coding

CONTENTS

1. INTRODUCTION

The digital video applications, which have arisen in the past years, identify a growing demand for high-quality video, e.g., using high resolutions, high frame rates, and high fidelity. Uncompressed videos generate an enormous amount of data [61]. To understand this statement, consider the following exercise: a video with a screen resolution of 1920 pixels wide and 1080 pixels high (referred to as 1080p resolution), composed of three-color layers (red, green, and blue) and with each pixel in each layer requiring 8 bits to represent the color value. Each frame of this video requires $1920 \times 1080 \times 8 \times 3$ = 49766400 bits ($\approx$ 6 MB). Since movies usually display, at least, 25 Frames Per Second (FPS), each second of this movie requires 25×6 = 150 MB of data. Assuming that movies usually last around two hours, which translates to $2 \times 60 \times 60$ = 7200 seconds, a 1080p color movie, running at 25 FPS, requires nothing short of 7200×150 = 1080000 MB: this corresponds to a terabyte of data. This simple exercise shows why video coding (or video compression) is indispensable, with applications in fields like television, streaming services, home surveillance, video conferencing, and many others.

Table 1.1 shows the typical screen resolutions for 720p High Definition (HD), 1080p Full High Definition (FHD), 1440p Quad High Definition (QHD), 4K Ultra High Definition (UHD), 8K UHD, and 16K UHD. For reference, recapitulating the earlier exercise where a single frame is $\approx$ 6 MB of data at 1080p, at 4K resolution, there are four times as much data. Likewise, the 8K resolution has four times the definition of 4K, and 16K has four times the definition of 8K, or 64 times more data than FHD. Figure 1.1 illustrates the scale and the visible area of the standard resolutions from 720p up to 16K.

Table 1.1 – Standard resolutions found on most and upcoming video content.

Resolution	Height	Width	Megapixels
720p HD	720	1280	0.92
1080p FHD	1080	1920	2.07
1440p QHD	1440	2560	3.69
4K UHD	2160	3840	8.29
8K UHD	4320	7680	33.18
16K UHD	8640	15360	132.71

The availability of video content would not be possible without the compression obtainable by video coding standards, including the H.26X and open-licensed VPX coding families. Examples of the H.26X coding family are the widely adopted MPEG-4 Part 10, H.264 Advanced Video Coding (AVC) [72], its successor H.265 High Efficiency Video Coding (HEVC) [60], and the recently released Versatile Video Coding (VVC) [5]. Meanwhile, some encoders from the VPX family are VP8 [2], its successor VP9 [41], and the recent AOMedia Video 1 (AV1) [47].

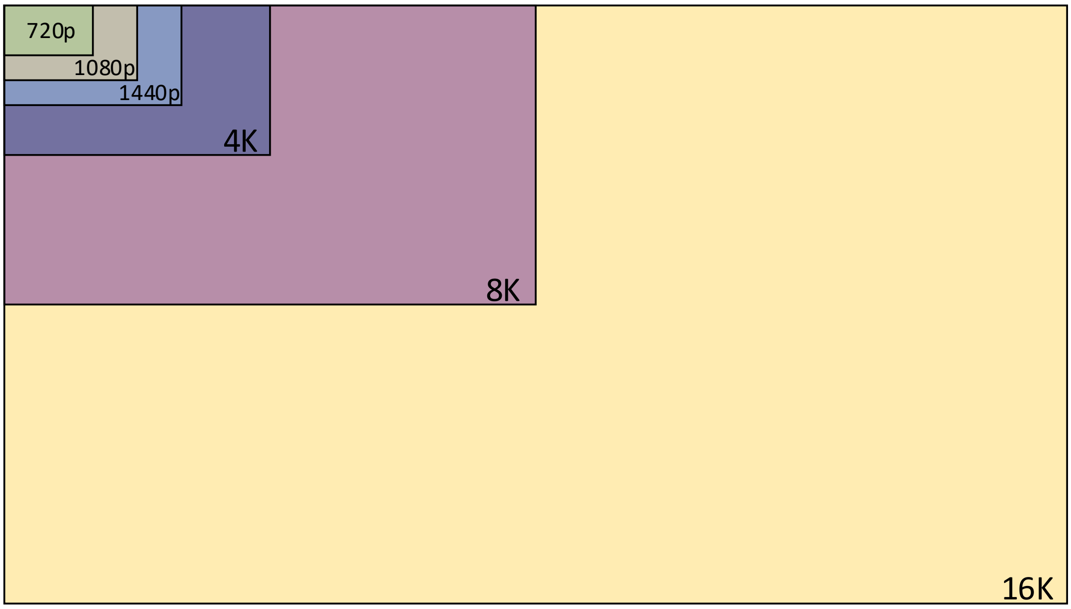


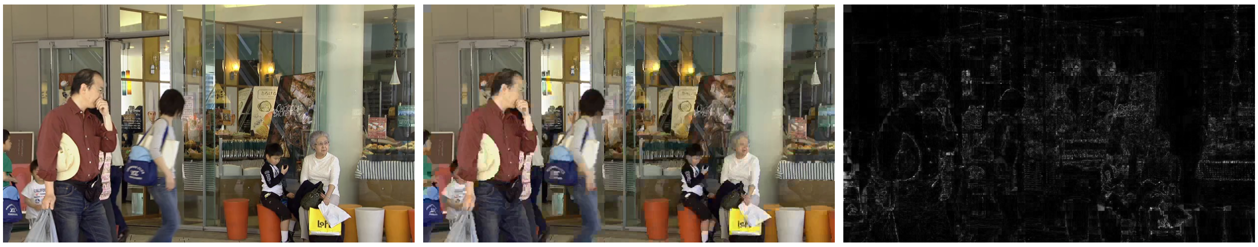
Figure 1.1 – Comparison of the visible area of different resolutions, from 720p to the upcoming 16K.

As the demand for higher definition videos increases, the next generation of video coding standards [10] must develop technologies that enable the efficient storage and distribution of video content. According to a study conducted by Cisco Systems Inc. [11], in 2016, 73% of all Internet traffic consisted of video content, and by 2021 Cisco Systems expects this ratio to increase to 82%. Therefore, the small bitrates with minimal perceivable quality degradation, the encoding times, among other requirements, are the commonly found goals in the standardization of the state-of-the-art in video coding. Video encoders adopt advanced data structures [27, 37], seeking the best trade-off between coding quality, bitrate, and computational effort to address the current and future requirements. As videos are usually encoded once, but decoded countless times, even small bitrate savings are of significant interest due to the limited communication and storage infrastructure on which the content distribution relies.

The main task of video coding is to compress and map raw input data, i.e., a temporally ordered sequence of two-dimensional digital pictures, into a bitstream that can be transmitted over a transmission channel [71]. A typical encoder can compress the enormous amounts of data in videos by a factor that ranges from 10 : 1 to 1000 : 1 [71]. Encoders achieve such high compression factors employing lossy compression, which are coding techniques that generate an approximation of the original input video instead of an exact reconstruction upon decoding.

Nowadays, all widely used video coding standards adopt a model referred to as Hybrid Video Coding, which consists of the combination of three fundamental methods, namely: (i) *Intra* and *Inter-frame prediction*; (ii) *Transform coding*; and (iii) *Entropy coding*. Intra-frame prediction explores the natural spatial redundancy found in video frames [29], while inter-frame prediction uses redundant information present in past-encoded frames [45]. The idea is to reconstruct the video frames using information already available in the partially reconstructed video; thus, reducing the amount of data required for reconstructing as approximation of the original input. Naturally, this predictive method generates prediction errors, referred to as prediction residuals [73]. Transform coding explores the spatial redundancies within the prediction residuals. During the encoding step, transform coding maps the residuals into the frequency domain with a reversible linear transform function, e.g., the Discrete Cosine Transform (DCT) [43]. The transformed residuals are then quantized, i.e., a precision reduction function reduces the dynamic range of the residuals: this is the most important lossy step of the encoding process, which discards the less visually relevant information of a video frame. Lastly, entropy coding packs the quantized residuals from the encoding process and the high-level syntax elements into the video bitstream losslessly. The decoding step performs the inverse operations, i.e., the decoder unpacks the bitstream, dequantizes the quantized residuals, transforming them back to the spatial domain, and, combined with the prediction data, produces an approximation of the original video.

Figure 1.2 exemplifies intra-frame prediction. Figure 1.2(a) depicts a video frame encoded with HEVC; Figure 1.2(b) corresponds to the intra-frame prediction, while the difference between Figure 1.2(a) and Figure 1.2(b) is the residuals shown in Figure 1.2(c).

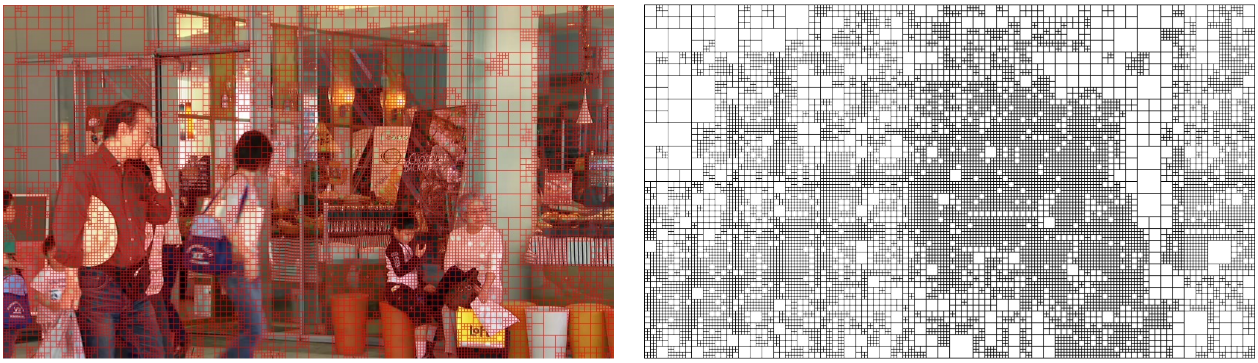


(a) Encoded frame (b) Intra-frame prediction (c) Intra-frame prediction residuals

Figure 1.2 – An encoded video frame and its intra-frame prediction residuals, obtained with the HEVC encoder; a better prediction quality results in reduced residuals amount.

Before computing the prediction and residuals, hybrid video coders segment frames into rectangular regions, creating a block-based structure such as the one exemplified in Figure 1.3 with the HEVC encoder. The contents of each block are intra- or inter-frame predicted using information from spatial or temporal neighboring blocks, resulting in a predicted block and the corresponding residuals. As such, each block represents an individual unit in the coded video, storing information such as the prediction used, residuals, among other syntax elements employed by the block-based hybrid video coding standard.

The block structure plays an essential role in video coding efficiency. Advances in coding standards and computational power availability have enabled larger block sizes for segmenting video frames, which is beneficial for high-resolution videos [61]. Prior video



(a) Segmented frame

(b) Segmentation structure

Figure 1.3 – Frame segmentation obtained with the HEVC encoder. The segmentation map applies to the entire frame, dividing each rectangular region into an individual information unit.

coding standards such as H.262, H.263 [46], VP8, and the well-known and still widely used H.264 adopt a *macroblock* of size 16 × 16 pixels. The macroblock corresponds to the largest block size on these coding standards, where the encoder segments video frames into regions according to the macroblock dimension. Depending on the features available in the coding standard, macroblock subdivision into subblocks may occur for refining intra- or inter-frame prediction. Although HD and UHD video content encoding still uses some of these older coding standards, the macroblock structure was designed for lower resolution content [61], from the Quarter Common Intermediate Format (QCIF) [81] at resolution 176 × 144 up to the standard definitions at resolutions 720 × 480 or 720 × 576. Due to the popularity of high-resolution video content, restricting the macroblock to 16 × 16 becomes inefficient as in HD and UHD content, a single intra- or inter-frame prediction parameter can describe many picture areas larger than 16 × 16. Modern coding standards like VP9 and HEVC adopt the *superblock* and the *Coding Tree Unit* (CTU), respectively, as their largest block, each containing up to 64 × 64 samples. Although different in name, the superblock and CTU use a quadtree [50] structure for recursively subdividing blocks, as illustrated by Figure 1.3(b). Even though larger block sizes enable better coding efficiency, they increase encoding and decoding delays, requiring more memory and higher computational complexity for the encoder [60]. However, increasing the encoder complexity is often acceptable as most video applications require encoding once but decoding countless times.

Although the process described in the earlier paragraphs correspond to most of the video compression effort, another element involved is *entropy coding*. For losslessly compressing the high-level syntax elements, video coding standards adopt an entropy coder [61], which explores the statistical properties of data generated by the coding process. The frequently occurring elements of the encoding process, like the high-level syntax elements and the quantized prediction residuals, can be encoded with fewer bits, further reducing the video bitstream. Meanwhile, the encoding of less often occurring information can use more bits without affecting the compression rate. The H.264 standard adopts two entropy coding models: the first adopts a code word table for all syntax elements except the quantized residuals [72], which uses Context-Adaptive Variable Length Coding (CAVLC). The second approach adopts Context-Adaptive Binary Arithmetic Coding (CABAC), enabling greater efficiency for compression of both high-level syntax elements and the quantized residuals (an improvement of 5% to 15%). CABAC is also the technique used by HEVC. VP8 and VP9 employ Boolean Arithmetic Coding, referred to as BoolCoder [21], which is similar in both theory and implementation to CABAC. Meanwhile, AV1 adopts a new entropy coder, which employs arithmetic coding with a sixteen-symbol alphabet instead of the two-symbol alphabet of CABAC.

1.1 Problem Statement

As of the time of writing this Thesis, HEVC presents better coding efficiency than its contemporary rival VP9 [16, 22, 67], and in most cases to the newer AV1 [1, 23, 66]. While the recently released VVC achieves improved coding efficiency compared to HEVC, the development of this Thesis was near its completion upon the final release of VVC. Therefore, this Thesis adopts HEVC as the baseline for the proposed intra-frame prediction models, although the technique may benefit all block-based coding standards. Section 2.10 presents a more thorough and technical comparison of encoding efficiency between HEVC and other coding standards.

Figure 1.4 presents the high-level flow of the video encoding and decoding processes, including frame partitioning and reconstruction, block prediction and reconstruction, residual transformations, quantization and dequantization, and entropy encoding and decoding. The main contribution of this Thesis is focused on the prediction step, which is highlighted in Figure 1.4. Although Figure 1.4 provides a simplified overview of the encoding and decoding flows, all processes are interdependent to some degree, depending on the coding standard.

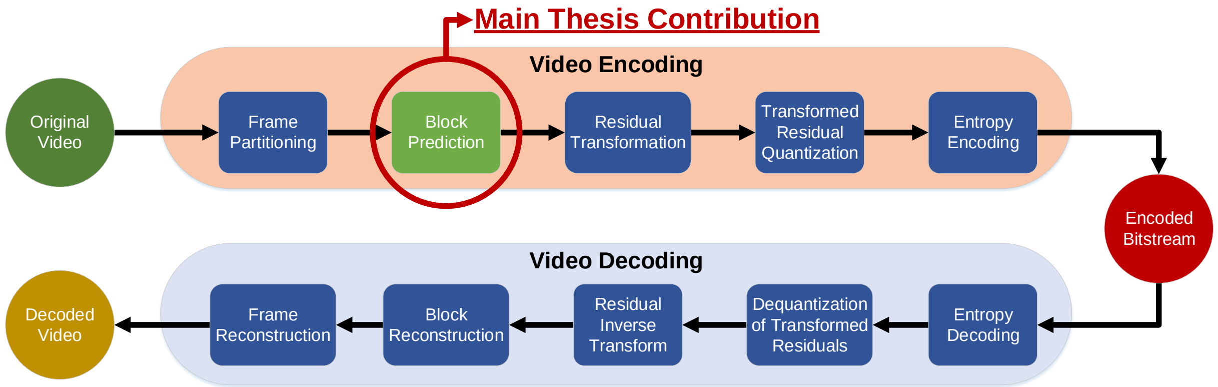


Figure 1.4 – A simplified representation of the video coding flow. The interdependence among the flow processes depends on the coding standard.

The prediction process plays a crucial role in the encoded video quality and efficiency; a high-quality prediction reduces the residuals, consequently reducing the bitstream size and/or improving the decoded video quality. Therefore, the investigation of novel prediction approaches for video coding becomes an exciting research topic for current and next-generation standards. Some new intra-frame prediction methods exist for enhancing video coding standards, such as extensions of HEVC — 3D-HEVC [62] and HEVC Screen Content (HEVC-SCC) [79]; however, fewer works exist focusing on improvements to the intra-frame prediction algorithms of HEVC [9, 20, 36, 38].

1.2 Main Objectives and Thesis Hypothesis

Given the exposed motivation and problem description, this Thesis aims to improve the block prediction process, as Figure 1.4 illustrated. The improvements target the HEVC standard with novel techniques for intra-frame prediction, improving the existing intra-prediction capabilities of HEVC, thus improving coding efficiency. Although this work targets HEVC, the ideas explored in this Thesis can be extrapolated to other coding standards. Meanwhile, the proposed techniques should aim to keep encoding complexity on par with state-of-the-art that also addresses enhanced intra-frame prediction. To achieve this goal, this Thesis fulfills a set of specific aims:

- Perform an in-depth study on the state-of-the-art of intra-frame prediction, identifying the techniques employed in current and previous video coding standards. Such exploration should provide insights about the capabilities and limitations of current intra-frame prediction methods;

- The definition of one or more solutions to the intra-frame prediction problem, with the subsequent implementation and validation of these solutions into the HEVC test Model (HM) [48] to evaluate the resulting coding efficiency. This step enables the later improvement of the most promising solutions and their adoption to develop newer solutions; and

- For the validated and chosen solutions, identify their associated drawbacks to achieve an efficient implementation of each model. This aim characterizes the refinement of the proposed solutions into the HM.

Having the set of objectives described, this Thesis has the following underlying hypothesis: “it is possible to propose new intra-frame prediction models that reduce the bitstream of the encoded videos while keeping the same objective video quality.”

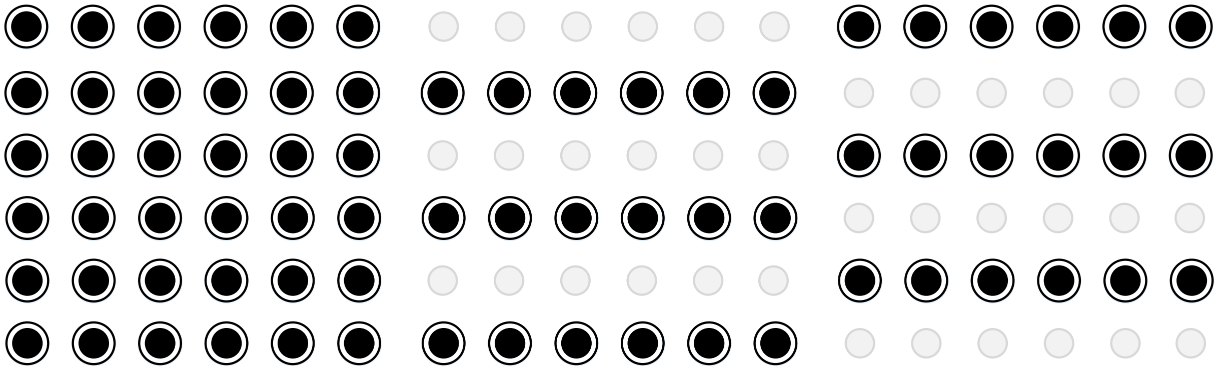
1.3 Document Structure

The organization of this Thesis follows six chapters. To provide a basis for readers to understand the fundamental concepts addressed in this Thesis, Chapter 2 presents an overview of video coding standards and theory. Chapter 3 details aspects specific to the HEVC standard, contemplating elements of its encoding flow like block structures and intra-frame prediction. Chapter 4 presents an overview of recent contributions in intra-frame prediction for video coding standards. To demonstrate the main Thesis contributions, Chapter 5 details the research performed to achieve the enhanced intra-frame prediction models, detailing the mathematical models and the necessary modifications of the HEVC standard to provide the proof of concept. Chapter 6 evaluates the proposed models for enhanced intra-frame prediction, supplying the evidence that validates the proposal of this Thesis. The extensive results contemplate aspects like the residual energy of predicted samples, coding efficiency, encoder and decoder complexity, comparing the results with the state-of-the-art. Finally, Chapter 7 concludes this Thesis, discussing the significance of this research and proposing future works.

2. VIDEO CODING OVERVIEW

Video compression is not a recent science; the first form of video compression dates back to 1929 when Ray Davis Kell patented what would be later known as inter-frame prediction [25], many years before a practical application for his technique. Due to the strong temporal correlation in a sequence of video images, i.e., the video frames, his idea consisted of transmitting the initial frame in its original form, while subsequent frames would carry only the difference from their immediately preceding frame.

Early forms of video compression employed *interlacing*, used since analog television broadcasts [25]. Considering that a two-dimensional space, composed of lines and columns, can represent video frames, interlacing transmits lines in an alternate form, e.g., even frames $0, 2, 4, \ldots, n$ carry the even (Figure 2.1(c)) or the odd (Figure 2.1(b)) lines, while odd frames $1, 2, 3 \ldots, n - 1$ carry the opposite lines of the even frames. This technique effectively halves the required amount of data for a video. If the frame rate is “high enough”, an interlaced video offers good visual fidelity, as interlacing relies on a characteristic of the HVS to integrate parts of information into a coherent whole. Interlacing is still in use today, although much less than progressive scan (Figure 2.1(a)) due to video quality reasons. Progressive and interlaced scanning is often identified in video standards by a single character appended to the resolution used in the video content, e.g., 1080p for denoting an FHD resolution using progressive scan or 1080i for the same resolution using interlaced scan.



(a) Progressive scan (b) Odd lines interlaced scan (c) Even lines interlaced scan

Figure 2.1 – Progressive scan and Interlaced scan: in (a), all lines are rendered from top to bottom; meanwhile, in Interlaced scan, only the odd (b) or the even (c) lines are rendered, also from top to bottom.

It was not until 1984 that the first recommendation for standardizing digital video coding came to be. The International Telecommunication Union (ITU) defined the H.120 standard [63], a codec for videoconferencing, which employed quantization and variable-length coding techniques to transmit video over dedicated point-to-point communication lines [25]. However, this standard had missing elements for practical use, only addressed by a later recommendation issued in 1988 [44]. As such, H.261, ratified in 1988, is considered the basis for modern video coding [44, 46].

Most digital video compression used today is lossy and based on *Rate-Distortion Theory* [71], as it does not reproduce every bit of every sample of a video frame precisely [44]. Video encoders use rate-distortion to minimize the bitrate necessary to encode videos while keeping acceptable quality, despite the loss of information of the encoding process. The following subsections detail some key concepts adopted by currently available coding standards, serving as a baseline for understanding this Thesis.

2.1 Video Coding Terminology

This section briefly lists some of the most used terms in video coding, which this Thesis will use for its demonstrations and explanations, according to the terminology employed in state-of-the-art [61, 70, 71]. Some terms are specific to elements in the video coding process, while others convey an interchangeable meaning according to context, as shown in Table 2.1.

Table 2.1 – Common terminology for video coding.

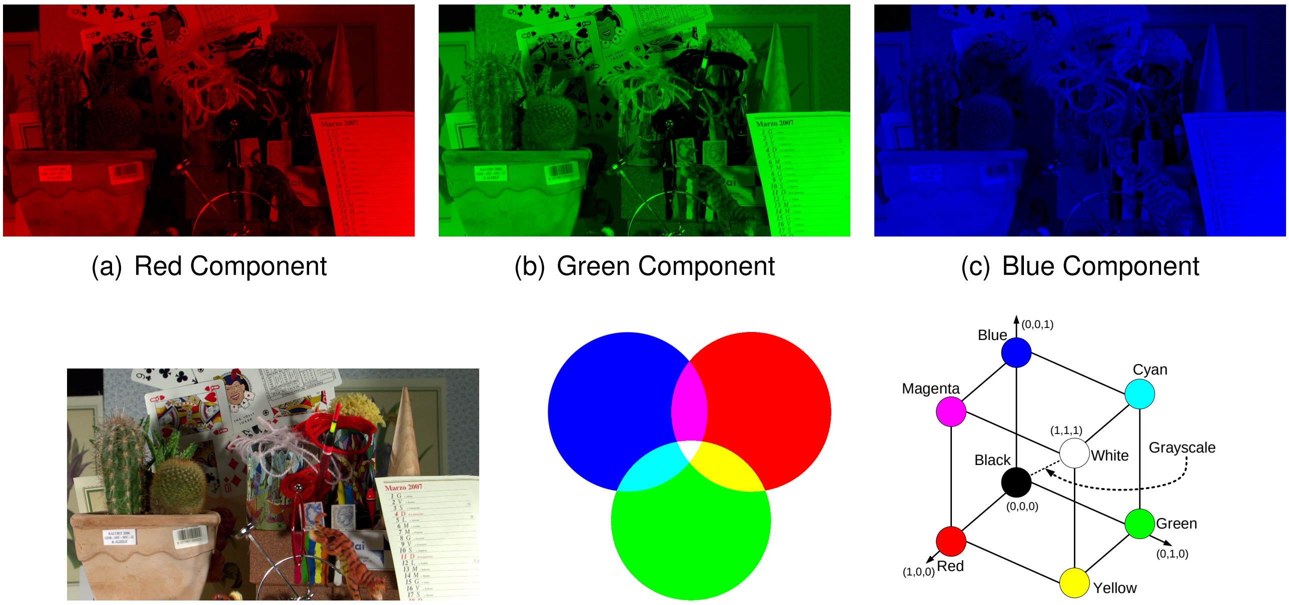
Bitrate	Refers to the number of bits per amount of time of the video stream, usually expressed in kilobits or megabits per second
Frame and Image	Both terms are employed interchangeably in this Thesis. A video frame is an image, and vice versa, corresponding to a still picture of a video
Frame rate	Refers to how many frames are shown for each second of video; most videos employ 24, 30, or up to 60 FPS
Quantization	Refers to the process of constraining values to a given interval. Usually employed to limit residuals originated from the prediction processes of video encoding, turning most values to zero
Bitstream	Refers to the encoded video data in binary format, encompassing the actual compressed video data and syntax elements of the coding standard
Resolution	Refers to the dimensions of a video frame, expressed as pixel width × pixel height

Samples	Refers to the color channels of the pixels that compose a video frame, i.e., in the RGB color space, a colored pixel is composed of a red sample, a blue sample, and a green sample
Residuals	Refers to errors originated from the prediction processes in video encoding
Texture	Refers to the characteristics of an image region. Textures may be complex or smooth, indicating the amount of contrasting differences in the image region
Natural Images	Refers to images originated from real-world events, e.g., most content obtained from filming with cameras and realistic computer generated graphics
Synthetic Images	Refers to artificially generated images, e.g., textual content and computer screen recordings
Codec	Refers to the encoder and decoder pair of a video coding standard
Artifacts	Refers to noise or anomalies in the video frames originated from the encoding process

2.2 The YUV Color Space and Subsampling

In digital media, such as television and computers, images employ an additive color model known as RGB, illustrated in Figure 2.2, whose name originates from the initials of its three primary colors, i.e., red, green, and blue. Usually, digital media encodes color with eight bits of precision, ranging from zero as the darkest value up to 255 as the brightest. The additive combination of RGB with $r = 0$, $g = 0$ and $b = 0$ corresponds to pure black, while $r = 255$, $g = 255$ and $b = 255$ to pure white. The RGB spectrum in 8-bit precision allows for $256^3 \approx 16 \times 10^6$ distinct colors and requires 24 bits to encode each pixel. Figure 2.2 illustrates the individual RGB components of an image and its resulting reconstruction.

Although RGB is a common way to encode colors, there are other possibilities. The YUV model encodes brightness separately from color information [39]. Like RGB, YUV uses three separate channels for color representation, namely: (i) **Y**, for representing *luminance*; (ii) **U**, for representing the *blue chrominance*; and (iii) **V**, for representing the *red chrominance*. The luminance corresponds to the color brightness value, derived by the weighted sum of each color component of RGB. The Recommendation BT.709, ratified by the ITU for



(a) Red Component

(b) Green Component

(c) Blue Component

(d) Original

(e) RGB additive properties

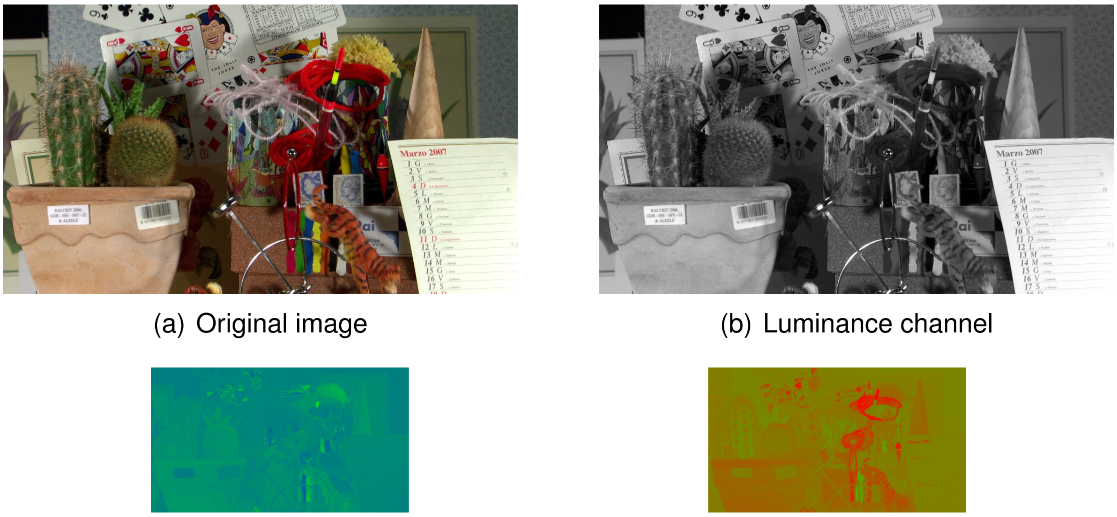
(f) RGB color space

Figure 2.2 – A digital image decomposed into its primary RGB colors. Combining (a), (b), and (c) produces the image (d), according to the additive properties shown in (e). Meanwhile, (f) shows the possible color combinations of the RGB space.

Radiocommunication (ITU-R) [42], standardizes luminance as $Y = 0.2125R + 0.7154G + 0.0721B$. This formula reflects the difference in color sensitivity of the Human Visual System (HVS); the blue light appears dimmest, the green light appears brightest, while the red light falls in between the other two colors. The chrominance corresponds to the difference of the luminance Y to the RGB color channels R and B. Meanwhile, the blue chrominance is $U = B - Y$, and the red chrominance is $V = R - Y$. With the YUV information, it is possible to reconstruct the RGB representation losslessly.

The YUV system dates to analog televisions. Early black-and-white televisions required a single channel for encoding the luminance (grayscale) signal. The YUV format was still in use after the introduction of color television to keep backward compatibility of color broadcast signals with black-and-white televisions. Old black-and-white televisions receive the YUV signal, using the Y channel to display video while discarding the chrominance channel. Meanwhile, newer television devices could reconstruct the original RGB color signal by combining the YUV channels.

So why is YUV still used today in digital videos, where color reproduction is certainly available? Assuming 8-bit precision for encoding each of the three YUV channels, it requires the same 24 bits per sample as RGB. The answer comes from the HVS. The human eye is less sensitive to changes in hue than to changes in brightness. Therefore, it is possible to reduce the amount of information in the chrominance channels of YUV without sacrificing the perceived image quality [39]. A common approach is to halve the chrominance resolution in the horizontal and vertical directions using a technique similar to interlaced scan, called Chrominance Subsampling [68], illustrated in Figure 2.3.



(a) Original image

(b) Luminance channel

(c) Blue chrominance

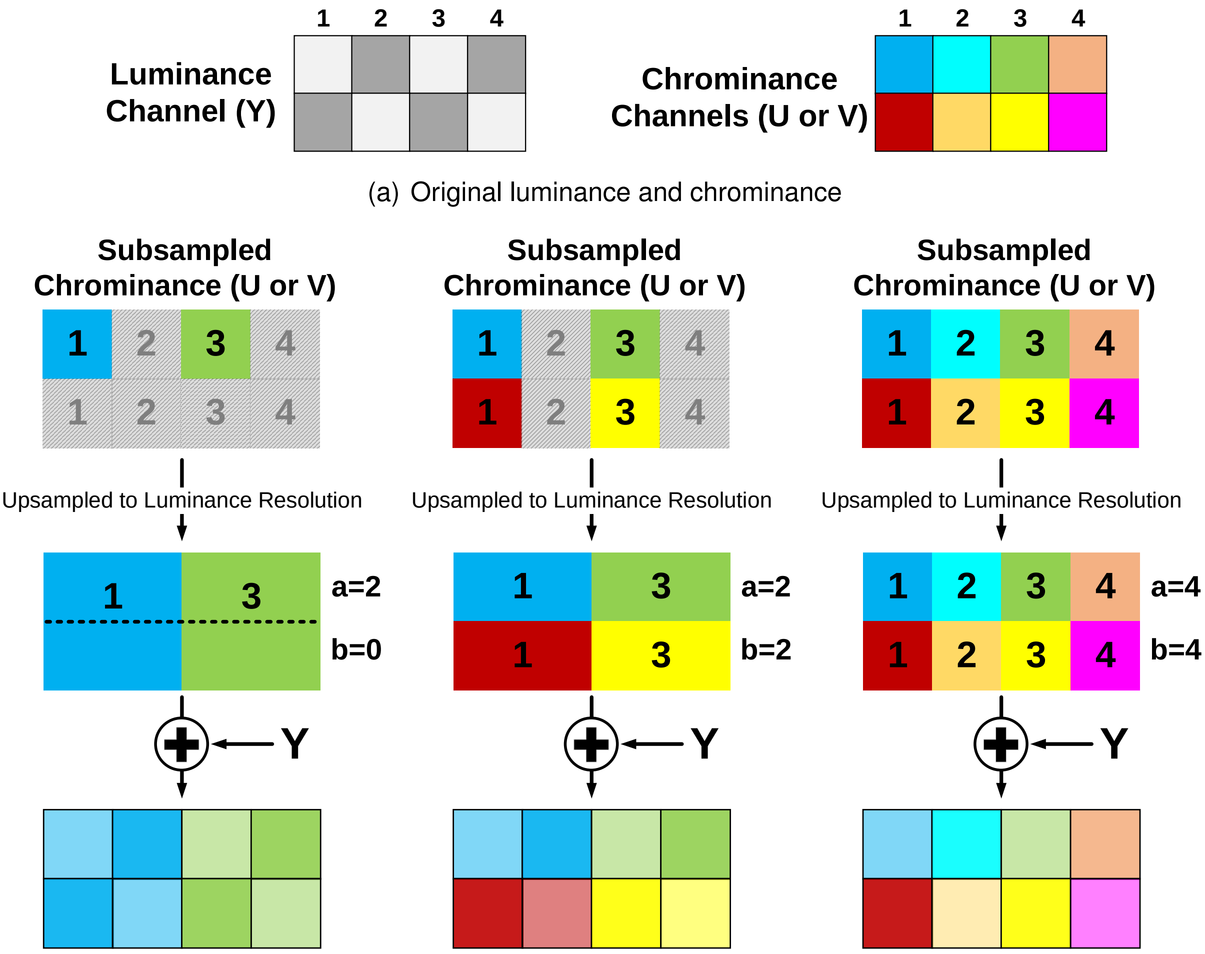
(d) Red chrominance

Figure 2.3 – A digital image decomposed into YUV format, using 4:2:0 chrominance subsampling. Each chrominance channel has 1/4 the resolution of the luminance channel.

Chrominance subsampling preserves the luminance channel while discarding rows and columns of the chrominance channels. The technique considers an image region composed of two rows and four columns of pixels, using the standard notation J:a:b, where: (i) J is the number of luminance samples taken from each row; (ii) a is the number of chrominance samples taken from the first row of the 2 × 4 region; and (iii) b is the number of chrominance samples taken from the second row of the 2 × 4 region.

The most used chrominance subsampling format is 4:2:0 [40], shown in Figure 2.4(b). This format considers the four available luminance samples of the first and the second rows, but only two chrominance samples of the first row and no chrominance samples from the second row. Chrominance subsampling 4:2:0 translates into using only every other column and every other row of the chrominance image region, reducing the color resolution to one quarter of its original size. Another subsampling format, 4:2:2, illustrated in Figure 2.4(c), employs a similar approach, but it does not skip the second row like in 4:2:0. Another possibility is not to discard the chrominance information at all, using 4:4:4 subsampling, which is equivalent to no subsampling. As shown in Figure 2.4(d), chrominance subsampling 4:4:4 keeps the same resolution as the original chrominance channel.

Chrominance subsampling in YUV encoding is a simple and efficient compression technique. Using 4:2:2 subsampling, YUV requires, on average, sixteen bits per sample compared to 4:4:4 subsampling or RGB, corresponding to a 33% reduction in video size. This decrease is quite substantial for a small trade-off in perceptual quality, and even more with 4:2:0 subsampling, requiring an average of 12 bits per sample: a 50% reduction in size.


Luminance Channel (Y)
1
2
3
4
Chrominance Channels (U or V)
1
2
3
4
Subsampled Chrominance (U or V)
1
2
3
4
1
2
3
4
Upsampled to Luminance Resolution
1
3
a=2
b=0
a=2
b=2
a=4
b=4
Y

(a) Original luminance and chrominance

(b) 4:2:0

(c) 4:2:2

(d) 4:4:4

Figure 2.4 – Chrominance subsampling examples. 4:2:0 subsampling (b) is the most used format in video coding due to its compression efficiency while retaining image fidelity.

2.3 Frame Partitioning and Scanning Order

Since the initial video coding proposals with H.261 [64], until the latest AV1 standard [47], video coding has used a block structure to segment video frames into smaller units, similar to the one shown in Figure 2.5. Since the luminance channel is the most relevant for keeping the perceptual quality of an encoded video, frame partitioning in video encoders typically uses this channel when deciding the proper partitioning scheme. Meanwhile, chrominance channels inherit the same partitioning layout from the luminance channel blocks.

The decision on how to partition the frames depends on the coding standard and encoding parameters; H.262, H.263, and H.264 use a fixed-sized macroblock of 16 × 16 luminance samples. The encoder may further subdivide these macroblocks into smaller regions depending on the prediction mode used. Like HEVC and AV1, other standards employ a quadtree structure with a maximum initial size of 64 × 64 and 128 × 128 luminance samples,

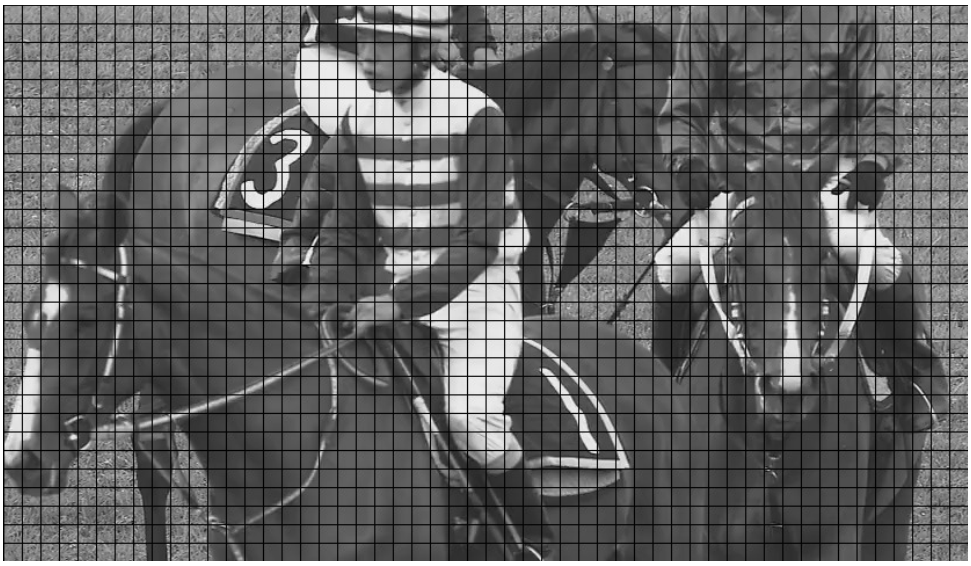
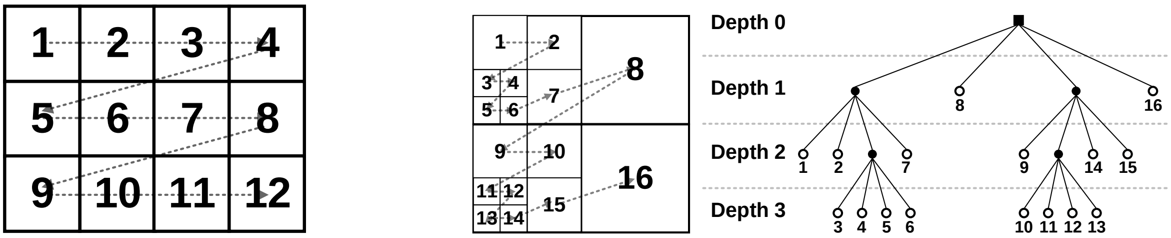


Figure 2.5 – Frame partitioning using H.264 macroblocks for the luminance channel.

respectively, with the possibility of subdivisions down to a 4 × 4 block. VVC, the successor of HEVC, also employs the 128 × 128 luminance samples as the largest block size. The reason for increasing partition sizes is the increasingly higher content resolution. Partitioning a QHD frame with 16 × 16 blocks, as used in previous coding standards, becomes inefficient since the output video bitstream must signal several frame-partitioning units. Besides, image segments sometimes consist of large regions of mostly homogenous texture, representable by only a few large blocks. Due to the rate-distortion algorithms employed in video encoders, the decision to subdivide a block usually occurs to better encode image regions with complex texture detail, which would result in significant quality degradation if encoded by larger blocks.

Another important consideration is the block scanning order. Given the frame partitioning, the encoder must signal to the decoder how to reconstruct blocks for each frame. The macroblock-based coding standards typically use raster-scan, shown in Figure 2.6(a), following a line-wise order from the upper-left to the lower-right block. Meanwhile, block traversal uses depth-first or z-scan order for quadtree-based block structures, as illustrated in Figure 2.6(b). Raster-scan and z-scan orders also ensure that the codec will have previously decoded blocks available for block reconstruction, except for blocks found at the topmost row and the leftmost column of a frame. These previously decoded neighboring blocks supply the samples the codec uses for block prediction, which is the basis for intra-frame prediction.



(a) Raster-scan

(b) Z-scan (based on [71])

Figure 2.6 – Block scanning order. Raster-scan (a) order follows a line-by-line fashion, from the leftmost to the rightmost block of each line. Meanwhile, z-scan (b) follows a depth-first traversal.

2.4 Frame Types and Group of Pictures

Current video coding standards typically classify frames into three types: (i) I-Frames, whose blocks are encoded purely by intra-prediction; (ii) P-Frames, whose blocks are encoded by either intra-prediction or inter-prediction using MCP with uni-prediction; and (iii) B-Frames, whose blocks are encoded by either intra-prediction or inter-prediction using MCP with uni- or bi-prediction. B-Frames offer the most flexibility for coded blocks. Uni-prediction in B-Frames may use forward or backward prediction from previously encoded frames.

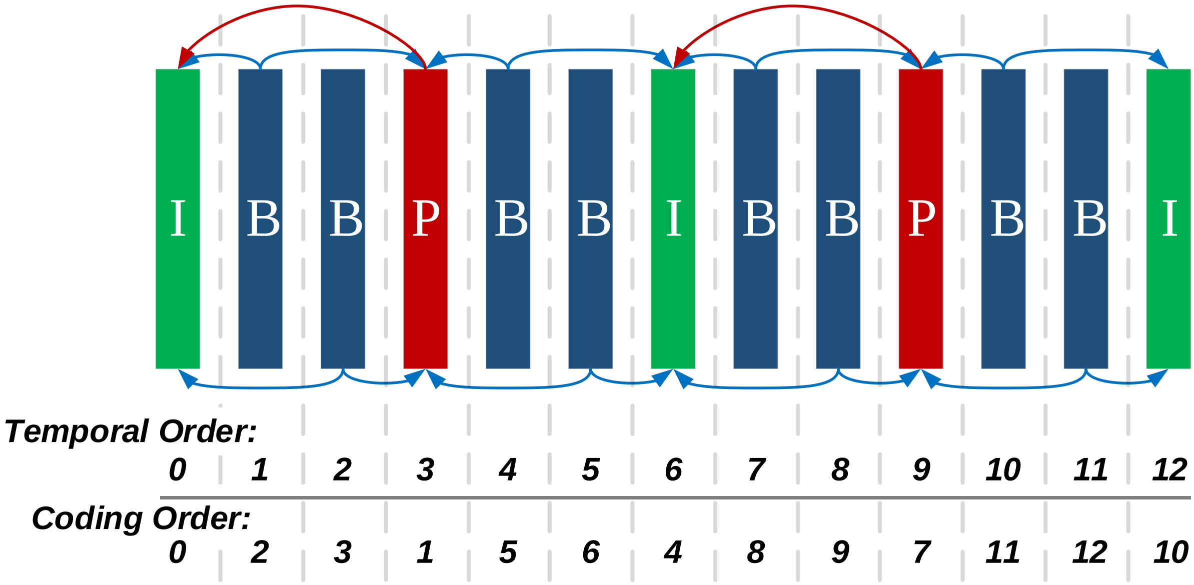


Figure 2.7 – An example of a typical GOP. The temporal order differs from the coding order. Except for the first frame which must always be an I-Frame, the remaining frames can be a combination of I-, P- or B-Frames, depending on the encoding configuration.

Coding standards may combine frame types in multiple ways. The typical Group of Pictures (GOP) defines a structure used throughout the encoding process for combining different frame types. While the first frame of any video is always an I-Frame, as there are no past encoded frames available for inter-prediction, later frames may be a combination of I-, P-

and B-Frames. Figure 2.7 illustrates an example of a GOP with I-Frames inserted at regular intervals. Depending on the encoding configuration, a GOP may solely consist of I-Frames, a single initial I-Frame followed by P- or B-Frames, or other combinations.

2.5 Intra-frame Prediction

Intra-frame prediction explores the statistical dependencies of block structures to compress video data [71]. Usually, when segmenting video frames, image elements are spanning multiple encoding blocks. Figure 2.8 shows elements such as the jockey's clothing spans across several blocks. Like the reins of the horse, other elements also span several blocks, creating a distinct contrast with the neck of the horse, which is not as homogenous as the previous example, but it is a distinguishable image feature. Intra-frame prediction relies on these continuous image elements to compress video frames. Instead of transmitting the entire block contents for reconstruction at the decoder, the encoder signals how to approximate block contents based on previously decoded blocks.

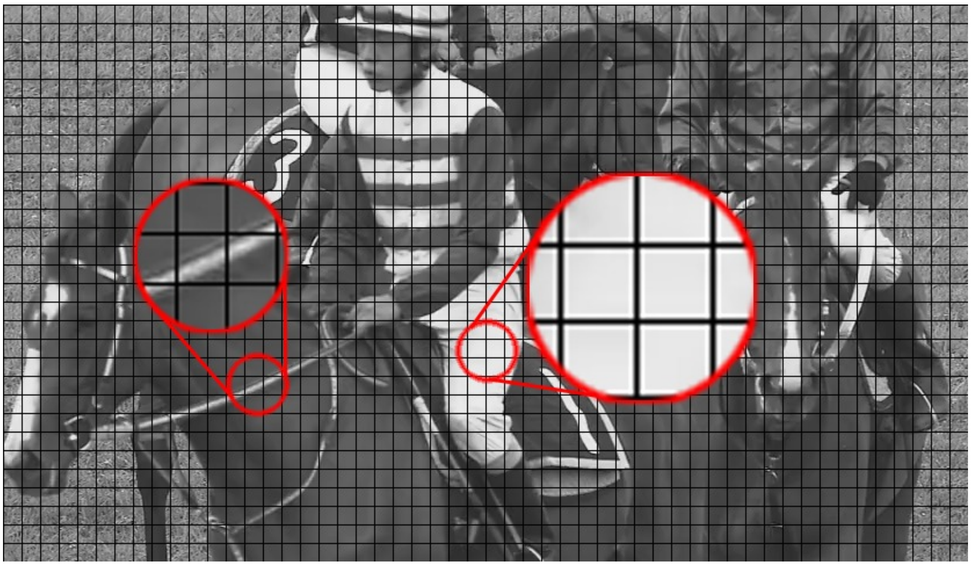


Figure 2.8 – Spatial redundancy in video frames. Intra-frame prediction relies on continuous image elements to extrapolate block contents during decoding due to the scanning order.

The general idea of intra-frame prediction is simple: the video encoder takes samples from neighboring blocks and, using a predefined set of intra-prediction modes, it attempts to find the mode which gives an approximation that best resembles the original content. One metric for measuring the approximation quality is the Sum of Absolute Differences (SAD). For an $n \times m$ block, Equation 2.1 demonstrates how to calculate the SAD, where i and j correspond to the rows and columns coordinates of the predicted block, respectively, and $o_{i,j}$ and $p_{i,j}$ to the original and the predicted samples, respectively. Lower values for SAD correspond to a better prediction of the block contents, while higher values correspond to a worse prediction.

$$\text{SAD} = \sum_{i=0}^{n} \sum_{j=0}^{m} |o_{i,j} - p_{i,j}| \tag{2.1}$$

Intra-frame prediction modes vary according to the coding standard. A numerical value identifies the prediction mode, known by both the encoder and the decoder, calculating sample values from a *reference array*. Thus, when encoding an intra-predicted block, the encoder must only signal to the decoder the numerical value associated with the prediction mode used to reconstruct the block instead of transmitting the entire block contents. The reference array corresponds to samples from the neighboring decoded blocks. Figure 2.9 illustrates some blocks computed by intra-frame prediction: the arrows are the direction used to calculate samples from the reference array. The Direct Component (DC) mode corresponds to a prediction mode employed for smooth or homogenous content, which averages the samples from the reference array to calculate a continuous value to all samples in the predicted block.

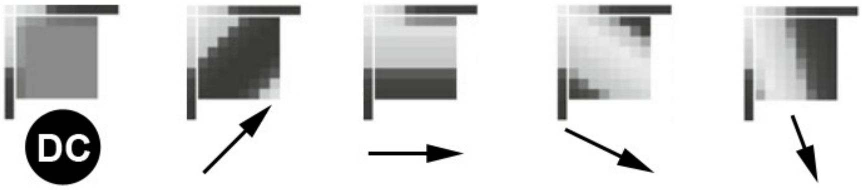


Figure 2.9 – Intra-frame prediction. The borders outside the predicted blocks correspond to the reference arrays, which are samples from neighboring decoded blocks. DC denotes a prediction mode that averages all samples from the surroundings, often used in homogenous texture regions (captured and edited from [61]).

Figure 2.10 illustrates the result of intra-frame prediction, which is the same frame shown in Figure 2.5. Although both images may look similar from a glance, there are errors, denoting intra-frame prediction limitations. While the homogenous texture regions like the jockey’s clothing show no visible errors, the high contrasting segments like the reins of the horse show distortions, compromising visual fidelity.

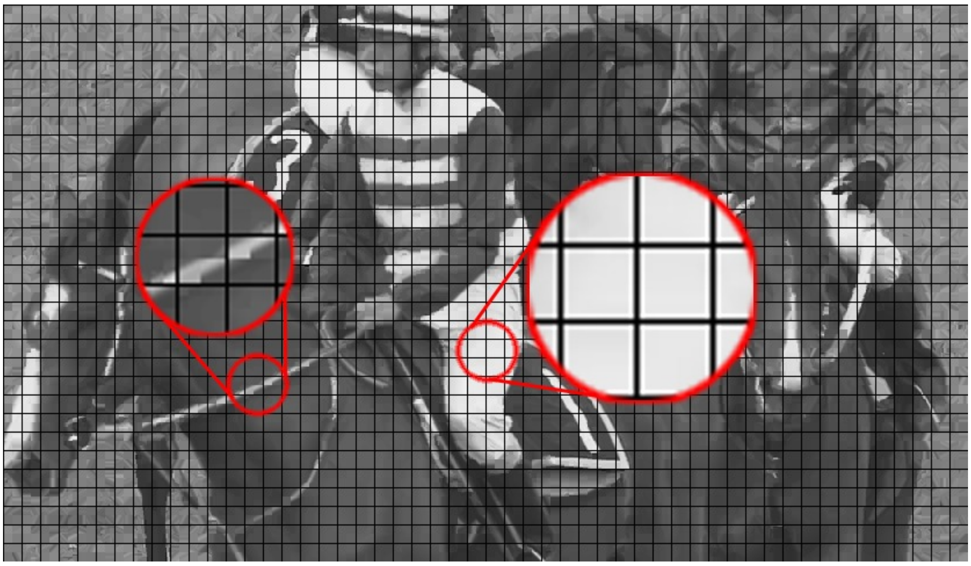


Figure 2.10 – Intra-frame predicted image. Visual artifacts can be seen in high contrasting regions due to errors in the prediction process.

2.6 Inter-frame Prediction

While intra-frame prediction handles the spatial redundancy of video frames, inter-frame prediction explores the temporal redundancy of the past encoded frames. The inclusion of inter-frame prediction fundamentally distinguishes video coding from still image coding [71]. Coding standards adopt Motion Compensated Prediction (MCP) to explore the temporal interdependencies among video frames. The idea, illustrated in Figure 2.11, consists of finding a motion vector for a predicted block from previously encoded frames. The motion vector indicates the displacement from the current predicted block to a region from previously encoded frames that corresponds to a well-matching area.

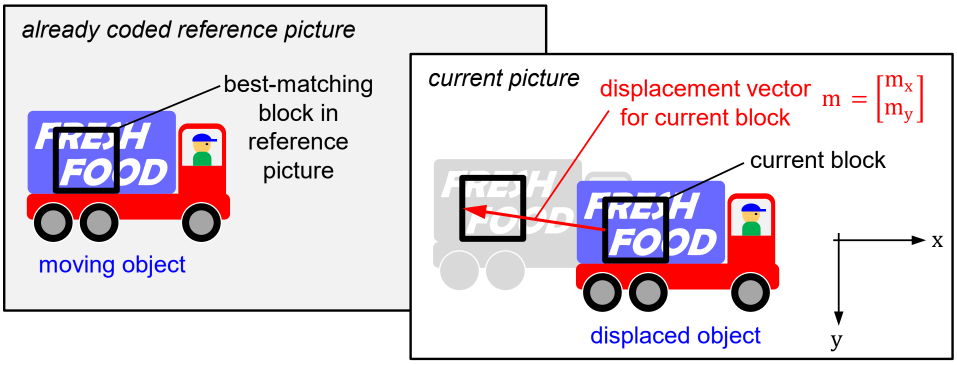

already coded reference picture
best-matching block in reference picture
FRESH FOOD
moving object
current picture
displacement vector for current block
$m = \begin{bmatrix} m_x \\ m_y \end{bmatrix}$
current block
x
y
displaced object

Figure 2.11 – Inter-frame prediction. Block motion estimation considers earlier frames, represented by a motion vector, which corresponds to the predicted block displacement to a past-encoded frame (captured from [71]).

The immediately preceding frame having the largest statistical redundancies for consecutive video frames does not mean it is the best source for MCP [71]. Due to differences in frames during acquisition like noise, short-time occlusions, among other effects, more distant frames may supply better sources for inter-frame prediction. Figure 2.12 illustrates how blocks from a frame may reference many already-encoded frames. Using more distant frames extends inter-frame prediction capabilities by offering better choices for MCP, improving encoding efficiency. Both the encoder and the decoder must keep a temporally ordered list of reference frames used for reconstructing a predicted frame, referred to as the reference picture list [71]. Any deviation between the reference picture list of the encoder and the decoder could result in different reconstructed frames.

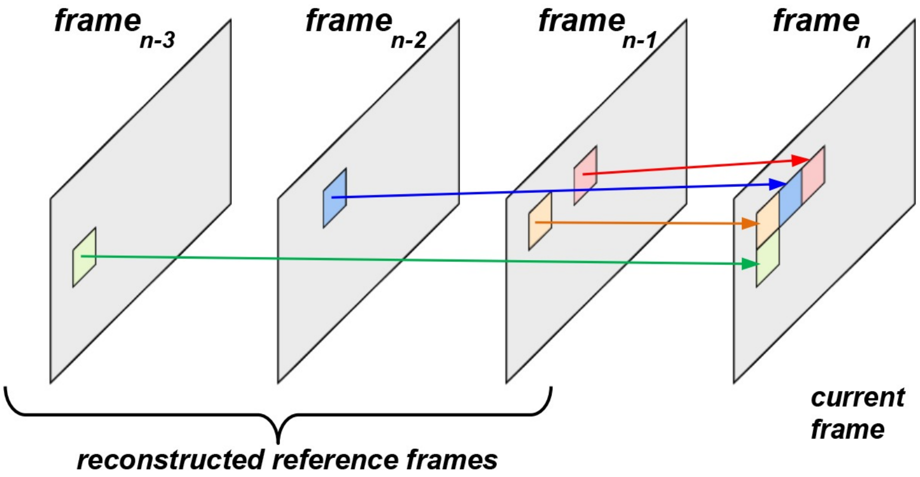


Figure 2.12 – MCP with multiple reference frames. A prediction block may reference other frames than the immediately preceding frame to achieve a better inter-frame prediction quality (captured and modified from [71]).

When multiple reference frames are available, new possibilities appear for MCP. Encoded blocks may employ *uni-prediction*, which is an MCP block that references a single already encoded frame. Alternatively, *bi-prediction* references two frames, interpolating the block contents between these two reference frames. For both uni- and bi-prediction, the reference frames may appear in any temporal order relative to the current frame, i.e., reference frames may come before or after the current frame. Given a group of sequential frames, encoding this group may occur sequentially or in an order different from its temporal order. Figure 2.13 illustrates the out-of-order encoding of four frames, where Frame 0 is the first encoded frame, followed by Frame 3, Frame 1, and Frame 2. This technique improves coding efficiency while requiring more memory to reconstruct the intermediate frames, implying in longer decoding delays for reconstructing the temporal order of the frames [71].

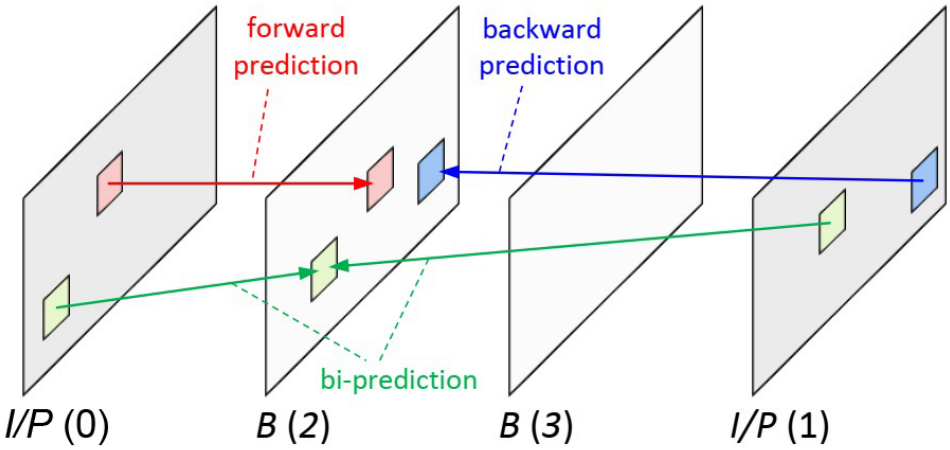


Figure 2.13 – I-Frames, P-Frames, and B-Frames. Using out-of-order encoding enables B-Frames to reference up to two frames in MCP, enhancing coding efficiency (captured from [71]).

2.7 Transformation and Quantization

Both intra- and inter-frame prediction represent only an approximation of block contents. The prediction process results in differences between the original and the predicted blocks, producing residuals that degrade the perceived quality for the decoded video unless they are transmitted along with the video bitstream.

Transform coding is a fundamental video coding technique [71]. The encoder transforms residual blocks from the spatial domain to the frequency domain using a linear transform function, typically the DCT. In the frequency domain, a set of sinusoidal functions represent the residual blocks, where a quantization function may irreversibly quantize the coefficients of the sinusoidal functions, since typically the quantization process results in sparse matrices. The decoder can reconstruct the quantized residual coefficients by applying a dequantization function. This process partially restores the residual coefficients [61], albeit with errors compared to the original values, especially for those coefficients that were zeroed by the forward quantization. Using an Inverse Discrete Cosine Transform (IDCT), the coefficients produce an approximation of the original residual blocks, which, combined with the block prediction, resembles the original block contents. Figure 2.14 illustrates the transform and quantization flows.

The prediction blocks correspond to a two-dimensional space of ordered samples. Each sample has a magnitude ranging from pure black to pure white, considering only the luminance samples of blocks, usually represented in 8-bit precision. The ordering of samples in the block corresponds to its coordinates in the 2-dimensional space, i.e., its row and column. The transformation of blocks from the spatial domain to the frequency domain consists of representing the block contents by a series of sinusoidal functions, each corresponding to a frequency sampled from the block. Figure 2.15 illustrates 64 2-dimensional sinusoidal

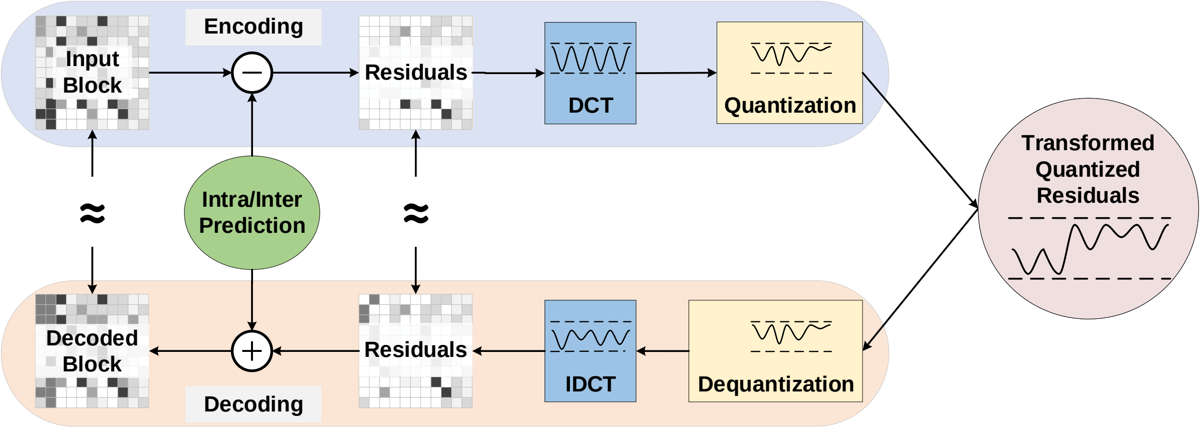


Figure 2.14 – Transformation and quantization. DCT domain transforms the residuals originated from block prediction, with the resulting transform coefficients being non-linearly quantized. The reverse process applies dequantization to the residual coefficients and rebuilds the residual block with the IDCT.

functions used with DCT for an 8 × 8 block. With their specific amplitude coefficients, the combination of each sinusoidal function can reconstruct any combination of 8 × 8 blocks.

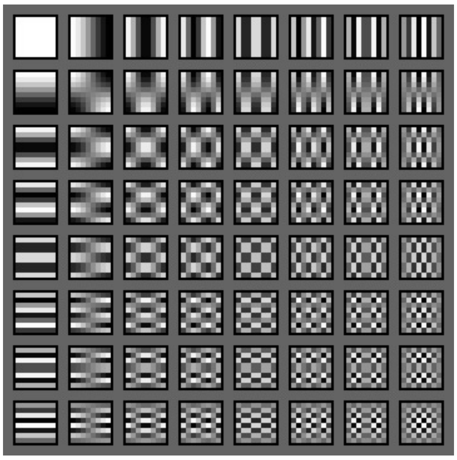
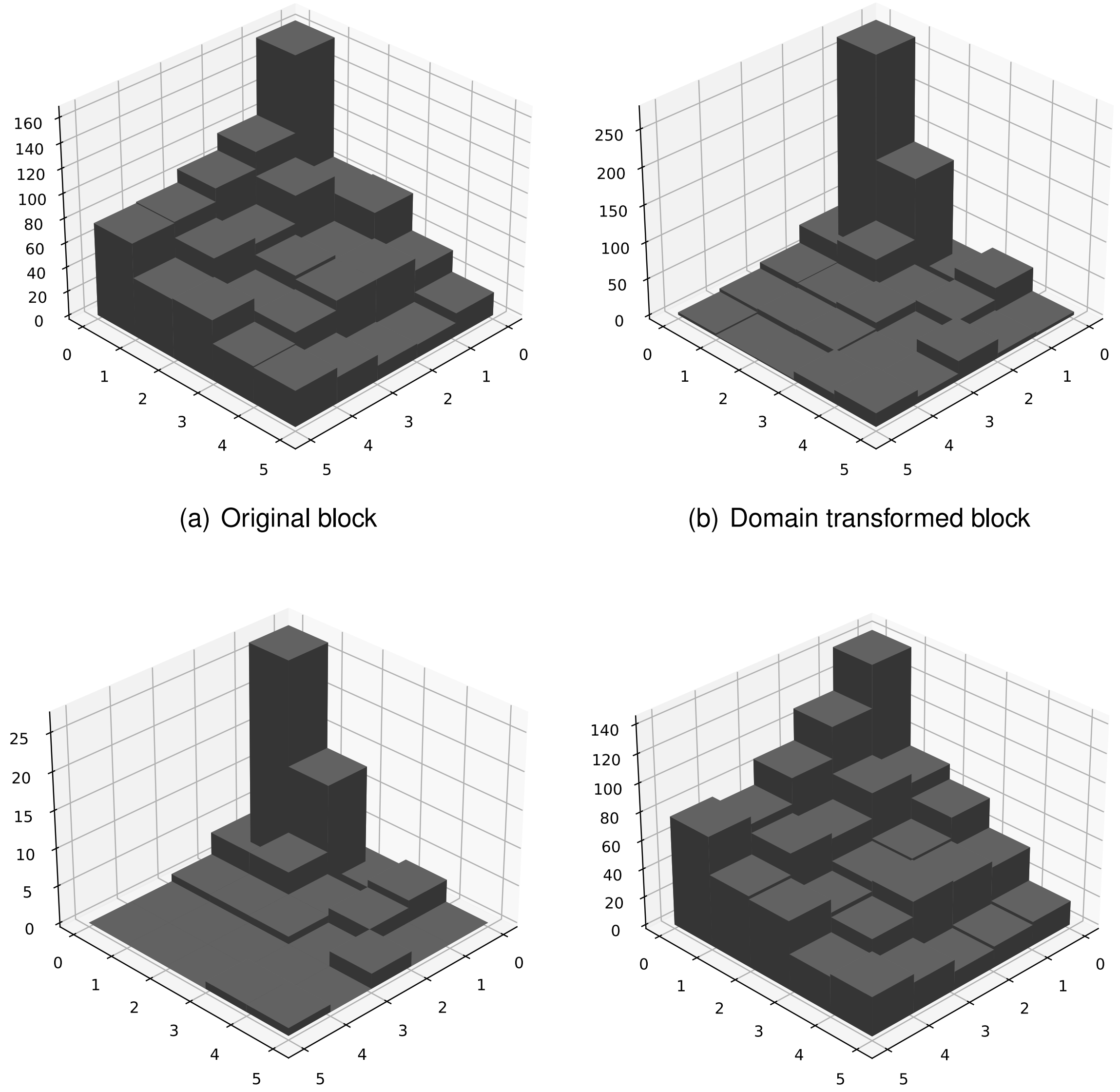


Figure 2.15 – DCT functions for an 8 × 8 block. In the upper left corner is the DC (constant) function. Darker shades are low amplitudes, while brighter shades are high amplitudes. Functions to the right represent higher horizontal frequencies, while functions to the bottom represent higher vertical frequencies (captured from [69]).

Unlike the spatial domain, where each sample is independent, altering a single coefficient of the sinusoidal functions, in the frequency domain, affects the entire image [12]. The higher frequency sinusoidal functions model the finer details in the prediction blocks, usually less perceivable by the human eye [69]; therefore, higher frequencies are generally subject to higher quantization than lower frequencies, without significant perceivable quality degradation.

Quantization truncates the coefficient values towards zero due to integer division, producing sparse matrices which improve entropy coding efficiency. Modern video coding standards adopt a Quantization Parameter (QP) starting at zero (no quantization), with gradual increases up to a limit specified by the video coding standard. A large QP results in a large quantization factor, subsequently increasing the number of zeroed coefficients; thus, improving the video compression ratio. However, increasing the quantization factor results in more truncated coefficients, reducing the quality of the decoded video.



(a) Original block

(b) Domain transformed block

(c) Quantized transformation

(d) Reconstructed block

Figure 2.16 – Block transformation and quantization. DCT maps the block contents to its frequency components. Later, quantization of the frequency coefficients reduces the amount of data necessary to represent the block contents, albeit at a precision loss.

Figure 2.16 exemplifies the quantization and dequantization process, where a particular quantization factor performs integer division of the transformed coefficient values and later multiplies the transformed coefficient values during dequantization. Given an arbitrary block in the spatial domain, shown in Figure 2.16(a), which is domain transformed using DCT, the result is a set of coefficients, as shown in Figure 2.16(b). These values of the coefficients correspond to weights associated with different sinusoidal functions, like the ones shown in Figure 2.16(c). Quantizing the coefficients reduces their values; e.g., using a quantization factor of 10, a coefficient of value 250 becomes 25, and a coefficient value of 9 becomes zero.

Observing Figure 2.16(b) and Figure 2.16(c), the quantization process results in zeroed coefficients (which will remain zero after dequantization) while other coefficients were truncated by the quantization factor: both situations imply in loss of information. However, even with quantization, the rebuilt transformed block can still achieve a good approximation of the original block contents, as illustrated in Figure 2.16(d).

2.8 Entropy Coding

The last stage of the encoding process consists of losslessly compressing information from the encoder flow, like block partitioning scheme, prediction modes, and the quantized residual coefficients. Entropy coding exploits statistical properties for compression such that the number of bits used to represent the data is logarithmically proportional to its occurrence probability [61]. For instance, few bits encode the often-used symbols, while a higher number of bits encodes the less often-used symbols. According to Shannon's information theory [56], for compressed data in binary form, the optimal average code length for a character with probability p is $log_2 p$.

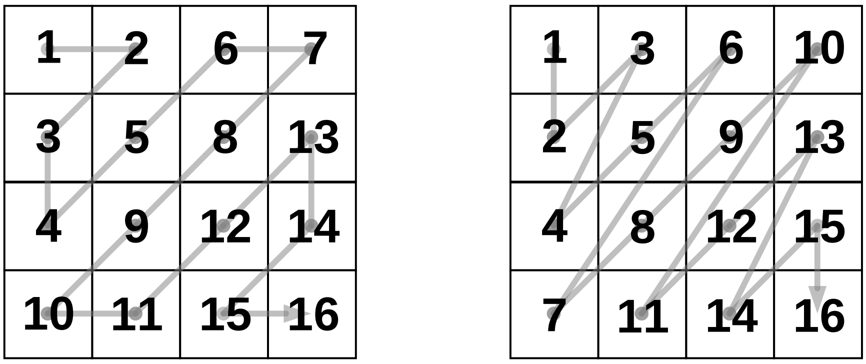
Compressing the high-level syntax elements originated from the encoding process requires the binarization of these elements. A binarization process defines a unique mapping of syntax elements to sequences of binary symbols [61]; e.g., a 101010 binary string could identify a syntax element that corresponds to the beginning of a prediction block. Video coding standards employ many binarization methods for signaling a numerical value N; some examples are *unary coding*, *exp-golomb*, and *fixed-length*. Unary is a simple encoding, using a zero-terminated binary string preceded by a series of ones for representing N. Exp-golomb uses code words with varying orders, expressed by the symbol k, to denote a prefix and a suffix. Prefixes have a length $L_N + 1$, while suffixes have length $L_N + k$, where $L_N = \lfloor log_2((N >> k) + 1) \rfloor$. Meanwhile, fixed-length encoding uses binary strings of length $\lceil log_2(N_{max} + 1) \rceil$, with the most significant bits signaled before the least significant bits. Table 2.2 shows these binarization models for $0 \leq N \leq 7$. When mapping high-level syntax elements, shorter binary values should map frequently-recurring symbols for reducing the bitstream size.

In residual blocks, coefficients have a probability of being non-zero depending on the employed quantization function. The high quantization used in high frequencies often results in zero-valued coefficients, while lower frequencies often result in non-zero values. Therefore, entropy encoders exploit these probabilities, playing a key role in video coding efficiency. Figure 2.17 shows an example of non-zero coefficient values. Like the DCT

Table 2.2 – Binarization models for unary, exp-golomb with k = 0 and k = 1, and fixed-length. Shorter binary strings should map the most probable high-level syntax elements to reduce bitstream size.

N	Unary	Exp-Golomb (k = 0)	Exp-Golomb (k = 1)	Fixed-Length (N_{max} = 7)
0	0	1	10	000
1	10	010	11	001
2	110	011	0100	010
3	1110	00100	0101	011
4	11110	00101	0110	100
5	111110	00110	0111	101
6	1111110	00111	001000	110
7	11111110	0001000	001001	111

functions shown in Figure 2.15, coefficients found in the upper left corner represent low-frequency components, while coefficients found in the bottom right corner represent high-frequency components. Figure 2.17(a) illustrates a scanning order that groups coefficients with similar probabilities. The zigzag scan employed in H.264 entropy encodes coefficients sequentially, starting from the DC component and progressing through the coefficients in a frequency increasing order. Another approach, shown in Figure 2.17(b), is the diagonal scan employed by HEVC. Due to the sequential grouping of coefficients with similar magnitudes, the entropy coder achieves better compression of these similar values.



(a) Zigzag scan

(b) Diagonal scan

Figure 2.17 – Coefficient scanning order. The H.264 coding standard uses zigzag scan (a), while HEVC employs diagonal scan (b).

Modern video coding standards like H.264, HEVC, VP9, and AV1, adopt binary arithmetic coding, enabling greater efficiency for compression of both high-level syntax elements and the quantized residuals. The arithmetic encoders employ context-adaptive models to compute symbol probabilities and statistical variations during the encoding process. Both the encoder and the decoder deterministically compute symbol probabilities at runtime to have the same probability information [61]. This approach enables an adaptation to symbol probabilities to achieve better compression ratios.

2.9 Performance Metrics

The assessment of video quality may be either subjective or objective. Subjective evaluation requires the use of people to score the perceptual video quality. It is very reliable; however, this approach is very time-consuming and impractical to evaluate large amounts of test sequences among different coding standards and configurations [67]. For this reason, quality measurement for video coding uses objective assessment most of the time. It consists of computational methods to score video quality, which are easily repeatable. The most used objective performance metrics in video coding standards are the Mean Squared Error (MSE) and the Peak Signal-to-Noise Ratio (PSNR). MSE is the mean of the squared differences between the gray-level of samples in two picture sequences I and I', for pictures of size X and Y [74], as shown in Equation 2.2. Meanwhile, Equation 2.3 defines PSNR in decibels, where m corresponds to the maximum value of a sample; in the case of videos using 8-bits per sample, $m = 255$.

$$\mathrm{MSE} = \frac{1}{XY}\sum_{x}\sum_{y}[I(f,x,y) - I'(f,x,y)]^2 \tag{2.2}$$

$$\mathrm{PSNR} = 10\log\frac{m^2}{MSE} \tag{2.3}$$

The MSE metric measures image differences, while PSNR measures image fidelity, i.e., how closely an image resembles a reference image [61]. In video coding, the reference image usually is the uncompressed video frame. MSE and PSNR also offer the important advantage of having their roots in computational problems, for which tools are readily available and fast to compute.

Researchers often compare and interpret the results using: (i) graphing the rate-distortion curves; and (ii) the Bjøntegaard Delta BitRate (BD-Rate). The graphical representation of the results shows the bitrate on the X-axis and the PSNR on the Y-axis. Figure 2.18 illustrates an example of a graphical comparison of two encoding implementations (either two encoders or the same encoder with different configurations, enhancements). From Figure 2.18, it is possible to analyze that Coding B achieves higher PSNR at a lower bitrate when compared to Coding A, thus having higher coding efficiency. For correlating the PSNR and the bitrate of encoded sequences, the BD-Rate is the reference performance assessment that compares the coding efficiency between video standards and coding configurations [3].

The BD-Rate expresses a rate difference, measuring the bitrate reduction of a codec implementation compared to another while keeping the same objective image quality [14].

The bitrate shows the number of bits processed per unit of time, usually expressed in Kilobits per second (Kbps) or Megabits per second (Mbps). Good coding efficiency translates into high PSNR with low Kbps, meaning high visual fidelity compared to the source and low data volume for storage or transmission. The computation of the BD-Rate considers an average percentage over at least four measurement points. Each measurement point corresponds to a quantization amount, which must be the same or at least equivalent for each evaluated codec in the comparison. In BD-Rate terms, negative values translate into improved coding efficiency of B over A, while positive values translate into decreased coding efficiency.

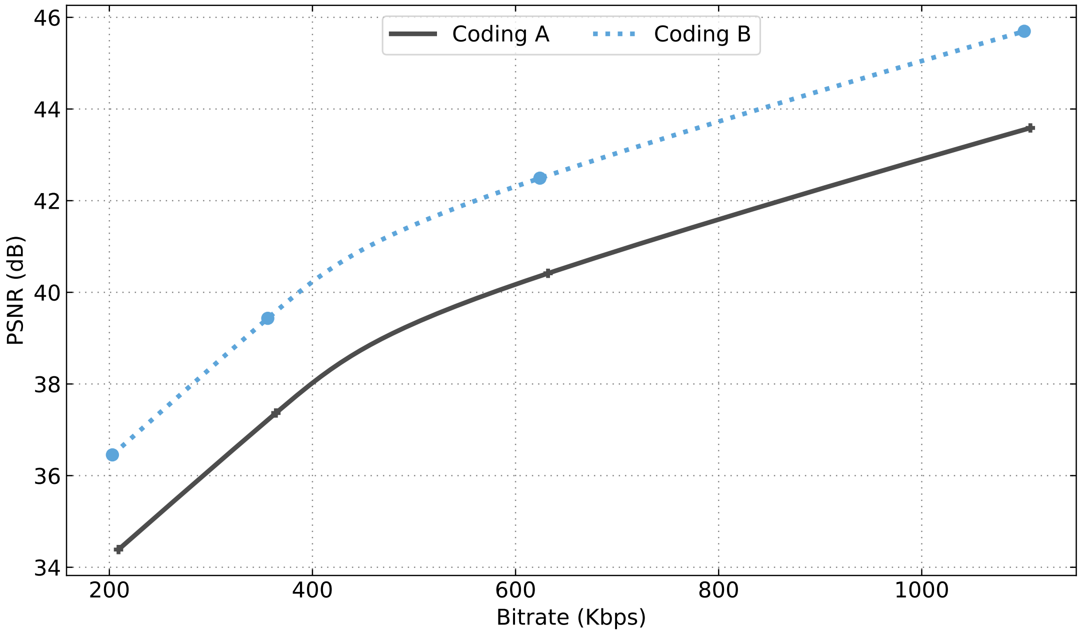


Figure 2.18 – Example of a rate-distortion curve for an encoded sequence using different coding configurations. Coding B shows improved efficiency over Coding A, as it offers a higher PSNR with a lower bitrate. Each dot in the curves denotes a quantization parameter: stronger quantization lowers bitrate at the cost of quality.

The test conditions for video coding standards typically employ a set of reference videos and coding configurations [48], e.g., using only I-Frames, different GOP configurations with P- and B-Frames, varying quantization parameters, among other coding configurations specific to each standard. The encoding of test sequences uses predefined coding configurations, followed by evaluating the PSNR and the bitrate of each encoded sequence. The varying quantization parameters generate the measurement points used for the comparison charts or calculating the BD-Rate.

2.10 Why HEVC?

Even though this Thesis adopts HEVC as the reference standard for the experimental evaluation, the proposed intra-frame prediction models can be used by other video coding standards, like the recent AV1 and VVC. The choice of HEVC for this Thesis considered aspects like the adoption of the coding standard, coding efficiency, and the availability of consolidated tools for the evaluations.

Currently, H.264, VP9, and HEVC are the three major worldwide standards used for distributing digital video. Most of the market still relies on H.264; a standard defined almost twenty years ago in 2003. The teams behind the H.26X and MPEG-X families of encoders combined efforts to create the H.264/MPEG-4 Part 10 AVC. Blu-ray media and internet videos adopted H.264 as the reference coding standard, which set it apart in terms of adoption. Most embedded devices to this day have a hardware implementation of the H.264 decoder, and most devices like mobile phones have a hardware implementation of the H.264 encoder for video recording. Major internet browsers also ship with H.264 decoding capabilities [32]. For these reasons, to this day, H.264 is still the most used video codec, but in recent years the group behind its development has shifted efforts towards the HEVC standard and, recently, to VVC, released in mid-2020.

To compete with the H.26X family of video coding standards, Google developed the VPX family, starting with VP8. Currently, the YouTube streaming platform uses VP9 to deliver most of its videos. The main advantage of the VPX family is a royalty-free business model, unlike the one used by H.26X. The royalty model used in HEVC is one of its most significant drawbacks for greater adoption [32]. Given the problematic situation for licensing HEVC, big tech companies like Google, Microsoft, Intel, Cisco, Mozilla, Netflix, and Amazon, have joined forces, forming the Alliance for Open Media (AOM) to develop royalty-free codecs, starting with AV1, which is the successor of VP9.

Since HEVC supersedes H.264, and disregarding its licensing problems, it is necessary to evaluate the current coding standards VP9, HEVC, and the newer AV1; the latter still in the early stages of adoption by tech companies at the time of this writing, but relevant nonetheless. In [67], the authors compared HEVC and VP9. Their results illustrate that HEVC and VP9 achieve near the same compression efficiency, with VP9 having a 2.24% BD-Rate reduction, on average. However, the coding efficiency depends on the type of test sequence used. In general, HEVC achieves better results with videos with higher motion, while VP9 performs better with videos with static or little motion. Authors of [23] compared H.264, HEVC, VP9; their tests show that HEVC obtained an average BD-Rate reduction of 32.8% and 38.4% compared to H.264 and VP9, respectively, concluding that VP9 achieved inferior performance compared to HEVC.

Using subjective metrics, the work in [1] compared HEVC, VP9, and AV1. Results showed HEVC and AV1 outperforming VP9, while for some video sequences, HEVC offered better results than AV1, and for other sequences, AV1 performed better than HEVC. AV1 and HEVC achieved average bitrate reductions of 28.1% and 27.3% compared to VP9, respectively, keeping similar quality. Subjective quality measurements placed AV1 and HEVC at 34.5% and 35.6% better than VP9, respectively. Meanwhile, the compression efficiency of

AV1 appeared as slightly worse than HEVC at 1.9%. In [22], the authors compared HEVC to H.264 and VP9, obtaining similar conclusions with HEVC averaging a 43.3% BD-Rate reduction over VP9 due to higher bitrate requirements for achieving similar visual quality.

Based on the studies mentioned above, and since HEVC supersedes H.264, HEVC offers better coding efficiency than VP9 and AV1. While the HEVC royalty model affects its adoption, HEVC is based on the H.26X family of codecs, for which consolidated tools and benchmarks exist, enabling an in-depth evaluation of new intra-frame prediction models. Despite the results of the recently released VVC showing considerable improvements over HEVC, the development of this Thesis was near its completion upon its release. Therefore, the studies presented in this Thesis for intra-frame prediction, despite being applicable for other coding standards, consider HEVC as the baseline codec for implementation and evaluation.

3. H.265 — HIGH EFFICIENCY VIDEO CODING

HEVC is the video coding standard defined by the Video Coding Experts Group (VCEG) of ITU-T[1] and the Moving Picture Experts Group (MPEG) of ISO/IEC[2], which are co-working in a partnership known as the Joint-Collaborative Team for Video Coding (JCT-VC) [60]. The first edition of the HEVC standard appeared in April 2013, with extensions released in later years to support 3D coding, referred to as 3D-HEVC [62], and Screen Content Coding (SCC) [65].

Like the past video coding standards from ITU-T and ISO/IEC, only the bitstream structure and syntax elements follow a standard and the mapping of bitstream elements for the generation of the decoded video. The mapping ensures any decoder implementation of HEVC can produce the same output given a bitstream that conforms to the standard. An advantage of this model is the flexibility for optimizing encoder implementations to many requirements, e.g., implementation cost, compression efficiency, energy consumption, and time to market.

HEVC uses the same video coding techniques discussed in the previous chapters. The reference encoder segments input frames according to a quadtree structure, using intra- and inter-frame prediction with transform coding. A decoder module inside the encoder reconstructs the encoded frame segments to predict subsequent blocks accurately. This last step is necessary to emulate decoder conditions so that intra- and inter-frame prediction uses the same references available at decoding time.

The standardization effort includes detailed text material and a reference software [48] to facilitate its dissemination. The HEVC design committee used the reference software during the standard creation, and it is openly available for the research community and industry alike. The following sections present a brief explanation of the HEVC standard according to its bitstream structure, emphasizing its intra-frame prediction, which is the focus of this Thesis.

3.1 HEVC Block Structures and Frame Partitioning

HEVC divides frames into square-shaped, disjoint units of the same size, named the Coding Tree Unit (CTU), serving as the root subdivision units of a quadtree structure and the primary processing unit of HEVC. The CTU is the combination of three Coding Tree Block (CTB) elements: one for the luminance channel, and two for each chrominance channel. The number of CTBs in video frames is the same for luminance and chrominance, regardless of

[1] ITU Telecommunications Standardization Sector

[2] International Organization for Standardization and the International Electrotechnical Commission

the chrominance subsampling, as chrominance CTBs adopt the subdivision of the luminance CTB. The CTU and the CTB dimensions are relative to the number of luminance samples, i.e., a $2^N \times 2^N$ picture area. The bitstream signals the parameter N with values of $N = 4, 5$, and 6 for CTU sizes of 16×16, 32×32, and 64×64 luminance samples, respectively. Using a large CTU is preferable due to the lower amount of collateral data [61]. Each subdivision of the CTU is known as a Coding Unit (CU), representing the processing units of video coding with a corresponding intra- or inter-prediction mode. Like in the CTU, each CU consists of three Coding Block (CB) elements, for which the encoder decides between using intra- or inter-frame prediction.

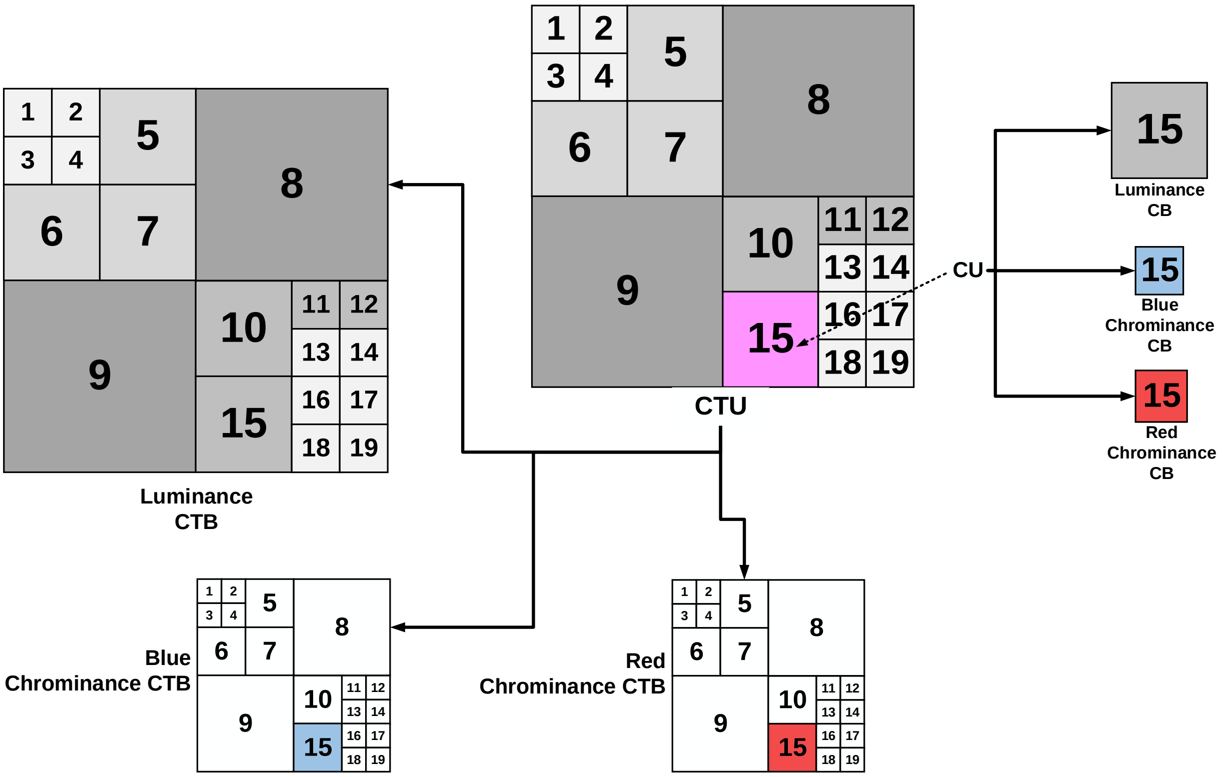
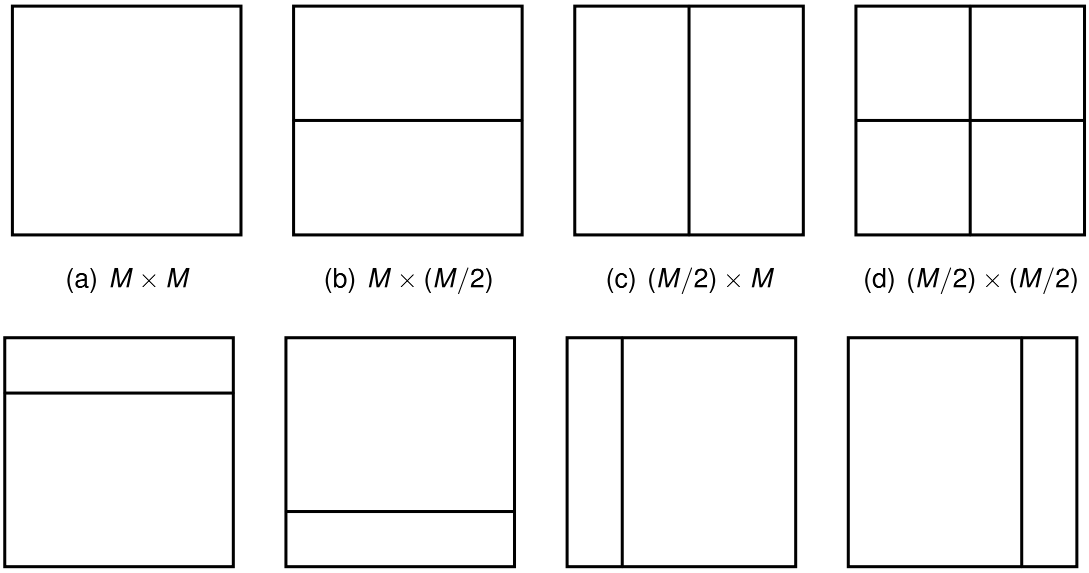


Figure 3.1 – HEVC partitioning example with 4:2:0 chrominance subsampling. The chrominance CTBs follow the same partitioning of the luminance CTB, down scaling if necessary. CUs in the CTU follow a z-scan encoding order.

Figure 3.1 exemplifies the HEVC partitioning. The luminance CTB defines the subdivision of all chrominance channels in the CTU. With 4:2:0 chrominance subsampling, as shown in the example, the chrominance CTBs and CBs have a quarter the size of their luminance counterparts. At the CTU level, a syntax element `split_cu_flag` specifies whether a CTU of size N has subdivisions, or if it corresponds to a whole CU. When `split_cu_flag` is true, the CTU partitioning consists of four equally sized CUs of dimensions $2^{(N-1)} \times 2^{(N-1)}$. The `split_cu_flag` may recursively appear at each CU, pointing to further subdivisions. The encoding order of CUs inside a CTU follows a depth-first order, also known as z-scan. Figure 3.1 also illustrates the encoding order as the numerical value of each CU. This encoding order ensures that, except for CUs found at the top or left frame boundaries, samples from CUs left and above to the CU are already encoded and available for intra-frame prediction.

The CU can be further partitioned into the Transform Unit (TU) for transform coding the prediction residuals. It follows the same CTU quadtree structure, denoted as the Residual Quadtree (RQT), allowing its recursive subdivision. Each color channel in the TU is represented by a Transform Block (TB), where each TB is associated to a CB. The reason for allowing the subdivision of CBs into TBs is the better resolution of smaller TBs in the spatial domain, at the cost of more syntax elements required to signal the TBs; meanwhile, larger TBs provide better resolution in the frequency domain [61]. Either way, it is up to the encoder to decide the best trade-off between the two. Like other block structures, chrominance TBs follow the same subdivision structure of the luminance TB.

Following the same model of the other data structures mentioned so far, a Prediction Unit (PU) combines the information of a luminance Prediction Block (PB) and two chrominance PBs. PU subdivision for intra-frame prediction supports only $M \times M$ or $(M/2) \times (M/2)$ modes, illustrated in Figure 3.2. The PB signals CB prediction mode; if the CB uses intra-frame prediction, the PB signals which of the 35 available modes to use for the luminance CB. Meanwhile, the chrominance PBs set a single mode for both chrominance channels, assuming one of five values, one of which is a reserved intra-prediction mode for chrominance, named `intra_derived`; this mode signals that the chrominance PBs use the same intra-prediction mode of the corresponding luminance PB.



(a) $M \times M$ (b) $M \times (M/2)$ (c) $(M/2) \times M$ (d) $(M/2) \times (M/2)$

(e) $M \times (M/4)$ (f) $M \times (3M/4)$ (g) $(M/4) \times M$ (h) $(3M/4) \times M$

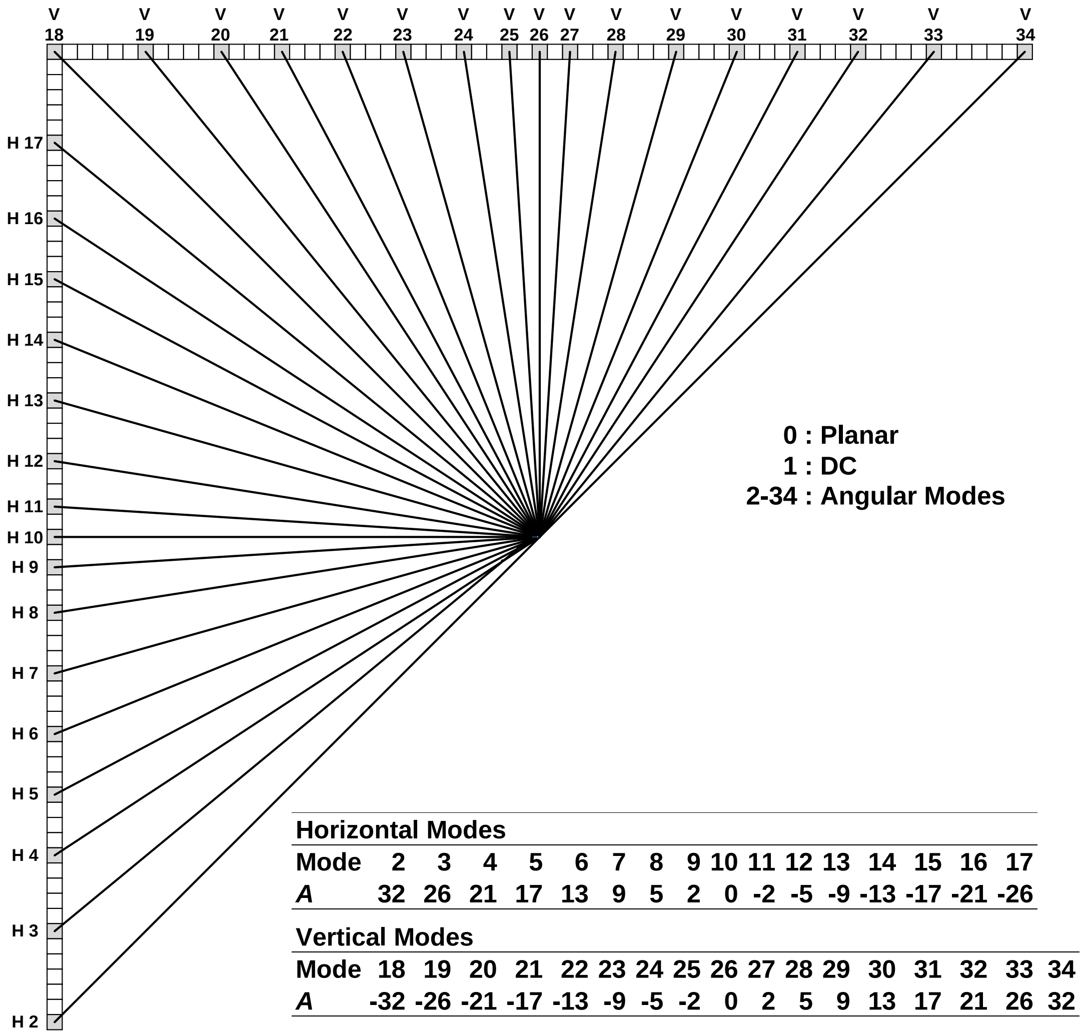
Figure 3.2 – Supported CU partitioning modes into one, two, or four PUs. These subdivisions depend on the prediction mode (intra- or inter-frame) and the CU size.

When using inter-frame prediction, the encoder may also split the luminance and chrominance CBs of a CU into other PBs. In inter-frame prediction, PBs encode the motion parameters of blocks used in P- and B-Frames, indicating if they use one or two reference frames to predict the block contents, as well as the reference frame index in the GOP and the motion vectors. Since there are no significant benefits for individually encoding motion parameters for luminance and chrominance PBs, the chrominance PB uses the same motion parameter of the luminance PB [61] so that the PU signals a single set of motion parameters in the bitstream. The partitioning of the CU in PUs may occur in eight forms, as Figure 3.2 illustrates. HEVC limits the $(M/2) \times (M/2)$ partitioning to a minimum CU size of 8×8 samples. If there is no CU subdivision, then the partitioning uses mode $M \times M$. PUs may also use symmetrical division with $M \times (M/2)$ or $(M/2) \times M$ partitioning, or asymmetrical division as shown in Figure 3.2(e) to Figure 3.2(h); the latter only available for CUs larger than 8×8 and for inter-frame prediction.

3.2 HEVC Intra-frame Prediction

The HEVC intra-frame prediction exploits the inherent spatial redundancy in frames to compress video data, using the high correlation of nearby samples and blocks [30, 59]. HEVC employs a range of intra-prediction models to predict diverse kinds of content efficiently. These models can be divided into two classes [29]: (i) Angular prediction, designed to model directional features typically present in high contrasting image regions; and (ii) Planar and DC prediction, designed to model smooth image content. The HEVC intra-frame predictor uses reference samples from adjacent (above and left) reconstructed blocks to approximate the predicted block contents. Figure 3.3 shows the 33 directions of HEVC angular intra-frame prediction, along with their corresponding angular parameters A. Although angular prediction is suitable for approximating blocks with high contrasting edges, it may result in blockiness or visible contouring in smooth image regions. Planar prediction overcomes these issues by creating a prediction surface without discontinuities at the block boundaries. In contrast, DC prediction is efficient for image regions with homogeneous textures, populating the predicted block with an average constant value computed from the reference samples.

To approximate block contents, intra-frame prediction employs a reference sample array obtained from neighboring reconstructed blocks. This array has $2n+1$ samples in the range $[-1, 2n-1]$, where n is the height or the width of the predicted block size, expressed as $n \times n$. Equations 3.1 and 3.2 obtain the reference array values for horizontal and vertical modes, respectively, where $p[x][y]$ corresponds to sample values of previously encoded blocks, indexed relative to the predicted block coordinates. Figure 3.4 presents a visual representation of this array using samples from the left or above the predicted block, where blocks in shaded gray represent the reference samples, the white blocks represent the predicted block.



Horizontal Modes

Mode	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A	32	26	21	17	13	9	5	2	0	-2	-5	-9	-13	-17	-21	-26

Vertical Modes

Mode	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
A	-32	-26	-21	-17	-13	-9	-5	-2	0	2	5	9	13	17	21	26	32

Figure 3.3 – HEVC intra-frame prediction modes and their associated angular parameters; 2-17 and 18-34 are the horizontal (H) and vertical (V) modes, respectively.

$$ref[x] = p[-1 + x][-1], (x \geq 0) \tag{3.1}$$

$$ref[y] = p[-1][-1 + y], (y \geq 0) \tag{3.2}$$

When samples are not available from neighboring blocks, HEVC defines how to populate the reference array as follows:

- In the unique case of no available samples, e.g., the first encoded block in a frame, all samples in the reference array assume the nominal average value according to the precision used by samples, e.g., with 8 bits precision, sample values are in the range $[0, 256)$; thus, the average value is 128;

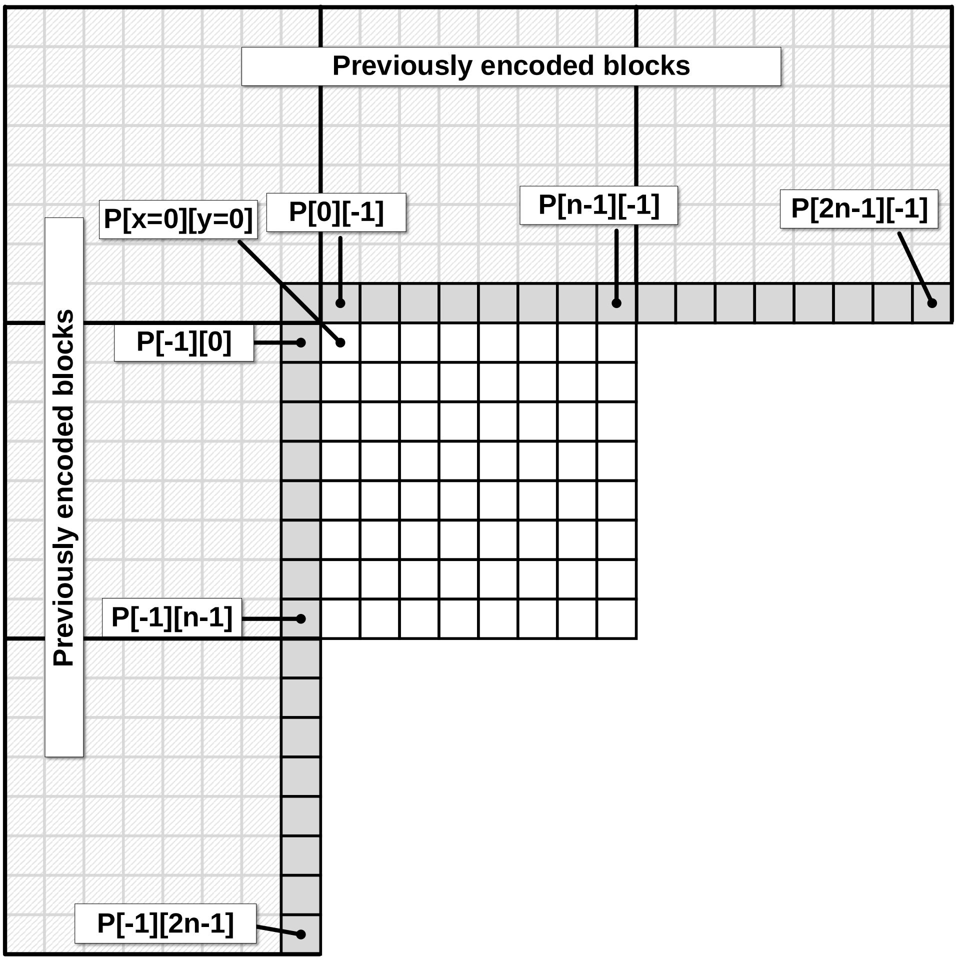


Figure 3.4 – Reference sample arrays for intra-frame prediction. Blocks in shaded gray represent the reference samples, with the predicted block (8 × 8 in this case) represented by the white blocks.

- When $p[-1][2n-1]$ is not available, it is substituted by the first available sample, scanning in counter-clockwise order, i.e., $p[-1][2n-2], \ldots, p[-1][-1], \ldots, p[0][-1], \ldots, p[2n-1][-1]$;

- All unavailable reference samples at the left from $p[-1][y]$, where $y = 2n - 2$, to, -1, are substituted by the immediate sample below, at $p[-1][y+1]$; and

- All unavailable reference samples above, from $p[x][-1]$, where $x = 0$ to $2n - 1$, are substituted by the immediate sample from the left, at $p[x-1][-1]$.

For prediction modes with negative angular parameters, i.e., intra-prediction modes 11 to 25, the reference array range is $[-n+1, n-1]$, where a process called *reference array extension* computes values in the range $[-n+1, -1]$. Equations 3.3 and 3.4 define a projection from the left and above reference arrays used to obtain these values for vertical and horizontal modes, respectively. B is the inverse angle parameter, given by $B = 2^{13}/A$, remembering that A corresponds to the angular parameter value. Figure 3.5 illustrates this procedure for a given prediction direction, where reference array extension copies samples from the left reference array to the negative indexes of the above reference array.

$$ref[x] = p[-1][-1 + ((x \times B \times 128) >> 8)], (x < 0) \tag{3.3}$$

$$ref[y] = p[-1 + ((y \times B \times 128) >> 8)][-1], (y < 0) \tag{3.4}$$

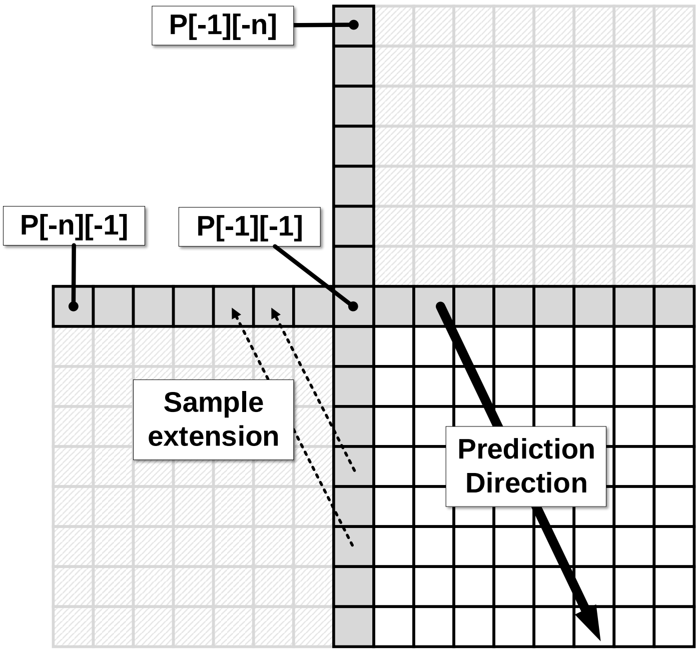


Figure 3.5 – Reference array extension. The procedure copies samples from the left reference array to the negative indexes of the above reference array, according to the prediction direction.

HEVC uses Equations 3.5-3.10 to compute the predicted sample values in both vertical and horizontal angular prediction modes [61]. Equations 3.5 and 3.8 compute the sample values *p* in a predicted block for vertical and horizontal modes, respectively; the equations perform linear interpolation using the two closest reference samples in the prediction direction. Equations 3.6 and 3.9 show the projected integer displacement *i* depends on the row *y* for vertical modes, column *x* for horizontal modes, and angular parameter *A* corresponding to the prediction mode. Finally, the fractional part of the projected displacement *f* (Equations 3.7 and 3.10) depends on the row *y* for vertical modes, the column *x* for horizontal modes, and the angular parameter *A*.

$$p[x][y] = ((32 - f) \times ref[x + i + 1] + f \times ref[x + i + 2] + 16) >> 5 \tag{3.5}$$

$$i = ((y + 1) \times A) >> 5, \text{for vertical modes} \tag{3.6}$$

$$f = ((y + 1) \times A)\&31, \text{ for vertical modes} \tag{3.7}$$

$$p[x][y] = ((32 - f) \times ref[y + i + 1] + f \times ref[y + i + 2] + 16) >> 5 \tag{3.8}$$

$$i = ((x + 1) \times A) >> 5, \text{ for horizontal modes} \tag{3.9}$$

$$f = ((x + 1) \times A)\&31, \text{ for horizontal modes} \tag{3.10}$$

While angular prediction calculates sample values from a single reference array, i.e., the left reference array for horizontal modes or the above reference array for vertical modes, Planar prediction uses both reference arrays. Equation 3.11 demonstrates how to compute sample values $p[x][y]$, which corresponds to averaging a vertical and a horizontal component, given by p_v in Equation 3.12, and p_h in Equation 3.13, respectively. The vertical component calculates sample values from the above reference array, while the horizontal component calculates sample values from the left reference array. Meanwhile, DC prediction averages the sample values from the above and left reference arrays, applying a single value to all predicted block samples.

$$p[x][y] = (p_h[x][y] + p_v[x][y] + n) >> (\log_2(n) + 1) \tag{3.11}$$

$$p_v[x][y] = (n - 1 - y) \times p[x][-1] + (y + 1) \times p[-1][n] \tag{3.12}$$

$$p_h[x][y] = (n - 1 - x) \times p[-1][y] + (x + 1) \times p[n][-1] \tag{3.13}$$

The HEVC encoder recursively evaluates all depths inside the CTU to select the best localized intra-frame prediction mode, where the block size ranges from 64 × 64 down to 4 × 4. For each block size, the reference HEVC software encoder [48] evaluates all 35 intra-prediction modes using Rough Mode Decision (RMD) [78], which employs the Sum of Absolute Transformed Differences (SATD) to estimate the encoding efficiency. SATD measures prediction quality by evaluating only the residuals of prediction blocks, as described by Equation 3.14, where $t[x][y]$ corresponds to the transform coefficient values for the residuals of the predicted block using the Hadamard transform [61].

$$\text{SATD} = \sum_{x=0}^{n} \sum_{y=0}^{n} |t[x][y]| \tag{3.14}$$

The most efficient prediction modes, according to the lowest SATD values, are then inserted into a Rate-Distortion list (RD-list) to be later evaluated by the encoder for its entire Rate-Distortion cost (RD-cost) [59]. RD-cost measures the bitrate requirements for a predicted block, considering its syntax elements, the prediction mode, and entropy coded transform-quantized residuals. Since computing the RD-cost for all prediction modes for all CTU subdivisions is a complex procedure, the encoder inserts the best candidate modes, according to the SATD, in the RD-list.

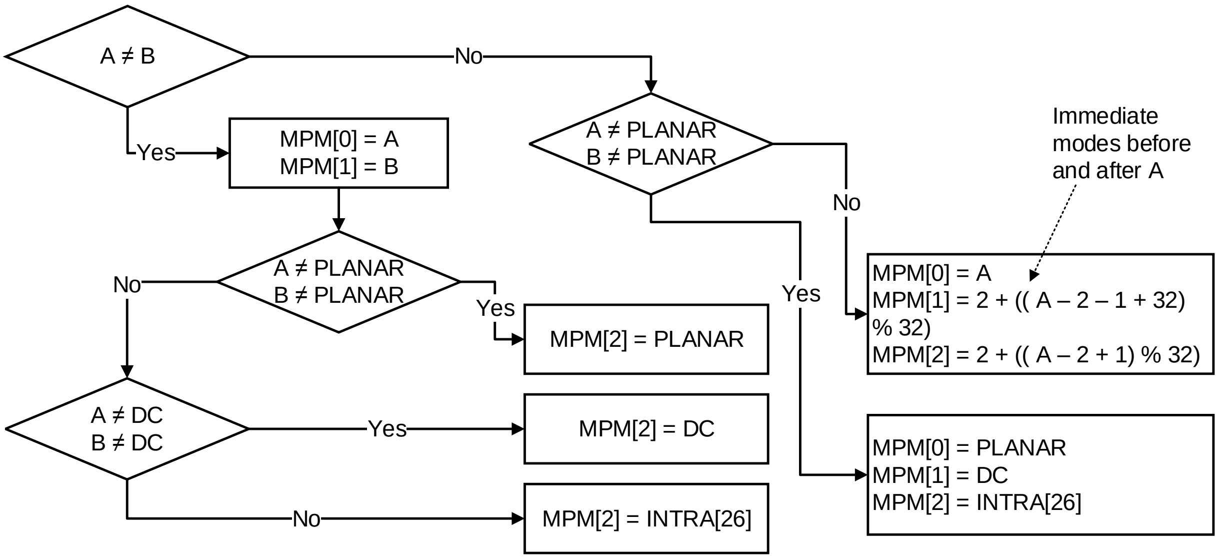


Figure 3.6 – HEVC MPM decision flow. Intra-prediction modes from neighboring blocks help to find the best prediction modes for the current predicted block. (based on [61]).

It is worth mentioning one important encoder optimization in HEVC, referred to as the Most Probable Mode (MPM). Along the prediction modes chosen according to the SATD, the RD-list also contains three MPM, derived from the prediction modes chosen for the two already encoded neighboring PUs, above and left of the current predicted block. Figure 3.6 illustrates the logic for choosing the MPMs, where *A* and *B* correspond to the prediction modes of the PU left and above of the predicted block, respectively. Since there is a strong correlation of the intra-prediction mode with its neighboring PUs, MPM can improve coding efficiency by reducing the bitstream size. When a PU uses a prediction mode from the MPM, the bitstream only must signal which of the three possible MPMs to use, instead of which of the 35 available modes for the intra-frame prediction.

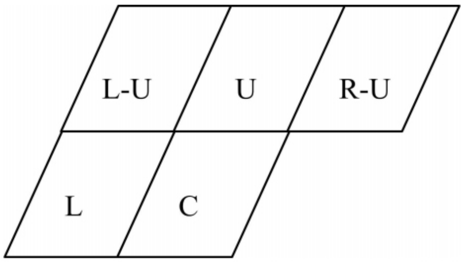
4. RELATED WORK

Section 4.1 discusses some works about the high computational complexity introduced by the HEVC intra-frame prediction, providing ideas which can be used for optimizing the models proposed on this Thesis. Meanwhile, section 4.2 discusses the state-of-the-art of intra-frame prediction, which is the focus of this Thesis and an essential subject for improving the current and future video coding standards.

Works in state-of-the-art addressing the intra-frame prediction problem [9, 20, 36, 38] adopt different techniques to improve coding efficiency, which is the focus of this Thesis. Section 4.2 discusses the state-of-the-art of this subject, which is essential for improving the current and future video coding standards.

4.1 HEVC Encoder Complexity Reduction

In [58], the authors propose a fast algorithm for reducing intra-frame prediction encoder complexity. Their work employs two approaches for reducing the amount of computation required for determining the CU depth and intra-prediction mode. Measurements show that the CTU depth is often small in homogenous frame regions, meaning it is unnecessary to subdivide a homogenous region into deeper CTUs to improve coding efficiency. Meanwhile, frame regions with complex textures often require deeper CTU partitioning to improve rate-distortion. The current CTU depth level $D = \sum_{i=0}^{4} \omega_i \alpha_i$ (ω is the value of the depth level and α is the weight of the neighboring CTU for the decision process) is predictable based on the spatially nearby CTUs since natural images neighboring CTUs, illustrated in Figure 4.1, usually hold similar textures.



(a) Neighboring CTUs used for depth estimation

	Left CU	Above CU	Left-Above CU	Right-Above CU
α_i	0.3	0.3	0.2	0.3

(b) Weight factors applied for neighboring CTUs

Figure 4.1 – Fast CTU decision algorithm. The neighboring CTUs are: C (current), L (left), U (above), L-U (left above) and R-U (right above). Each neighboring CTU weights differently in the depth decision process (captured from [58]).

The authors refer to the CTU depth levels as treeblocks, classifying each level into one of four categories. Measurements on the probabilities of CTU depth levels for test videos have demonstrated that treeblock category *I* has 89% of its CTUs with a maximum depth of 1, treeblock category *III* has 97% of its CTUs with depths ranging from 1 to 3, while the other treeblock categories have their own probabilities. Thus, the intra-frame prediction may only evaluate block partitions in the most probable depths instead of testing all CTU depths.

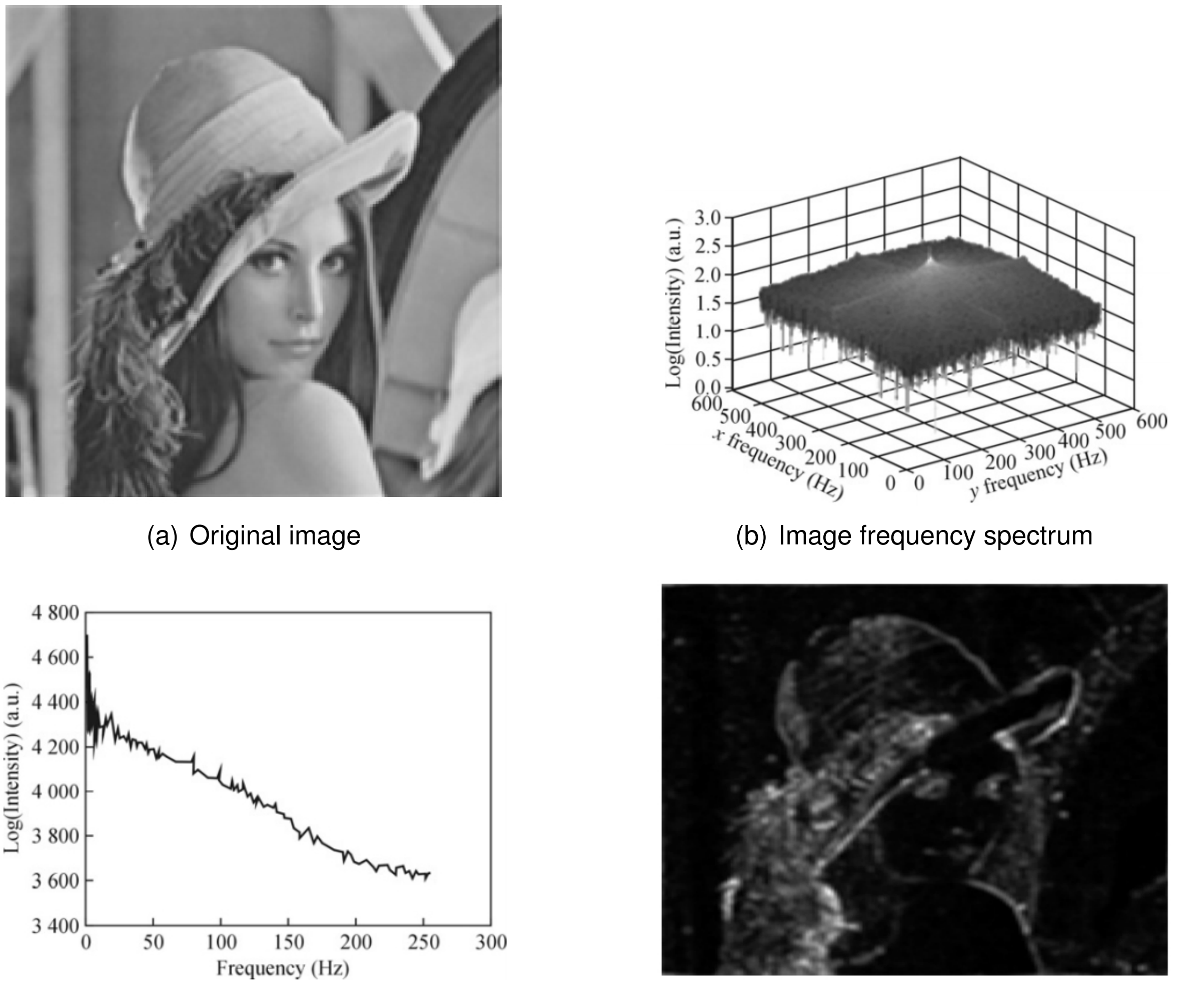
The second part of the algorithm proposed in [58] pertains to the intra-prediction mode decision. Statistical evaluations demonstrated that the first element of the RD-list computed with RMD contains the optimal prediction mode in 59% of the cases; MPMs are the optimal modes of 34% of the cases, and the CU of depth $D - 1$ of the current predicted block has the optimal intra-prediction mode in 23% of the cases. Thus, RD-cost computation only uses these modes, which dramatically reduces the encoding time.

Experimental results show an average encoding time reduction of 21%, with a maximum of 36% and a minimum of 13% for the tested video sequences. Meanwhile, BD-Rate increases by 1.74% on average, with a maximum of 3.04% and a minimum of 0.97%. The measured PSNR stays approximately the same as without their optimizations, translating into a reasonable trade-off given the significant encoding complexity reduction.

The authors of [35] make the same observations as [58], i.e., the encoding of homogenous frame regions use large CUs while complex texture regions use small CUs. However, their proposal measures the texture complexity of CUs according to the standard deviation of its samples. A correlation of block standard deviation and CTU subdivision indicates the optimal depths for coding the image region, skipping non-optimal depths, thus reducing encoding complexity. Additionally, their work also employs statistical analysis of CU samples to determine the probable prediction direction. The block content classification considers eight directional groups: vertical, horizontal, and six groupings of diagonal directions. Depending on the grouping which best describes the block direction, the encoder only has a subset of the 33 possible angular directions of HEVC to evaluate. Combined with the optimized CU size algorithm, [35] achieved a substantial average encoding time reduction of 56%, with a BD-Rate penalty averaging 1%.

The work [15] focuses exclusively on a fast CU size selection. Their approach contemplates regions of interest on images, according to the human visual perception. A region of interest considers content that attracts the human perception in the images, defined by salient regions. Meanwhile, the authors treat non-salient regions as background information, perceived with lesser priority by the HVS. Thus, by extracting salient regions during encoding, it is possible to decide the CTU partitioning scheme for image regions without the typical exhaustive recursive search of HEVC.

The image block is transformed into the frequency domain, obtaining the salient map, and then mapped with a natural logarithmic function to extract the most significant amplitudes of the frequencies. The resulting logarithmic amplitude spectra, once inverse transformed back to the spatial domain, reveals the significant image edges the HVS perceives the most. Figure 4.2 illustrates this idea for a whole picture.



(a) Original image

(b) Image frequency spectrum

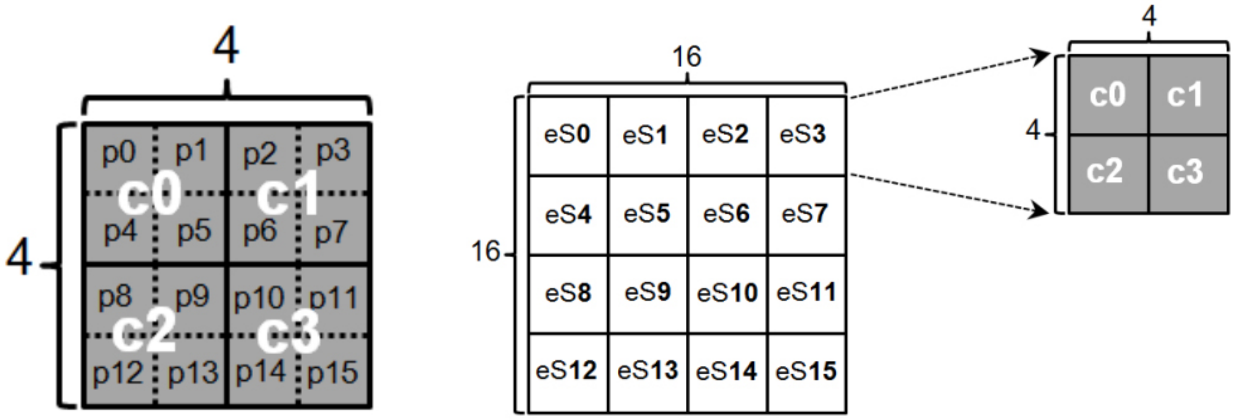
(c) Frequency distribution

(d) Salient map

Figure 4.2 – Salient map extraction. Mapping of the relevant frequencies takes place after the original image undergoes a frequency domain transformation (captured from [15]).

The salient map allows to estimate the entropy level of the image block. Homogenous image regions translate to low entropy, while complex texture image regions translate to high entropy. According to an entropy threshold, the encoder evaluates the entropy levels for all CTU depths, from 64 × 64 down to 4 × 4. If the entropy variance from $Depth_n$ to $Depth_{n-1}$ is greater than the threshold, the encoder splits the CTU according to $Depth_n$; otherwise, the encoder splits the CTU according to $Depth_{n-1}$. Experimental results show an average reduction of 37.9% in the encoding time and a BD-Rate increase of 0.62%.

Focusing on reducing the amount of intra-prediction mode evaluations, [13] proposed an algorithm for determining edge directions in prediction blocks. The idea is to evaluate the original block contents, categorizing the prediction direction into one of five groups, similarly to the earlier discussion in [35]. Computing the dominant edge of a prediction block occurs in PUs of size 4 × 4, which the algorithm subdivides into four 2 × 2 regions, labeled as $c0$, $c1$, $c2$, and $c3$, shown in Figure 4.3(a). Each weight of the region determine the PU category into vertical, horizontal, 45° edge, 135°, and non-directional edge. The edge direction algorithm subdivides PUs larger than 4 × 4 into 4 × 4 blocks, as shown in Figure 4.3(b), mapping each 4 × 4 block to one of the five categories. The category with the highest number of occurrences in the PU becomes the dominant edge.

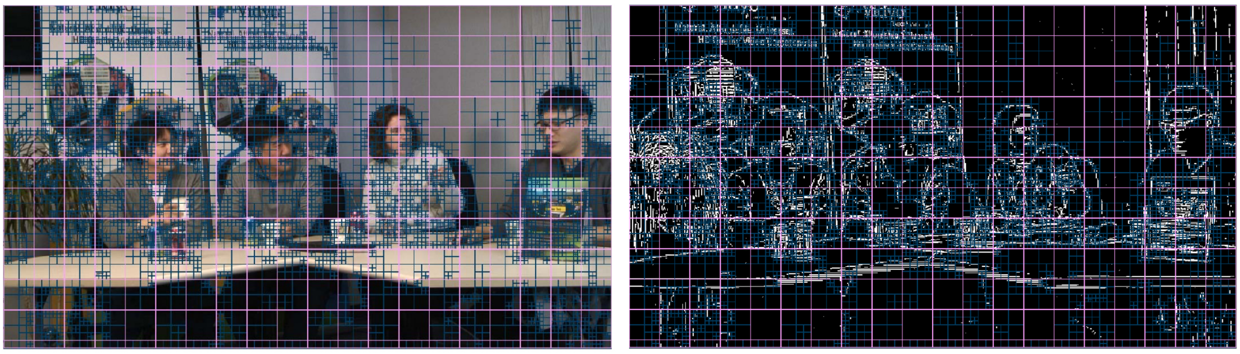


(a) 4 × 4 PU edge computation

(b) 16 × 16 PU edge computation

Figure 4.3 – Edge computation for different PU sizes. The algorithm subdivides blocks larger than 4×4 into 4×4 blocks for edge computation. Meanwhile, the dominant edge of a 4×4 block is determined according to a subdivision of its contents in 2×2 regions (captured from [13]).

Each one of the five PU categories holds a subset of the 35 HEVC intra-prediction modes. Therefore, intra-prediction computation requires evaluating only this subset, decreasing encoding complexity for choosing a proper intra-prediction mode. Their results show an encoding time and PSNR reduction of 32.08% and 0.02*dB*, respectively, and a bitrate increase of 0.9%.



(a) Frame CTU subdivision

(b) Frame outliers with overlayed CTU subdivision

Figure 4.4 – Frame CTU partition and its outliers. Smaller CUs occur in more complex texture regions, which correspond to the frame outliers (captured from [24]).

In [24] the authors proposed reducing the encoding complexity, resembling the work of [15]. Using image outliers, they determine how to split the CTUs according to complex texture frame regions. Figure 4.4 illustrates the correlation of CTU partitioning (Figure 4.4(a)) for a given frame with its outliers (Figure 4.4(b)). Following a statistical analysis of outlier information, their method can determine a CTU depth, which offers good coding efficiency for the given frame region. The experimental results show an average encode time saving of 50% with 0.7% of BD-Rate penalty. Their work also proposes a refinement for entropy coding that mitigates the coding efficiency loss by reorganizing the CABAC BINS according to their new CTU partitioning strategy.

4.2 HEVC Intra-Frame Prediction Improvements

In [20], the authors use a Markovian model based on neighboring blocks to predict sample values from a linear equation. A training window uses the information of previously decoded samples, which varies according to the predicted sample location at the frame. The application of linear regression to samples from the training window estimates the predicted sample values, minimizing the prediction error by applying the least-squares criterion. The results showed, on average, a PSNR increase of 0.41dB and a bitrate reduction of 7.34%, both compared to H.264, although with an increase in encoding and decoding times close to 150%.

An extension of [20] exists for HEVC in [36], where the authors proposed Sparse Least Squares Prediction (SLSP), following the same principle of a training window for predicting sample values. The training window locally computes coefficient values, avoiding the necessity to transmit additional information in the HEVC bitstream, as both the encoder and decoder can implicitly recover the necessary information. Figure 4.5 illustrates the concept of the training window. The causal search window **W** of dimensions $D_H \times D_V$ corresponds to the training window, and the template **C** is the set of training samples, encompassing a region of **T** samples left and above of the predicted block **P**.

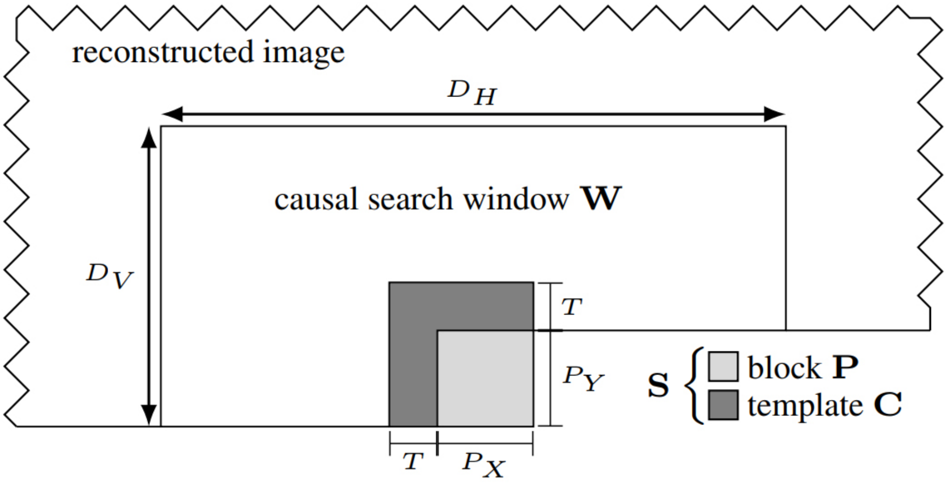


Figure 4.5 – SLSP training window. Selected samples **C** from the training window **W** are used to predict the contents of the current predicted block **P** (captured from [36]).

SLSP finds samples inside the region **W** that show a high correlation with the **T** samples, producing a set of linear coefficients used to predict all samples of block **P**. For the experiments, the SLSP prediction mode replaces one of the angular intra-frame prediction modes of HEVC. Results illustrate PSNR gains of up to 1dB across all test sequences, with a bitrate reduction of at most 15%, at the cost of an order of magnitude increase in computational complexity due to the search sample training procedures.

The authors in [38] improve HEVC intra-frame prediction by using more reference samples for predicting block contents. Instead of relying solely on the standard linear filter used by HEVC, they use a Discrete Cosine Transform Interpolation Filter (DCT-IF) to improve intra-prediction performance, especially for small blocks such as 4 × 4 and 8 × 8. HEVC employs a two-sample filter to compute prediction units in intra-angular modes, as shown in Equations 3.5 and 3.8 in Chapter 3, e.g., to predict a particular sample S in a prediction block, depending on the prediction mode used, a reference sample A_0 is used with its adjacent sample A_1. HEVC uses this filter regardless of the block characteristics. Using DCT-IF enables the use of ≥ 2 reference samples, e.g., A_{-1}, A_0, A_1, A_2.

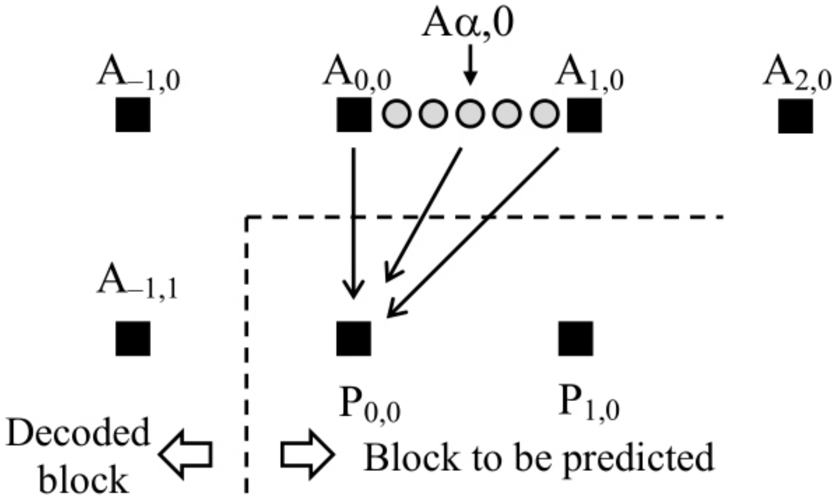
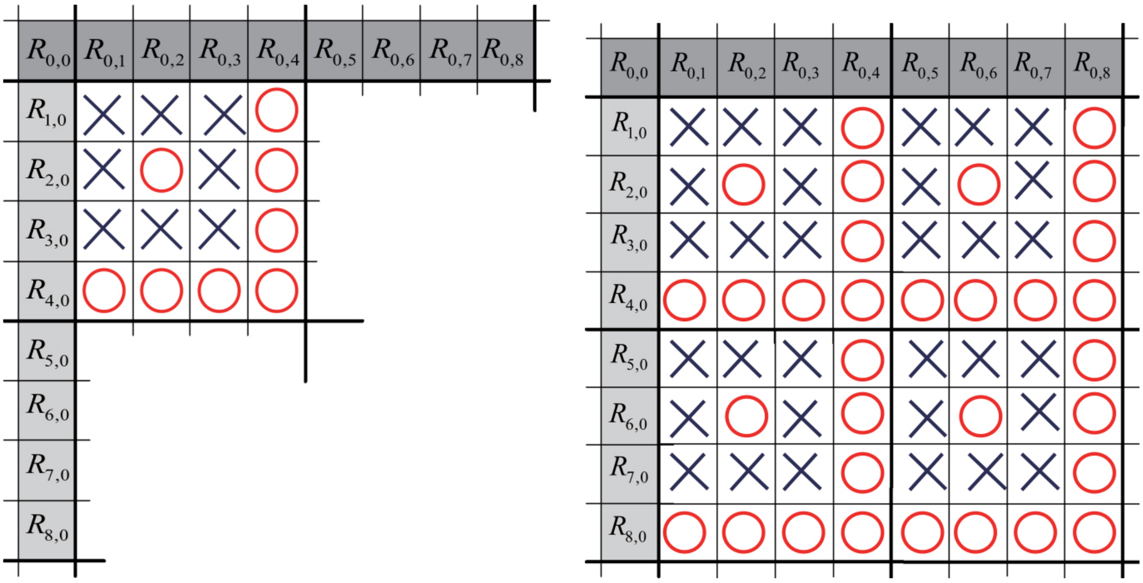


Figure 4.6 – Reference samples interpolation. Depending on the angular intra-frame prediction mode and reference array size, DCT-IF interpolates samples at the fractional positions (captured from [38]).

Depending on the angular intra-prediction mode and reference array size, DCT-IF interpolates reference samples at the fractional positions, i.e., A_0, $A_{1/2}$, A_1 as Figure 4.6 illustrates. Their results show average BD-Rate reductions ranging from 0.31% to 0.34%, depending on the filter size, with maximum gains up to 2.2%. Since the authors change the prediction methods, both the encoder and the decoder require more computational effort. However, the results have shown, at most, an increase of 6% and 1% in encoding and decoding times, respectively.

In [9], the authors introduced a technique for intra-frame encoding that employs constrained quantization [80]. Studies in video coding [26] show that predicted samples closer to the reference samples in intra-prediction have a high correlation, producing fewer residuals than samples found farther from the reference samples. Therefore, [9] proposes to encode only half of the PBs samples, which are farther away from the reference samples and subject to larger residuals. Interpolation of the reference array and previously decoded samples reconstructs the remaining, small residual samples. Figure 4.7 shows the selected samples for constrained quantization, where encoded samples correspond to the red circles and interpolated samples to the blue crosses.


$R_{0,0}$ $R_{0,1}$ $R_{0,2}$ $R_{0,3}$ $R_{0,4}$ $R_{0,5}$ $R_{0,6}$ $R_{0,7}$ $R_{0,8}$
$R_{1,0}$ $R_{2,0}$ $R_{3,0}$ $R_{4,0}$ $R_{5,0}$ $R_{6,0}$ $R_{7,0}$ $R_{8,0}$

(a) 4 × 4 PB

(b) 8 × 8 PB

Figure 4.7 – Coding patterns for constrained quantization. Encoded samples correspond to the red circles, while interpolated samples to the blue crosses (captured from [9]).

Since encoding considers only half the samples in PBs, quantization prioritizes these samples to retain higher image quality. This approach results in better reconstruction quality for the encoded samples, translating into better quality for the interpolated samples. The experiments show an average reduction of $\approx 2\%$ in BD-Rate, with an increase of 130% in the encoding complexity. Meanwhile, the authors report that the decoding complexity stays unchanged.

All previously discussed work considers the same reference sample array model of HEVC for intra-frame prediction. In [33], the authors use reference lines further than the above and left block boundaries to approximate PB contents. Considering the first reference line at the block boundary as L_0, their work uses lines up to L_M, limited at $0 \leq M \leq 3$. Figure 4.8 illustrates their multiline approach for intra-prediction, where M is the line offset from the default reference line at L_0.

The encoder chooses to use multiline intra-prediction according to standard rate-distortion metrics, evaluating all combinations of line offsets and intra-frame prediction modes. Evaluations show that $\approx 40\%$ of prediction modes used their multiline approach, indicating that the authors obtained better intra-prediction when using lines further than L_0 as the

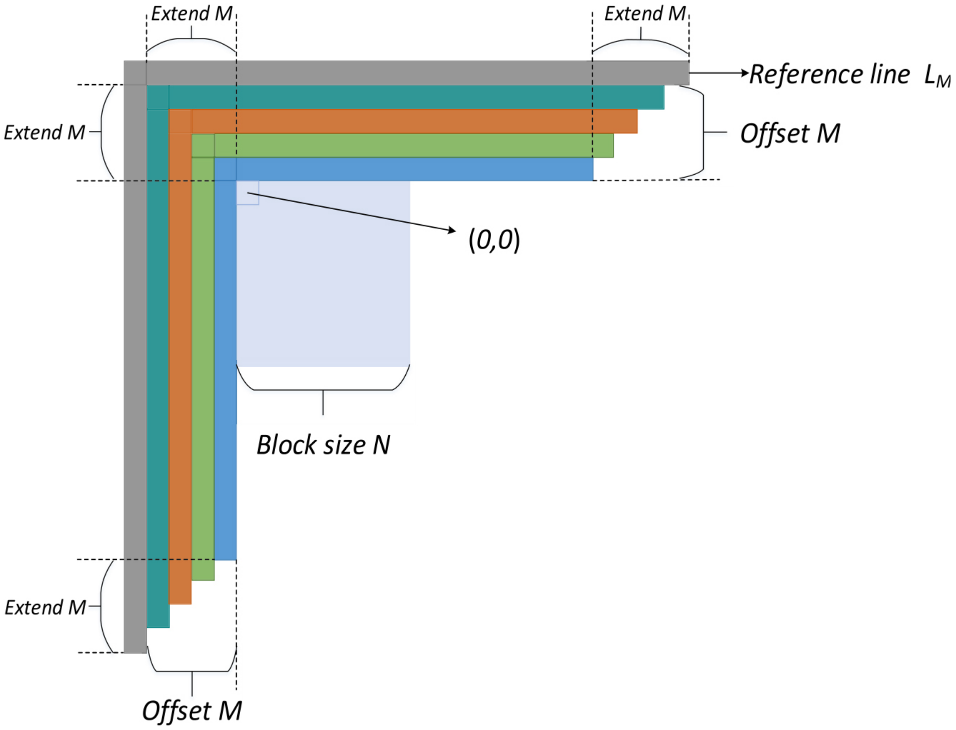


Figure 4.8 – Multiline intra-prediction approach. Increasing the offset M from the default reference line requires a corresponding extension of the reference array (captured from [33]).

reference array. Signaling of the offset parameter requires embedding its value into the HEVC bitstream. The approach is similar to the signaling of reference indexes of P and B frames, indicating for each PB which reference array to use for decoding. Experimental results show an average BD-Rate reduction of 2.3%, with a maximum of 4.3%, while encoding time almost quadruplicates, at 363%. The work of [33] is also one of the many contributions to the state-of-the-art about intra-frame prediction that the VVC standard adopts as Multi-Reference Line (MRL) prediction [8].

Another intra-prediction technique, which the VVC standard adopts, is Position Dependent Prediction Combination (PDPC) [49]. PDPC addresses one inherent limitation of intra-prediction in HEVC: the coding process adopts the same directional prediction technique, regardless of the block size. According to [49], samples in the PB closer to the reference array provide better prediction quality than distant samples. This prediction quality degradation becomes even more apparent in larger PBs, with HEVC applying smoothing filters that depended on the block size and prediction type. The previously mentioned work about constrained quantization [80] posed similar considerations about the degree of correlation of predicted samples closer to the reference array, indicating an interesting research venue for predictive coding techniques.

Figure 4.9 illustrates a predictor matrix for an 8×8 PB in HEVC. The vertical axis numbered 0 to 34 corresponds to the HEVC intra-prediction mode, while the horizontal axis correlates the reference sample, indicated by coordinates $[-1, y]$ and $[x, -1]$ for the left the top reference arrays, respectively, with the predicted block, indicated by the small blocks in Figure 4.9. The colors of the small blocks stand for the correlation of prediction mode and reference sample in each PB, with 0 correlation as gray, positive correlation as lighter colors, and negative correlation as darker colors. The pattern in Figure 4.9 shows that samples further away from the reference array tend to blur with their neighboring samples since the correlation of these predicted samples with the reference samples decreases. The earlier work on constrained quantization [80] also makes this same observation of the decreasing correlation of samples further away from the reference array.

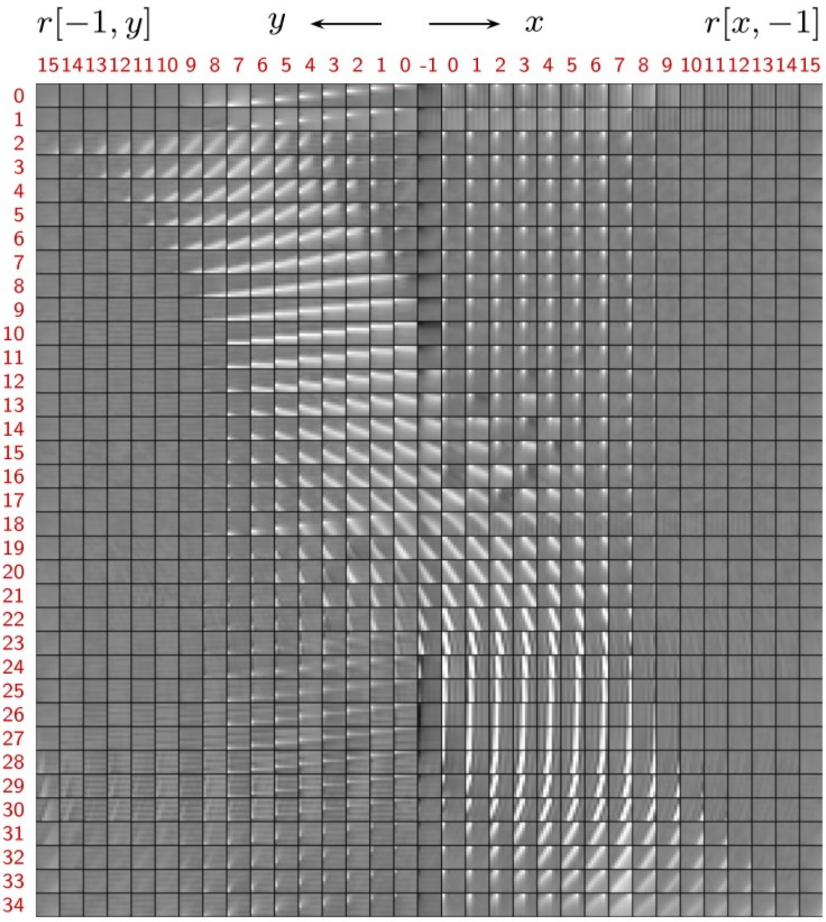


Figure 4.9 – PDPC sample correlation matrix for an 8 × 8 PB in HEVC (captured from [49]).

PDPC employs the statistical analysis of filtered and unfiltered reference samples to decide the prediction parameters based on the predicted sample position, prediction mode, and block size. PDPC uses the prediction parameters to solve a set of linear equations that best approximate the predicted sample values in PBs, using the correlations seen from the HEVC intra-prediction modes and reference samples. The experimental results for PDPC show an average BD-Rate reduction of 2% while keeping low computational complexity and enabling a full parallel implementation.

4.3 Overview

Although this chapter provides only a brief overview of the state-of-the-art works about encoder complexity reduction, it is still relevant to this Thesis. The proposed techniques for improving intra-frame prediction, discussed in Chapter 5, result in increased encoder computational complexity, addressable by the ideas discussed in Section 4.1. Most of the proposed work addressed two key aspects of the intra-frame prediction process: (i) CTU subdivision; and (ii) intra-prediction mode selection. Providing optimized strategies for dealing with these aspects of intra-frame prediction showed significantly improved encoding times with minimal coding efficiency penalties. Therefore, further exploration of similar strategies may contribute to future work on this Thesis.

Table 4.1 – Overview of related work.

Author	Year	Details	Gains	Trade-off
Lucas et al. [36]	2015	Linear regression with search window	15% bitrate reduction, 1dB PSNR gains	Order of magnitude increase in encoding times
Matsuo et al. [38]	2012	Interpolator refinement with ≥ 2 samples	$0.31\% - 0.34\%$ average BD-Rate reduction	6% increase in encoding times
Chen et al. [9]	2017	Encodes samples further from the reference array. Reconstructs with interpolation	$\approx 2\%$ average BD-Rate reduction	130% increase in encoding times
Li et al. [33]	2018	Multiline reference arrays (MRL, adopted by VVC)	2.3% average BD-Rate reduction	Quadruplicates encoding times. Bitstream modifications
Said et al. [49]	2016	Statistical prediction based on sample properties (PDPC, adopted by VVC)	2% average BD-Rate reduction, full parallel implementation	Encoding gains depend on the number of evaluated sample properties
Proposed Work		Polynomial regression	Up to 2.45 dB PSNR gains	Greatly reduced coding efficiency
		Curve-based angular modes	Up to 4% average BD-Rate reduction	Increases encoding times depending on amount of curves. Bitstream modifications

Related work shows that exploiting intra-frame prediction algorithms can improve coding efficiency by obtaining a better prediction of encoded block contents. The techniques employed by the authors vary, as shown in Table 4.1, targeting distinct aspects of the intra-frame prediction process, often proposing novel ideas with few similarities among them. This Thesis is one such case, providing a novel and promising approach for enhancing the intra-prediction process with curve-based prediction modes since it enables the modeling of characteristics found in natural images with higher accuracy than the usual straight lines adopted by intra-predictors of video coding standards, like HEVC.

5. PROPOSED INTRA-PREDICTION MODELS

The HEVC intra-frame prediction demands high computational effort, as the encoder recursively evaluates all prediction modes according to the CTU subdivision structure. Additionally, intra-frame prediction produces residuals that need to be transform-coded, which may degrade the decoded video quality. However, as shown in Chapter 4, increasing encoding effort is often acceptable since the encoding of most broadcast videos occurs only once, while its decoding occurs countless times. As such, more accurate frame prediction models can decrease residuals, consequently improving coding efficiency.

Planar (mode 0) and DC (mode 1) are the most used modes out of the 35 available for HEVC intra-frame prediction, as illustrated in Figure 5.1. These values correspond to the average sum of the prediction modes used in the HEVC Test Sequences, according to the Common Test Conditions (CTC) defined in JCTVC-AF1100 [57]. These are reasonable values since most contents in natural images correspond to smooth surfaces, represented by modes 0 and 1, while complex textures are mostly pure vertical or horizontal, identified by intra-prediction modes 26 and 10, respectively. However, not all image content falls into these classes; hence, new prediction modes other than the previously mentioned offer opportunities for enhancing intra-frame prediction and encoding efficiency.

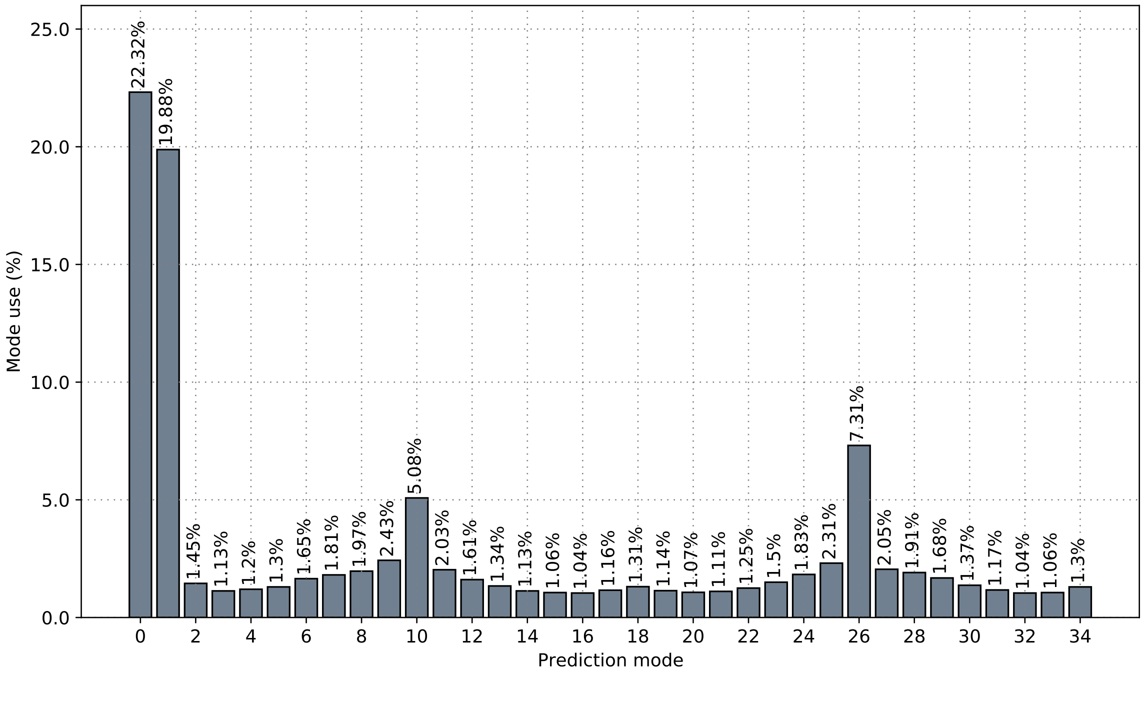


Figure 5.1 – Average use of HEVC intra-frame prediction modes. Planar and DC modes, used for smooth image contents, represent the most used prediction modes, followed by pure vertical (26) and pure horizontal (10) modes.

Two publications already supply the contributions of this Thesis to the state-of-the-art and corroborate with the underlying hypothesis: (i) in the International Conference on

Electronics Circuits and Systems ICECS [18]; and (ii) in the Electronics Letters journal [19]. The work shown in [18] proposes a new intra-frame encoding mode based on two-dimensional polynomial surfaces, which approximates sample values according to polynomial regression. The polynomial coefficients describe the block contents with high quality, albeit at much higher bitrate requirements than HEVC. Following the same idea of new prediction models that better represent block contents, the work presented in [19] proposed a new intra-frame prediction model based on curved lines, extending the coding capabilities of the HEVC intra-frame predictor. Instead of assuming only angular straight lines in predicted blocks, the use of curved lines captures texture characteristics more accurately. The curve-based prediction models improve the coding efficiency by reducing the prediction residuals, albeit at increased encoding times.

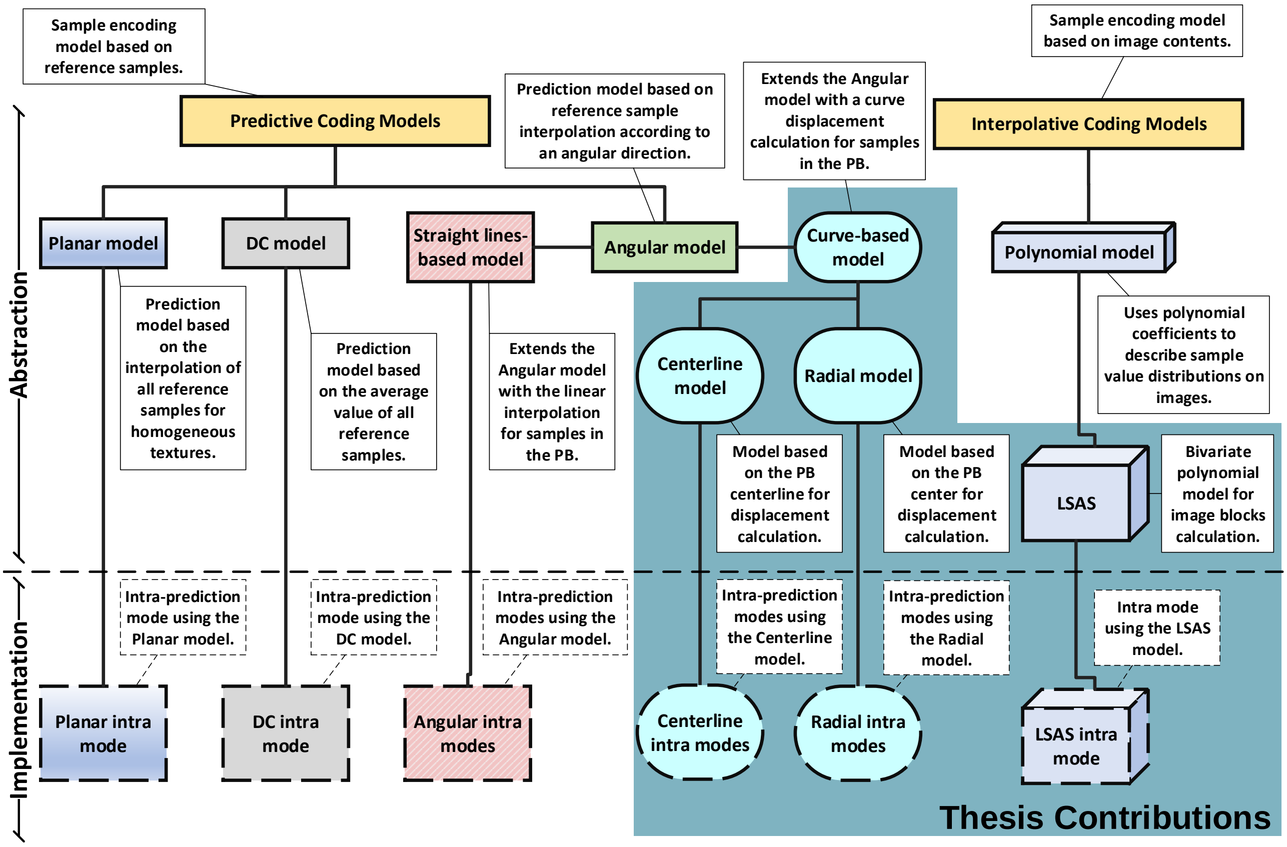

Sample encoding model based on reference samples.
Predictive Coding Models
Prediction model based on reference sample interpolation according to an angular direction.
Extends the Angular model with a curve displacement calculation for samples in the PB.
Sample encoding model based on image contents.
Interpolative Coding Models
Abstraction
Planar model
DC model
Straight lines-based model
Angular model
Curve-based model
Polynomial model
Prediction model based on the interpolation of all reference samples for homogeneous textures.
Prediction model based on the average value of all reference samples.
Extends the Angular model with the linear interpolation for samples in the PB.
Uses polynomial coefficients to describe sample value distributions on images.
Centerline model
Radial model
Model based on the PB centerline for displacement calculation.
Model based on the PB center for displacement calculation.
LSAS
Bivariate polynomial model for image blocks calculation.
Implementation
Intra-prediction mode using the Planar model.
Intra-prediction mode using the DC model.
Intra-prediction modes using the Angular model.
Intra-prediction modes using the Centerline model.
Intra-prediction modes using the Radial model.
Intra mode using the LSAS model.
Planar intra mode
DC intra mode
Angular intra modes
Centerline intra modes
Radial intra modes
LSAS intra mode
Thesis Contributions

Figure 5.2 – Intra-frame encoding models. The implementation of each model is specific to the video coding standard, where intra modes identify the intra-frame coding models in the codec.

Figure 5.2 illustrates a diagram of the intra-frame encoding models used in HEVC and in this Thesis. The description for the predictive and interpolative coding models originate from a work from 1985 called “*Second-generation image-coding techniques*” [28], which, besides many other techniques, proposed the use of polynomial surfaces for encoding image regions.

The HEVC standard uses predictive coding models for intra-frame prediction, using reference samples from neighboring blocks for predicting the block contents. The curve-based model extends the idea of the angular prediction model, which interpolates reference frames according to an angular direction, with the inclusion of a curve displacement calculation for each predicted sample. In contrast to the predictive coding models used almost exclusively in modern coding standards, this Thesis proposes the Least-Squares Approximation Surfaces (LSAS) interpolative coding model to describe block contents. The implementation of each model is specific to the video coding standard, where intra modes identify the intra-frame coding models in the codec. These two contributions will be detailed in the next sections.

5.1 Least-Squares Approximation Surfaces

The first attempt at improving intra-frame prediction consists on the idea of segmenting images on block regions akin to modern image and video coding standards. To encode block regions, a two-dimensional polynomial function describes an approximation surface of the sample values. Thus, only the polynomial coefficients are necessary to reconstruct sample values that represent an arbitrary image block.

The LSAS model [18] estimates the sample values in prediction blocks by deriving the coefficients of the polynomial surfaces. Given an $n \times n$ block, where a pair of xy coordinates determines each sample in the block, Equation 5.1 shows the bivariate polynomial p assumed by the LSAS model as the surface equation. The polynomial order h is configurable to approximate the function, although requiring more coefficients for the calculations.

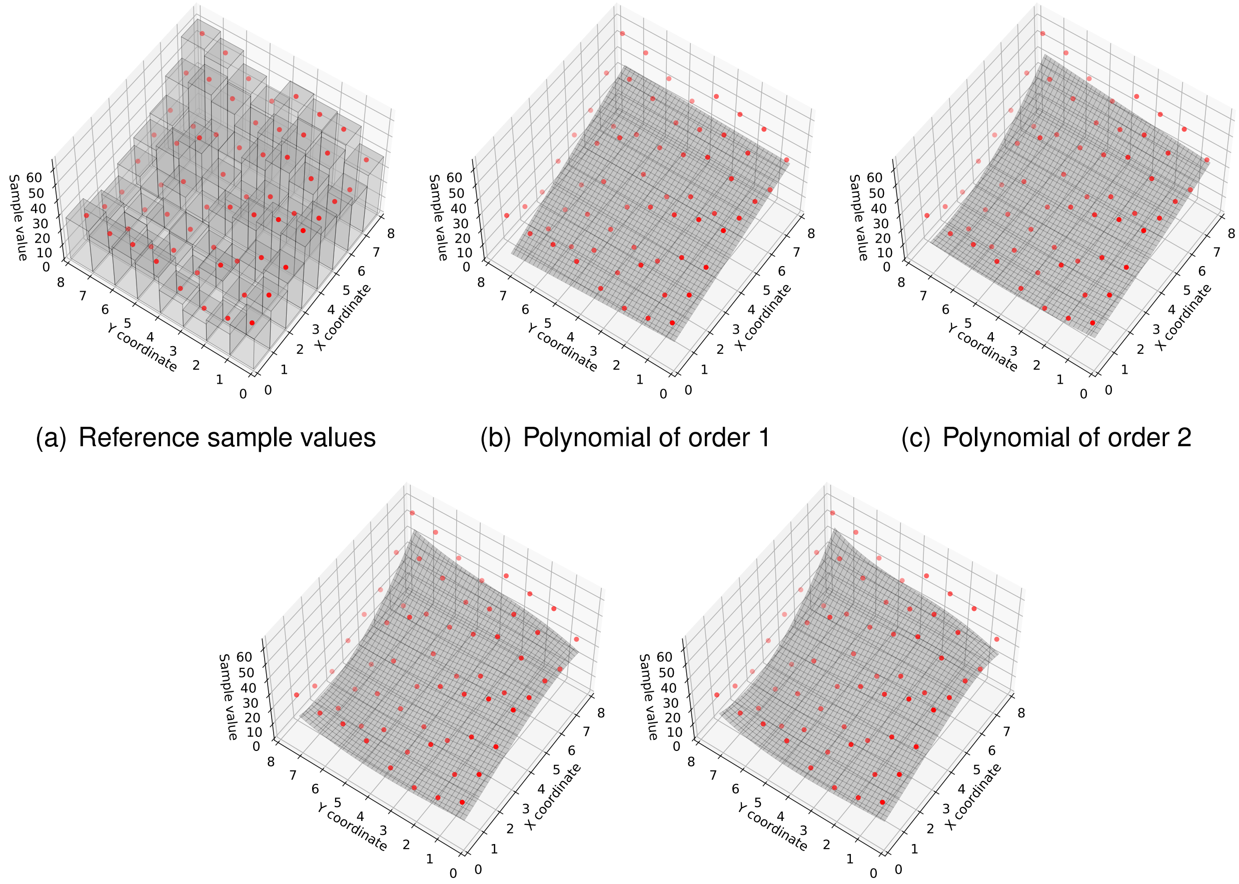
$$p(x, y) = \sum_{k=0}^{h} \sum_{l=0}^{h} a_{(k \times (h+1)+l)} x^k y^l \quad | \quad \forall k + l \leq h \tag{5.1}$$

A system of linear equations calculates the coefficients of each bivariate polynomial, whose approximation function has $Ax = b$ format regarding both xy coordinates, where A is the matrix of sample coordinates and b the sample values. The problem is then finding the set of coefficients in x that satisfy the equality. Additionally, solving the linear equations requires the definition of an error function. Equation 5.2 illustrates the error function E for a bivariate polynomial of order $h = 1$, represented by three coefficients, where z_{ij} is the measured sample value and $(a_0 + a_1 x_i + a_2 y_i)$ is the surface prediction. Regarding an image, x and y correspond to the sample coordinates, while z to the sample value. Solving the coefficients results in a fitness function that approximates samples in an image region of size $n \times n$.

$$E(a_0, a_1, a_2) = \sum_{i=0}^{n} \sum_{j=0}^{n} [(a_0 + a_1 x_i + a_2 y_j) - z_{ij}]^2 \tag{5.2}$$

Since the linear equation system has way more equations than coefficients due to the low order polynomials used for approximating the high amount of input samples, it requires the multiplication of A by its transpose A^T, and to solve the linear equation $A^T A\hat{x} = A^T b$, with $\hat{x}$ as the least squares solution [31]. Applying Equation 5.2 for solving the linear system $A^T A\hat{x} = A^T b$ results in the three linear equations shown in Equation 5.3, which are easy to solve.

$$\begin{bmatrix} \sum_{i,j=0}^{n} 1 & \sum_{i,j=0}^{n} y_j & \sum_{i,j=0}^{n} x_i \\ \sum_{i,j=0}^{n} y_j & \sum_{i,j=0}^{n} y_j^2 & \sum_{i,j=0}^{n} x_i y_i \\ \sum_{i,j=0}^{n} x_i & \sum_{i,j=0}^{n} x_i y_i & \sum_{i,j=0}^{n} x_i^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum_{i,j=0}^{n} z_{i,j} \\ \sum_{i,j=0}^{n} y_i z_{i,j} \\ \sum_{i,j=0}^{n} x_i z_{i,j} \end{bmatrix} \tag{5.3}$$



(a) Reference sample values

(b) Polynomial of order 1

(c) Polynomial of order 2

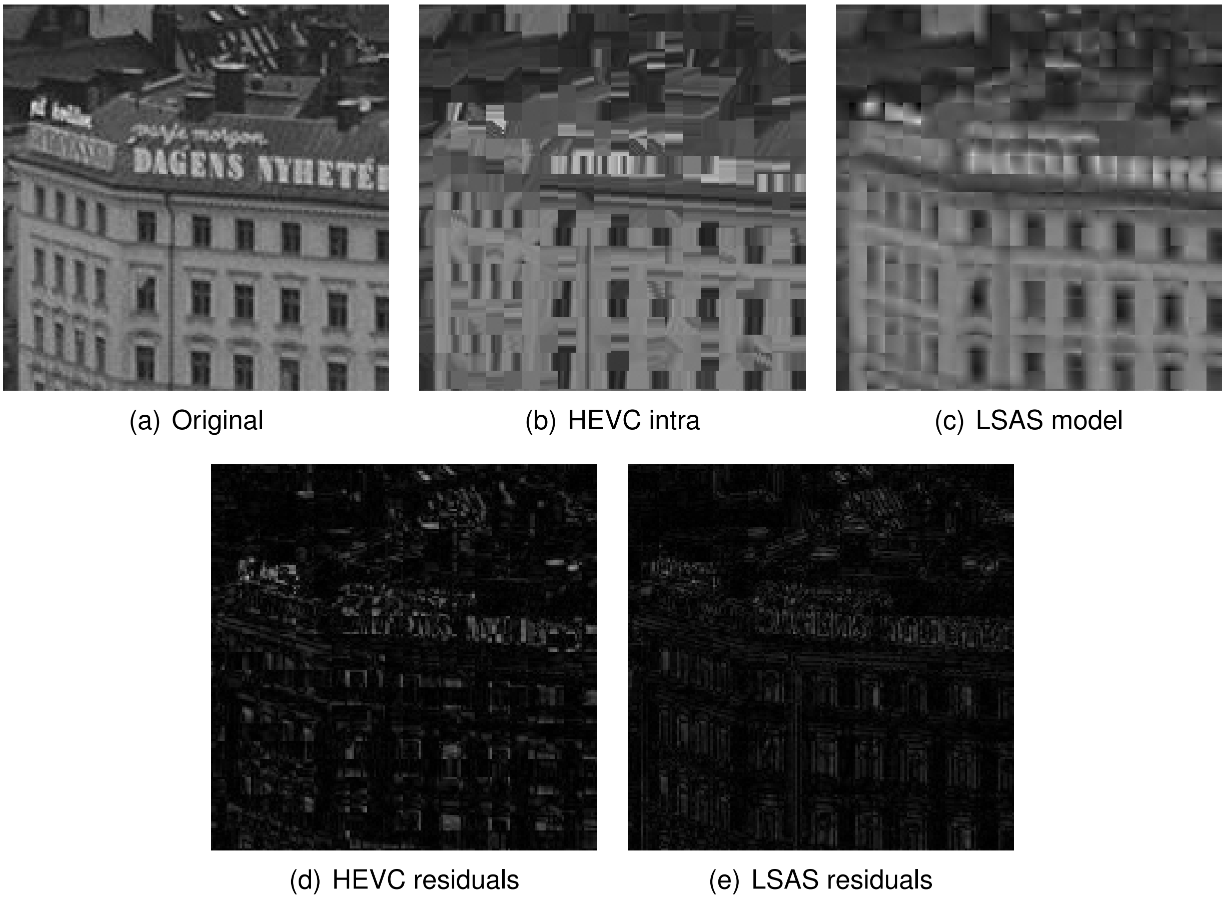
(d) Polynomial of order 3

(e) Polynomial of order 4

Figure 5.3 – LSAS model approximation surfaces for four different polynomial degrees. Red dots denote the original samples, where their heights correspond to the sample value. Higher degree polynomial surfaces achieve a better approximation of the original values.

The solution for Equation 5.3 uses the Gauss-Seidel [77] iterative method until achieving a satisfactory convergence for the coefficients. According to the experimental analysis, no more than twenty iterations were necessary in all cases to compute a solution. The result is a polynomial surface whose coefficients fit the sample distribution in the block space, minimizing the sample value error as much as possible. Figure 5.3 illustrates four approximation surfaces for an arbitrary image block. The red dots represent the original sample values in the block space, where the height of the red dots denote the sample value, i.e., its luminance or chrominance value. Employing higher-order polynomials produces surfaces with a better approximation of the block contents, thus minimizing the error. Still, computing higher-order polynomials requires additional computational effort and more polynomial coefficients for describing the image region.

The proposed technique adapts well to image regions with curved edges inside a prediction block due to the nature of the approximation surfaces. Additionally, there are no spatial dependencies of neighboring blocks to compute the coefficient values, enabling a parallel execution of LSAS modes.



(a) Original (b) HEVC intra (c) LSAS model

(d) HEVC residuals (e) LSAS residuals

Figure 5.4 – LSAS model compared to HEVC intra-prediction. Comparison of prediction models: (a) original image; (b) HEVC intra-frame prediction; (c) LSAS model. (d) shows the residuals for the HEVC intra-predictor, while (e) is the residual for the LSAS model.

The first drawback of using polynomial approximation surfaces is the resulting smoothing of image areas in neighboring samples close to a high contrasting region. Figure 5.4(a) illustrates an image region with significant contrast variation. Figure 5.4(b) shows the approximation computed by HEVC, combing all 35 intra-prediction modes to predict the image region. Meanwhile, Figure 5.4(c) shows the LSAS model using order four polynomial surfaces. The figures illustrate that the HEVC prediction produces sharper and well-defined edges, while the LSAS model produces smooth surfaces. Figures 5.4(d) and 5.4(e) illustrate the residuals for HEVC and LSAS model, respectively, where LSAS visibly presents fewer residuals than HEVC intra-prediction.

The second drawback, and the most significant, is the coding efficiency. Each coefficient is a floating-point number, which requires several bits to encode its value accurately. Meanwhile, increasing the polynomial order results in more coefficients, increasing even further the bitstream requirements. Assuming 16-bit floating-point precision, a first-order polynomial surface requires 3 × 16 = 48 bits for encoding a PB. Table 5.1 shows the bitstream requirements per PB for encoding different polynomial orders. Compared to HEVC, this cost is much higher, as a PB in HEVC requires up to 6 bits for encoding one of its 35 intra-prediction modes.

Table 5.1 – LSAS model bitstream requirements per PB for polynomial orders $1 \leq h \leq 4$, assuming 16-bit and 32-bit floating-point precision for the coefficients.

		Required bits	
Polynomial order (*h*)	**Coefficients**	**16-bits precision**	**32-bits precision**
1	3	3 × 16 = 48	3 × 32 = 96
2	6	6 × 16 = 96	6 × 32 = 192
3	10	10 × 16 = 160	10 × 32 = 320
4	15	15 × 16 = 240	15 × 32 = 480

5.2 Curve-based Intra-frame Prediction

The LSAS model offers possibilities for extending the capabilities of HEVC angular intra-frame prediction. HEVC intra-frame prediction calculates samples in a PB according to previously reconstructed samples from neighboring PBs. According to the PB prediction direction, modes 2 to 34 use two-point interpolation to draw straight lines from the reference array. Figure 5.5 illustrates this idea for all HEVC angular intra-prediction modes.

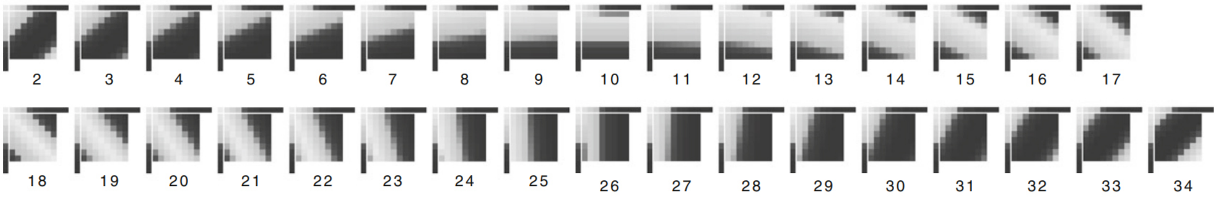


Figure 5.5 – HEVC angular intra-prediction modes. Block prediction follows straight lines from the reference array, according to the prediction direction (captured from [61]).

One interesting aspect of the LSAS model is that non-straight lines may present a possibility for modeling block contents, much like in natural images. For the polynomial surface shown in Figure 5.3(e), consider its projection into a two-dimensional plane, illustrated in Figure 5.6; each contour line corresponds to samples with similar values in the PB, while the color gradient corresponds the luminance variation among samples. Figure 5.6 clarifies that the polynomial surfaces can capture such detail of the original image region; thus, the second attempt for improving HEVC intra-frame prediction uses this idea.

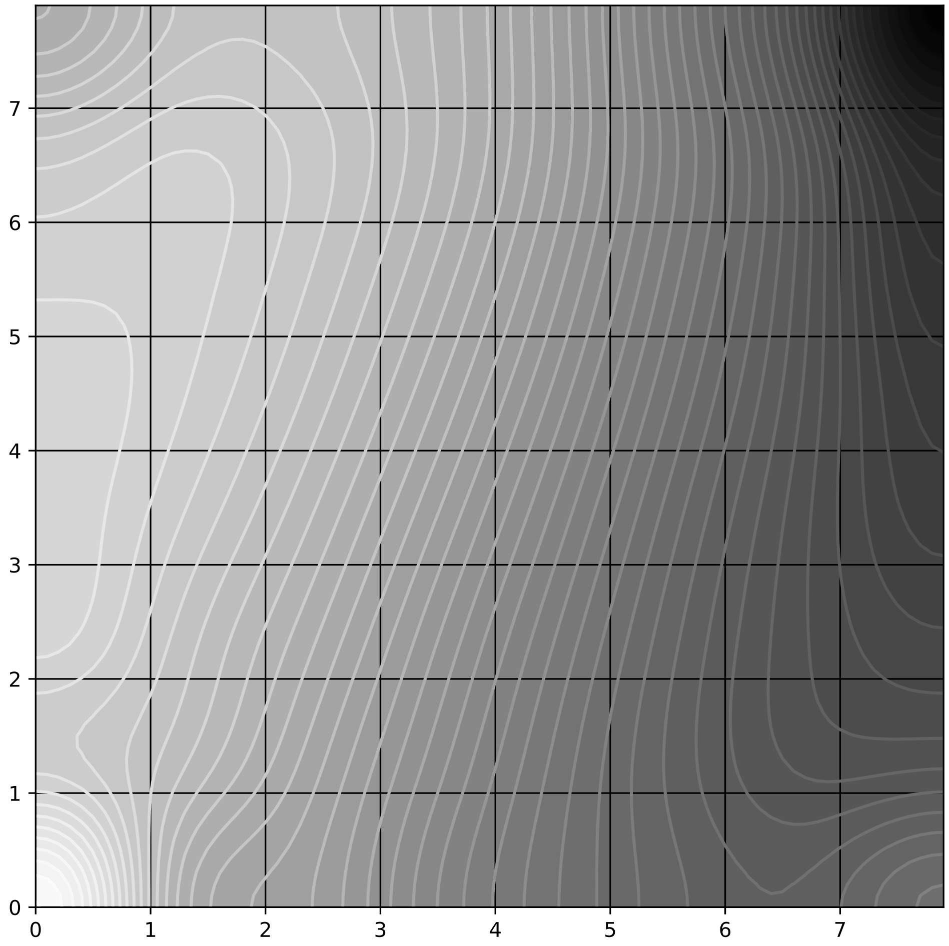


Figure 5.6 – LSAS model polynomial surface projection of the polynomial of order 4 surface in Figure 5.3(e). Each cell corresponds to a sample in the 8 × 8 PB, while the color gradient representing the luminance variation among samples. The contour lines represent samples with similar luminance in the PB.

The core contribution of this Thesis is a novel approach to compute intra-frame predicted blocks. Each PB sample is subject to a displacement operation based on the sample position and a curve displacement value; thus, extending the 33 angular intra-prediction modes of HEVC. The proposed curve-based prediction model considers that non-straight lines in angular intra-prediction improves natural image content modeling. Following the same encoding techniques of HEVC, this novel approach considers the reference samples from neighboring reconstructed PBs to approximate block contents. For this purpose, a new curve parameter, which specifies the intensity of the curve, changes the HEVC intra-frame predictor to compute new PB possibilities. The idea is to model the curves as arcs, extending away from the prediction direction, using a curve displacement value ω, either to the left of the angular direction (negative displacement value) or to the right of the angular direction (positive displacement value). This Thesis presents two curve-based models for computing the curves in PBs: (i) the Centerline model, discussed in Section 5.2.1, uses an anchor relative to a central line in the PB, which assumes two possible configurations depending on whether it is a vertical or a horizontal prediction mode; and (ii) the Radial model, discussed in Section 5.2.2, places an anchor at the center of the PB.

Considering the idea of fitting a curve parameter in the HEVC intra-frame interpolator, each curve-based model modifies Equation 3.5 and Equation 3.8, shown in Section 3.2, at the reference sample indexing step, applying a displacement calculation, pointing to other samples in the reference array. Both curve-based models also require modifications to the reference arrays and the HEVC bitstream, detailed in Sections 5.2.3 and 5.2.4, respectively. Another important consideration is to avoid floating-point operations in video codec implementations, to prevent encoder-decoder mismatch and drift caused by different floating-point implementations of the target platforms [61]. Thus, the implementation of both curve-based models considers only integer operations.

5.2.1 Centerline Model

The first curve-based model considers a centerline across the PB. This line can be either vertical or horizontal, according to the angular prediction mode. Horizontal modes employ a vertical centerline, while vertical modes employ a horizontal centerline. The ω value applies a stronger curve displacement to samples closer to the centerline, while samples closer to the PB border are less affected by the displacement.

The Centerline model assumes *Hb* as half of the block size and *d* as the distance from the predicted sample to the centerline of the block, given in Equations 5.4 and 5.5 for vertical and horizontal modes, respectively. Equation 5.6 describes the weight of the curve displacement applied to the reference sample computed by HEVC intra-frame prediction. This equation defines that samples close to the border of a prediction block are less affected by ω. In contrast, samples near the center are most affected by the curve displacement since $Hb - d$ is minimum in the borders and maximum near the center, remembering that the division operation uses integer division. Figure 5.7 illustrates this behavior for an 8 × 8 PB using vertical prediction modes.

$$d = \begin{cases} Hb - y - 1, & y < Hb \\ y - Hb, & y \geq Hb \end{cases} \tag{5.4}$$

$$d = \begin{cases} Hb - x - 1, & x < Hb \\ x - Hb, & x \geq Hb \end{cases} \tag{5.5}$$

$$cshift = ((Hb - d) \times \omega) \div Hb \tag{5.6}$$

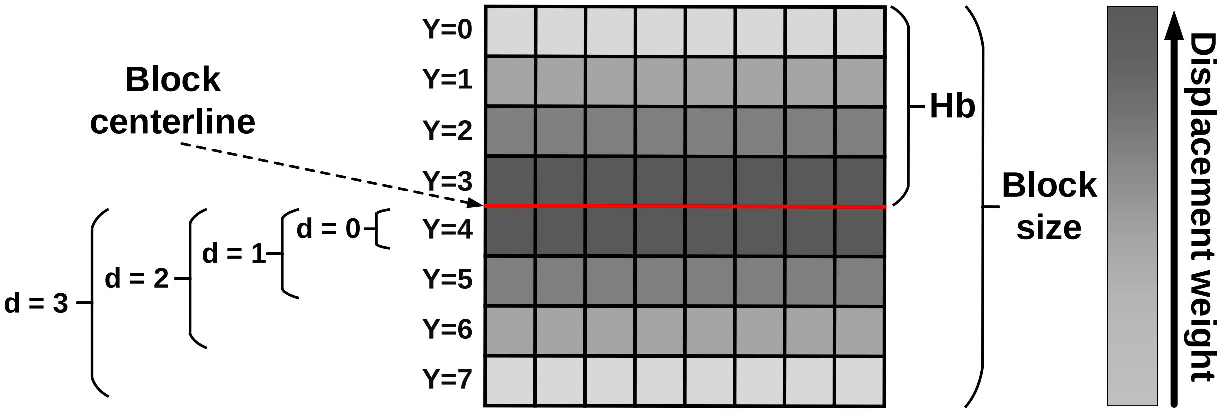


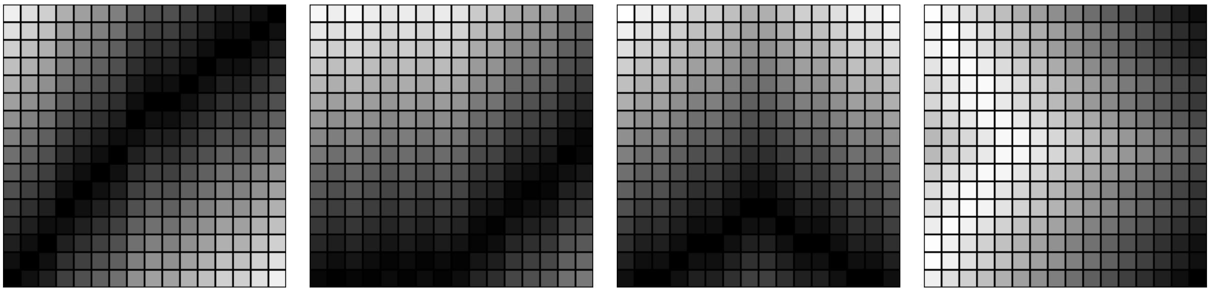
Figure 5.7 – Centerline model curve displacement weight (ω) for an 8 × 8 PB in vertical prediction modes. Samples located near the centerline are more influenced by ω, while samples further away are less affected [19].

After solving Equation 5.6, the *cshift* parameter changes Equations 3.5 and 3.8 at the reference sample indexing step, resulting in the new equations 5.7 and 5.8, for vertical and horizontal modes, respectively. Intra-prediction then uses these new Equations to compute the predicted sample values. When $\omega = 0$, i.e., without any curve displacement, the prediction behaves exactly like in the standard HEVC intra-prediction, as the *cshift* parameter is zero. Still, the angular intra-frame predicted sample values with the new equations implies in minimal computational overhead. Compared to the original HEVC intra-predictor, the only requirement is solving Equation 5.6 for determining the reference sample values used in the PB.

$$p[x][y] = ((32 - f) \times ref[x + i + cshift + 1] + f \times ref[x + i + cshift + 2] + 16) >> 5 \tag{5.7}$$

$$p[x][y] = ((32 - f) \times ref[y + i + cshift + 1] + f \times ref[y + i + cshift + 2] + 16) >> 5 \tag{5.8}$$

Figure 5.8 illustrates four curve examples that the Centerline model can represent for 16 × 16 PBs. The Centerline model creates arcs according to the PB direction, i.e., horizontal or vertical prediction modes. The curve inflection point is shifted away slightly from the PB center according to the prediction angular direction. This behavior occurs since sample displacement only considers the straight-line distance from the predicted sample to the centerline, thus producing the effect exemplified in Figure 5.8(b).



(a) Mode 2, $\omega = 2$ (b) Mode 5, $\omega = -3$ (c) Mode 10, $\omega = 5$ (d) Mode 27, $\omega = -5$

Figure 5.8 – Centerline model intra-prediction examples for a 16 × 16 PB.

5.2.2 Radial Model

The second curve-based model considers the distance of samples from the center of the prediction block. The distance calculation assumes the simple Euclidean distance $d(p_1, p_2)$ between two points in a two-dimensional space, given in Equation 5.9. The ω value applies a stronger curve displacement to samples located closer to the PB center in the Radial model. In contrast, farther samples are less affected by the curve displacement value.

$$d(p_1, p_2) = \sqrt{(p_{2x} - p_{1x})^2 + (p_{2y} - p_{1y})^2} \tag{5.9}$$

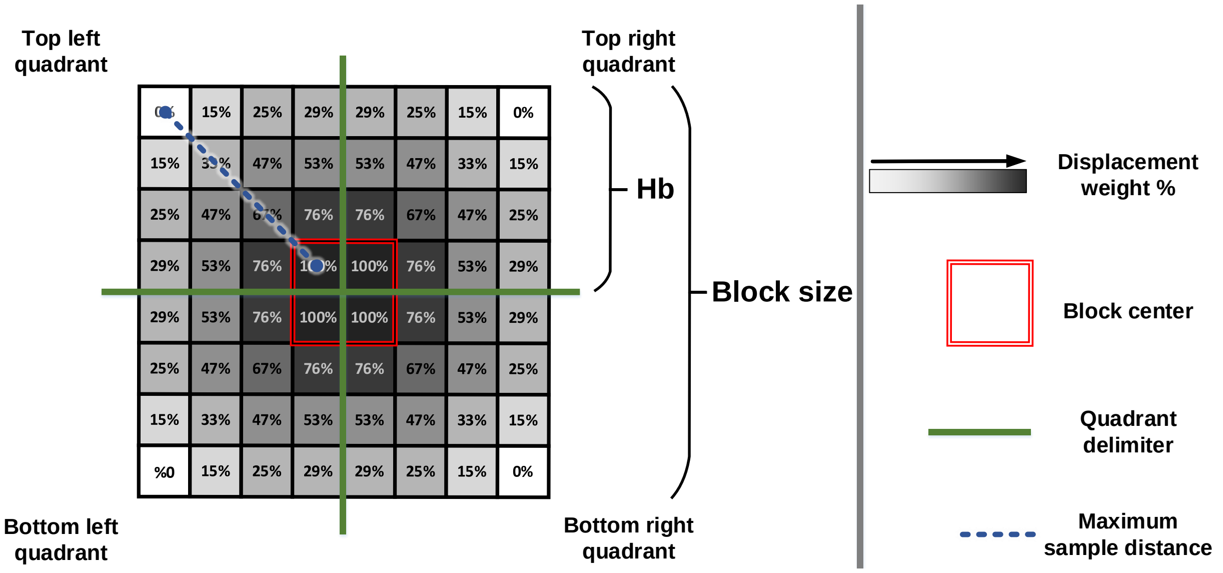
The accurate modeling of the Radial model requires addressing four essential aspects:

(i) PBs in HEVC are squares consisting of an even number of samples in both dimensions; thus, it is impossible to define a specific sample as the PB center, i.e., the geometrical PB center is not a sample. Therefore, the PB center consists of the four central samples, as illustrated in Figure 5.9;

(ii) This curve-based model considers the sample row and column relative to the closest of the four samples inside the PB center, i.e., for measuring the distance of a sample at the top left quadrant, the Radial model considers the top-left sample in the PB center. Figure 5.9 illustrates the four quadrants that divide each PB, i.e., top left, top right, bottom left, and bottom right quadrants;

(iii) The maximum distance a sample may have from the PB center occurs at the corners; e.g., for the 8 × 8 PB example in Figure 5.9, one of the maximum distances is in the sample three rows and three columns away from the center; and

(iv) Samples inside the PB center have a distance of zero.



0%	15%	25%	29%	29%	25%	15%	0%
15%	33%	47%	53%	53%	47%	33%	15%
25%	47%	67%	76%	76%	67%	47%	25%
29%	53%	76%	100%	100%	76%	53%	29%
29%	53%	76%	100%	100%	76%	53%	29%
25%	47%	67%	76%	76%	67%	47%	25%
15%	33%	47%	53%	53%	47%	33%	15%
%0	15%	25%	29%	29%	25%	15%	0%

Figure 5.9 – Radial model displacement weight (ω) for an 8 × 8 PB. Samples found near the center are more influenced by ω, while samples further away are less affected. The numerical values correspond to the displacement weight applied to each sample.

Representing the sample coordinates in the PB assumes that each sample s has an x and a y component for the row and column coordinates, respectively. Since the maximum distance a sample may have to the PB center is relative to its position in the quadrant, and samples inside the PB center have zero distance, then the maximum row and column distances for a sample are half of the block size minus one distance unit, which corresponds to $Hb - 1$, and $x_{max} = y_{max} = Hb - 1$. Thus, considering the Euclidean distance described in Equation 5.9, the maximum distance d_{max} of a sample from the PB center corresponds to $d_{max} = \sqrt{2 * (Hb - 1)^2}$.

Since the distance of each sample s considers the nearest sample inside the block center, the distance $d(s)$ of any sample is $d(s) = \sqrt{(s_x)^2 + (s_y)^2}$. The PB weight distribution is reached by applying a Curve Displacement Weight (CDW) to each sample according to d_{max} and its distance d from the center, with $CDW(s) = 1 - (d(s)/d_{max})$. The values inside each sample of Figure 5.9 correspond to the CDW value (expressed as a percentage of CDW) applied to each sample, similarly to the displacement weights of the Centerline model.

The Radial model implementation employs static tables with the CDW values of each PB size since the distance of each sample depends only on the PB size. Equation 5.10 defines the radial shift value $rshift$, applied to each sample s, remembering that the division operation uses integer division. Like in the Centerline model, $rshift$ changes Equations 3.5 and 3.8 at the reference sample indexing step, resulting in Equations 5.11 and 5.12 for vertical and horizontal modes, respectively.

$$rshift = (\omega * CDW(s)) \div 100 \tag{5.10}$$

$$p[x][y] = ((32 - f) \times ref[x + i + rshift + 1] + f \times ref[x + i + rshift + 2] + 16) >> 5 \qquad (5.11)$$

$$p[x][y] = ((32 - f) \times ref[y + i + rshift + 1] + f \times ref[y + i + rshift + 2] + 16) >> 5 \qquad (5.12)$$

Figure 5.10 illustrates the same four examples shown in Figure 5.8 for the Centerline model. The main difference is the curve inflection point, which is at the PB center in the Radial model. The curves bend around the PB center, forming a pattern that resembles a water ripple, as Figures 5.10(a) and 5.10(b) illustrate. However, for higher ω values, the curve around the PB center resembles an ellipse vertex, as shown in Figures 5.10(c) and 5.10(d). The elliptical patterns occur due to the variable curve displacement weights applied to each sample, with higher ω values tightening the curve around the PB center.

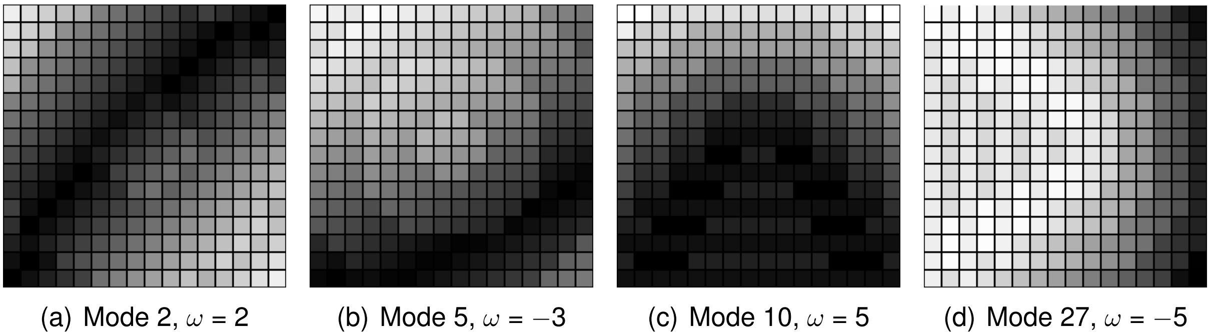

(a) Mode 2, $\omega = 2$ (b) Mode 5, $\omega = -3$ (c) Mode 10, $\omega = 5$ (d) Mode 27, $\omega = -5$

Figure 5.10 – Radial model intra-prediction examples for a 16 × 16 PB.

5.2.3 Reference Array Extension

The curve displacement and the intra-frame prediction direction may force some predicted samples to index references beyond the standard array shown in Figure 3.4. For instance, a prediction mode with positive angular parameter A and a large negative *shift* parameter may index a reference sample in the range $p[x][y]$ where $x < -1$ or $y < -1$. As described in Section 3.2, the reference array length is $2n + 1$ samples, indexing neighboring PBs according to the sign of the angular parameter A of the prediction mode.

To compensate for this limitation of the original HEVC implementation, the curve-based intra-prediction models always require the extension of the reference array and a larger reference array size with $3n$ samples, where n corresponds to the PB size, given as $n \times n$. Figure 5.11 illustrates the new reference array used when computing PBs with $\omega \neq 0$. As in HEVC, the reference array extension technique generates the reference samples indexed in the range $[-n, -1]$, while the generation of the remaining reference samples in the range $[0, 2n - 1]$ uses the same method employed by HEVC.

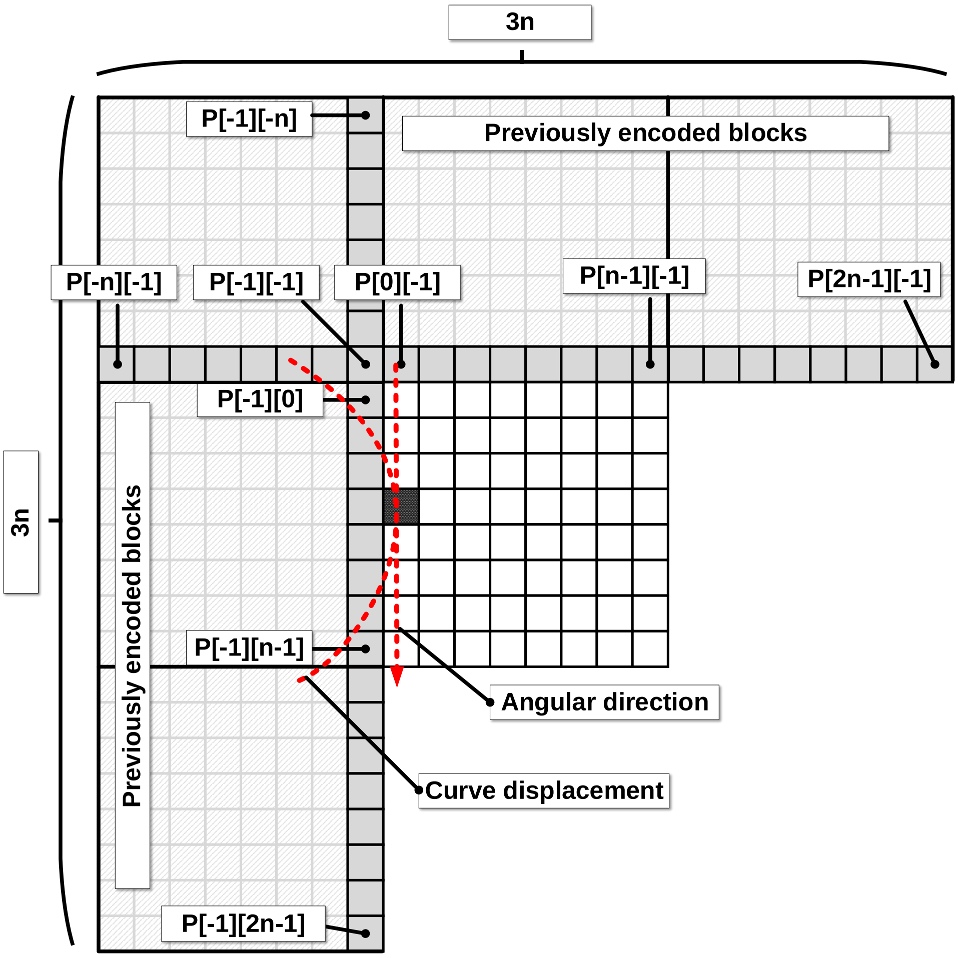


Figure 5.11 – Extended reference array for intra-frame prediction with curve displacements. The *shift* parameter requires using a larger reference array with samples from both positive and negative angular parameter situations.

While the previous modification of the reference array deals with situations related to the angular parameter A, there are situations when the *cshift* and *rshift* parameters may point to samples outside the range $[-n, 2n - 1]$. In large PBs or when ω is a large value, the predicted samples computed by the Centerline and Radial models may index elements outside the allowed range. Whenever such a situation arises, the adopted approach is to truncate the reference array index to the nearest value inside the allowed range $[-n, 2n - 1]$. This approach ensures that sample prediction always have a reference sample for the interpolator.

5.2.4 Syntax Elements and Bitstream Considerations

The HEVC standard defines four syntax elements for encoding an intra-predicted PU, namely:

(i) prev_intra_luma_pref_flag — a 1-bit flag derived from neighboring blocks, specifying if a PU uses one of the MPM;

(ii) mpm_idx — a 2-bit element, specifying the MPM index;

(iii) rem_intra_luma_pred_mode — a 5-bit element present when a PU does not use MPM; this element identifies the intra-frame prediction mode for the PU; and

(iv) intra_chroma_pred_flag — a 3-bit element, specifying the prediction mode in the chrominance PBs.

If a PU uses MPM, then rem_intra_luma_pred_mode is absent to reduce the bitstream size; this is an important optimization, as one of the MPMs is often the chosen intra-prediction mode for a PU. Additionally, even though HEVC uses 35 intra-frame prediction modes, rem_intra_luma_pred_mode has only 5 bits, limiting it to encode up to 32 prediction modes. This limitation optimizes the HEVC bitstream since the three missing modes in rem_intra_luma_pred_mode are in the MPM list.

Both curve-based models need modifications in the HEVC bitstream to signal the ω value used in each PB. The necessary binary length for encoding ω depends on the number of ω values used in the experiments, called θ, which is always an even number to achieve the symmetric displacement of samples in the experiments; e.g., $\theta = 2$ represents curve displacement values $\omega = \{-1;1\}$, while $\theta = 4$ represents $\omega = \{-1,1,-2,2\}$. The experiments, discussed in Chapter 6, adopt $\theta(m) = 2m$ for m in the range $[1,16]$.

This Thesis presents two approaches for encoding ω: (i) using fixed-length encoding, adopting the necessary binary length for encoding ω according to the number of ω values; (ii) using static Huffman codes, where each ω value assumes a binary length according to its probability of occurrence in the PUs.

The fixed-length encoding requires two additional syntax elements, namely:

(i) curve_displ_flag — a 1-bit flag, specifying if the ω value is not zero, i.e., the PU uses a curve; and

(ii) curve_displ — an n-bit element, specifying the ω value.

With fixed-length encoding, the curve_displ element requires n-bits, where $n = \lceil \log_2 \theta \rceil$; e.g., $\theta = 2$ requires 1 bit for encoding ω values -1 and 1, as shown in Table 5.2. Whenever $\omega = 0$ in the selected intra-frame predicted block, the curve_displ_flag element is 0, and the curve_displ element is absent in the bitstream. Therefore, the syntax element that identifies the ω value is only present when the predicted block requires it, following the same optimization idea used with rem_intra_luma_pred_mode for the MPM in PUs. This design choice reduces the overhead of the additional syntax elements in situations when they are unnecessary.

Table 5.2 – Curve displacement bitrate requirements. The required number of bits for fixed-length encoding is $\lceil \log_2 \theta \rceil$.

Number of displacements (θ)	**Required bits (n)**	**ω values**
2	1	$(-1, +1)$
4	2	$(-1, +1, -2, +2)$
6	3	$(-1, +1, -2, +2, -3, +3)$
8	3	$(-1, +1, -2, +2, -3, +3, -4, +4)$
	⋮	
20	5	$(-1, +1, -2, +2, \ldots, -10, +10)$

⋮

Meanwhile, the Huffman codes only employ the `curve_displ` element to signal the ω value. This work assumes static Huffman codes, known by the encoder and decoder. The ω values assume lengths according to their use probability. According to the results that Chapter 6 discusses, the experiments show that most PUs use prediction modes with $\omega = 0$, i.e., without curve displacements; this is coherent with the evaluation shown in Figure 5.1, where PUs mostly use Planar and DC modes. Therefore, according to the probabilistic nature of Huffman trees, the encoding of $\omega = 0$ should use short binary strings. Experimental evaluation shows that, for the tested scenarios, the Huffman trees always encode $\omega = 0$ with a single bit, working similarly to the `curve_displ_flag` used in fixed-length encoding since a single bit indicates whether the PU uses a curve displacement value. Figure 5.12 illustrates this idea, where the binary string "0" signals the $\omega = 0$ value. Any other string that begins with a "1" corresponds to $\omega \neq 0$. As θ restricts the number of available ω values, there is a specific Huffman tree for each value of θ adopted in the experiments.

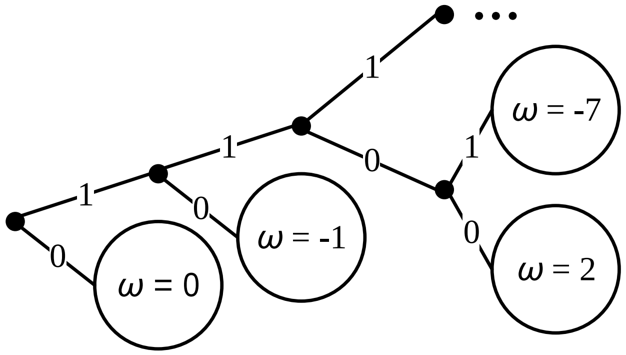


Figure 5.12 – Example of a Huffman tree representation for signaling the ω parameter in the PU.

Huffman encoding eliminates most of the negative aspects of fixed-length encoding, enabling the binary representation of ω according to its use frequency in the encoded bitstreams, encoding the often-used prediction modes with shorter binary strings while encoding the less used prediction modes with longer binary strings. Huffman codes also do not “waste” bits as in fixed-length coding for the θ values that are not a power of 2; i.e., while θ = 2 requires 1 bit for representing curve displacements $\omega = -1$ and ω = 1, and θ = 4 needs 2 bits for representing the curve displacements $\omega = -1$, ω = 1, $\omega = -2$ and ω = 2, θ = 6 requires 3 bits, whose binary combinations are not fully used for encoding the binary representation of the six values of ω, since 2^3 enables up to eight values for encoding ω.

Table 5.3 – Training sequences used for Huffman codes calculation. The binary length for the ω values depends on the prediction mode and the ω used across all PUs in all training sequences.

Resolution (pixels)	**Sequence name**	**Frame count**	**Frame rate (FPS)**	**Bit depth**
3840 × 2160	Foreman	515	60	10
3840 × 2160	Mobile	738	60	10
2560 × 1600	NebutaFestival	301	60	10
2560 × 1600	PeopleOnStreet	150	60	10
1920 × 1080	ArenaOfValor	600	60	8
1920 × 1080	Kimono	240	24	8
1920 × 1080	HoneyBee	600	120	8
1024 × 768	ChinaSpeed	500	30	8

To avoid biased results, this Thesis builds the Huffman codes according to a separate training sequence, shown in Table 5.3. The test sequences used for training adopt the JCTVC L1100, an earlier revision of the CTC for HEVC [4], and open repositories like [75].

6. EXPERIMENTAL RESULTS

This chapter presents the experimental results conducted for the LSAS and curve-based models, adopting the following organization into three sections. Section 6.1 details the software, test conditions, and performance metrics used for the experiments. Section 6.2 presents and discusses the experimental setup and results for the LSAS model, using a reduced set of experiments due to implementation limitations of LSAS. Finally, Section 6.3 presents and discusses the experimental setup and results for the curve-based models using a complete set of experiments with the Centerline and Radial models, a combination of both models, and a concluding with a comparison with the state-of-the-art.

6.1 Methodology

All experiments used the HM version 16.18 [48], modifying the HM to accommodate the LSAS and curve-based models while using the HM built-in tools to measure the bitrate, PSNR, and the encoding times of the test sequences. The HM modifications also enable an evaluation of the intra-prediction mode used for each PU and the measurement of the residual energy of predicted samples.

All tests used a computer cluster with eight nodes, each with two Intel Xeon E5520 processors, for a total of 8 processors and 16 threads per node, and 16 gigabytes of memory on each node. Each encoder instance ran as a dedicated single-threaded process, with a maximum of 1 encoder instance per available thread in the cluster, performing I/O operations through a dedicated gigabit connection with an external storage unit, thus minimizing outside interference as much as possible.

The JCTVC-AF1100 CTC for HEVC [57] defines a set of encoder configurations for evaluating HEVC encoder implementations. The CTC establishes a common benchmark for evaluating coding standards, ensuring the fair comparison among different works in state-of-the-art. The AF1100 CTC specifies the use of four encoding configurations using 8-bits per sample (All-Intra-8, Random-Access-8, Low-delay-B-8, and Low-delay-P-8) and four 10-bits per sample configurations (All-Intra-10, Random-Access-10, Low-delay-B-10, and Low-delay-P-10). The All-Intra configurations encode the entire video using only I-Frames, disregarding intra-frame prediction entirely. The Random-Access are the typical encoder configurations using a GOP with I-, P-, and B-Frames. Lastly, the Low-delay configurations always begin with one I-Frame, while subsequent frames use only P-Frames (Low-delay-P configurations) or only B-Frames (Low-delay-B configurations). The AF1100 CTC requires the use of four QP parameters with values 22, 27, 32, and 37.

The AF1100 test sequences cover various video characteristics, ranging from high-definition to low-definition videos, natural and synthetic images on the video frames, and high and low motion. Table 6.1 details each AF1100 test sequence. Video classes A1 and A2 correspond to high-definition videos with 4K UHD resolution and captured at 10-bit depth for increased image fidelity. The Class B videos use 1080p FHD resolution with a combination of 8- and 10-bit depth depending on the test sequence. Video classes C and D use low-definition videos with sub HD resolutions, while videos from class E use 720p HD resolution. Lastly, the class F videos consist mostly of textual content, such as computer screen recording or television broadcast with text legends.

Table 6.1 – HEVC AF1100 test sequences used for the experiments.

Class	Resolution	Sequence Name	Frame Count	Frame Rate (fps)	Bit Depth
Class A1	3840 × 2160	Tango2	294	60	10
		FoodMarket4	300	60	10
		Campfire	300	30	10
Class A2		CatRobot	300	60	10
		DaylightRoad2	300	60	10
		ParkRunning3	300	50	10
Class B	1920 × 1080	MarketPlace	600	60	10
		RitualDance	600	60	10
		Cactus	500	50	8
		BasketballDrive	500	50	8
		BQTerrace	600	60	8
Class C	832 × 480	BasketballDrill	500	50	8
		BQMall	600	60	8
		PartyScene	500	50	8
		RaceHorses	300	30	8
Class D	416 × 240	BasketballPass	500	50	8
		BQSquare	600	60	8
		BlowingBubbles	500	50	8
		RaceHorses	300	30	8
Class E	1280 × 720	FourPeople	600	60	8
		Johnny	600	60	8
		KristenAndSara	600	60	8
Class F	832 × 480	BasketballDrillText	500	50	8
	1920 × 1080	ArenaOfValor	600	60	8
	1280 × 720	SlideEditing	300	30	8
		SlideShow	500	20	8

The encoding configurations employ a specific set of the test sequence video classes:

- the All-Intra configurations use all test sequences from all video classes;
- the Random-Access configurations use all test sequences from video classes A1, A2, B, C, D, and F; and
- the Low-delay configurations use all test sequences from video classes B, C, D, E, and F.

This Thesis uses the following performance metrics for the evaluations:

- **PSNR and bitrate**—The HM built-in tools report the PSNR and the bitrate, measuring the encoding quality and the data requirements of each encoded test sequence, respectively;
- **Coding efficiency**—The BD-Rate measurement is the main metric used for evaluating the coding efficiency of the curve-based prediction models. The BD-Rate calculation uses the PSNR and bitrate measurements from the HM. Additionally, the results demonstrate the graphical comparison of the rate-distortion performance for some test sequences;
- **Use of intra-prediction modes and PU size**—The modifications made on the HM enable a detailed report indicating the intra-prediction mode used by each PU. The results consider the PU size representativeness, i.e., each PU of size $n \times n$ has a weight according to their area $A = n^2$; thus, measuring the number of samples encoded by each PU size and intra-prediction mode. This weighting is necessary since while there might be several 4×4 blocks in an encoded test sequence, a single 64×64 block has 256 times more samples than a 4×4 block;
- **Residual energy**—The residual energy evaluations consider each intra-predicted sample of the test sequences. Modifications made on the HM capture the residual information at encoding time, supplying an accurate measurement of the residual energy of all test sequences. The measurements consist of the SAD of predicted samples to their reference values from the source frame; and
- **Encoding and decoding times**—The HM provides an accurate measurement of the encoding and decoding times of each test sequence, enabling a comparison among encoding configurations and encoder/decoder implementations.

6.2 LSAS Model Evaluation

An accurate estimation of the LSAS bitstream requirements demands extensive modifications in the HEVC bitstream format, binarization, and CABAC implementations for dealing with the LSAS requirements; therefore, the results for the LSAS model should consider that encoder decisions do not consider the real bitstream requirements of LSAS. As discussed in Section 5.1, encoding the polynomial surface coefficients requires the insertion of much more information in the PUs than HEVC uses. The HEVC PU requires up to 11 bits, as shown in Section 5.2.4, while LSAS requires at least four times as much for encoding the polynomial coefficients. Thus, until a better transmission method for the polynomial surface coefficients exists, the bitstream requirements of LSAS makes this technique unfeasible for achieving a coding efficiency comparable to modern standards. Embedding the LSAS polynomials into the PUs is undergoing research, and so far, there is only a rough estimation of its bitrate requirements.

6.2.1 Experimental Setup

The HM modifications follow the same approach used in [36], replacing one of the 35 HEVC intra-prediction modes with an LSAS intra mode. According to Figure 5.1, the angular mode 32 was the least used among all test sequences. Therefore, the LSAS intra mode replaces the HEVC intra-prediction mode 32 for the experiments.

Figure 6.1 details the experimental setup for the LSAS model. The LSAS intra mode uses up to fourth-order polynomial surfaces, with each polynomial order representing a distinct encoder implementation. The experiments consider only a small subset of the AF1100 CTC due to the current limitations of the LSAS model regarding the necessary HEVC bitstream requirements, using only the All-Intra-8 encoding configuration with the four QP parameters specified in the CTC. The set of experiments for evaluation consider 520 distinct test scenarios, encompassing:

- 104 encoder configurations:
  - The All-Intra-8 encoding configuration with its 26 test sequence configurations.
  - The four QP parameters, applied to each test sequence configuration.
- 5 encoder implementations:
  - The baseline HEVC implementation used for comparisons.
  - The four HEVC encoder implementations using LSAS with polynomial orders ranging from 1 up to 4.

Due to the bitstream implementation limitations of the LSAS model, only the PSNR measurement is available for evaluating the encoding quality of the test scenarios. The results for the LSAS model also measure the use of intra-prediction modes and PU sizes considering the PU size representativeness.

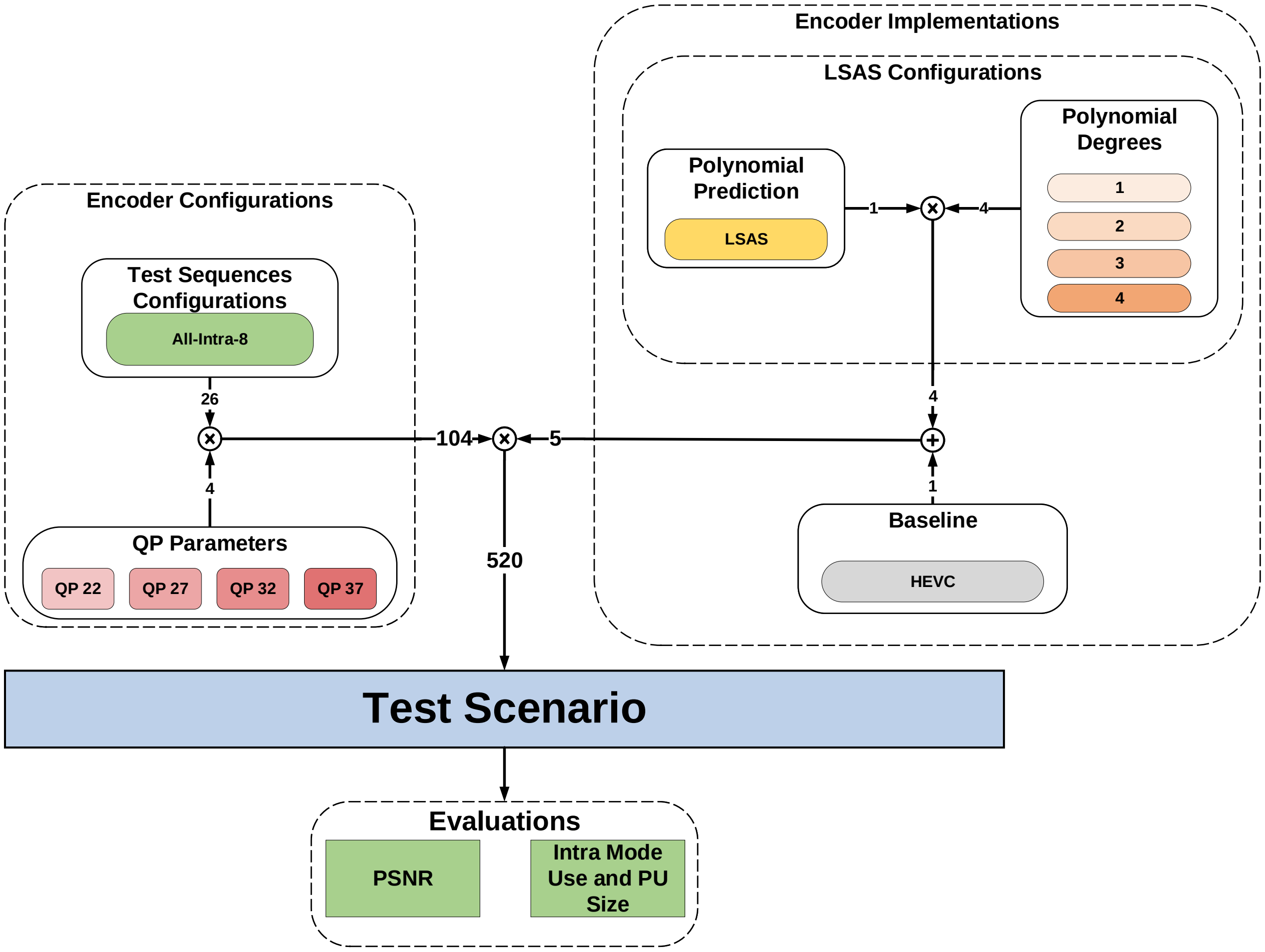


Figure 6.1 – LSAS model experimental setup. The experiments consider only a small subset of the AF1100 CTC due to the current limitations of the LSAS model regarding the necessary HEVC bitstream modifications.

6.2.2 PSNR Improvements

The improved prediction quality provided by the LSAS intra mode improves the PSNR of the encoded video sequences by reducing the residuals. Considering that the HEVC encoder quantizes the domain-transformed residuals, having fewer residuals translates into high-quality frame reconstruction since there is less information loss during the quantization steps of the encoding process. Table 6.2 shows improved PSNR for all test sequences, grouped by the average of each video class and polynomial order.

The LSAS prediction quality improves with higher-order polynomial surfaces. While the first-order provides a slight increase in PSNR, the subsequent PSNR improvements are substantial. It is important to mention that since there are no means to estimate the bitrate costs of LSAS, the encoder considers a PU using the LSAS intra mode to have the same bitrate costs like any other intra-prediction mode. This problem can mislead the encoder when selecting the prediction mode, as LSAS can falsely indicate improved coding efficiency

Table 6.2 – Average LSAS PSNR improvements grouped by video class for all test sequences.

	Polynomial order			
Video class	**1**	**2**	**3**	**4**
Class A1	0.19 dB	0.98 dB	1.79 dB	2.40 dB
Class A2	0.18 dB	0.97 dB	1.77 dB	2.35 dB
Class B	0.29 dB	1.05 dB	1.59 dB	1.96 dB
Class C	0.12 dB	0.63 dB	1.05 dB	1.39 dB
Class D	0.08 dB	0.53 dB	0.98 dB	1.33 dB
Class E	0.43 dB	1.19 dB	1.81 dB	2.25 dB
Class F	1.67 dB	1.96 dB	2.24 dB	2.45 dB
Average	0.42 dB	1.04 dB	1.60 dB	2.02 dB

since the actual bitstream size appears smaller than it should. Regardless of the correct bitrate requirements evaluation by the encoder, it is possible to evaluate the capabilities of LSAS from a pure prediction quality point of view.

6.2.3 Use of LSAS Intra Mode and PU Sizes

Table 6.3 shows that the HEVC encoder chooses the LSAS prediction mode most of the time, as it achieves better prediction quality than the standard HEVC intra-prediction modes.

Table 6.3 – LSAS intra mode use for up to fourth-order polynomial surfaces. The encoder chooses LSAS for most PUs due to the improved prediction quality.

Polynomial order	**HEVC intra modes**	**LSAS intra mode**
1	30.09%	69.91%
2	12.97%	87.03%
3	8.56%	91.44%
4	6.75%	93.25%

Figure 6.2 illustrates the PU sizes use for different LSAS polynomial orders. LSAS appears to offer better prediction quality when using 16 × 16 PUs, showing a considerable increase in the use 16 × 16 PUs when comparing the PU use distribution of the original HEVC encoded test sequences. The use of 16 × 16 PUs continue to increase for the higher-order polynomials, up to 64.71% with fourth-order polynomial surfaces, as detailed in Table 6.4. Meanwhile, employing LSAS reduced the use of large PUs, with the use of 64 × 64 and 32 × 32 PU sizes dropping by 83.40% and 93.10%, respectively, for LSAS with fourth-order polynomial surfaces. Meanwhile, smaller PUs were used more, with increases of 148.36%,

42.72%, and 16.47% for PU sizes 16 × 16, 8 × 8, and 4 × 4, respectively, also with fourth-order polynomials.

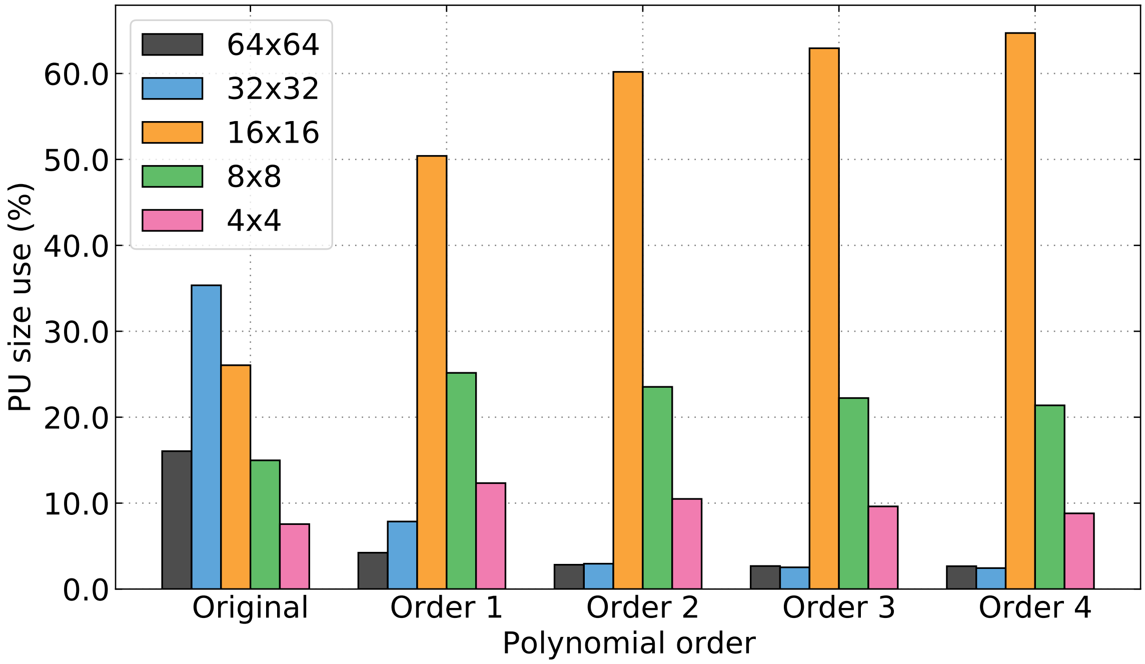


Figure 6.2 – LSAS PU size use. Large PU sizes have decreased in use, while 4 × 4, 8 × 8, and specially 16 × 16 PU sizes have increased.

Table 6.4 – Comparison of LSAS PU size. The 16×16 PUs have increased in use considerably with LSAS when compared to the original HEVC encoded test sequences.

	Original	Polynomial order			
PU size		**1**	**2**	**3**	**4**
64 × 64	16.05%	4.23%	2.83%	2.68%	2.66%
32 × 32	35.35%	7.86%	2.95%	2.53%	2.44%
16 × 16	26.05%	50.41%	60.19%	62.94%	64.71%
8 × 8	14.98%	25.16%	23.53%	22.23%	21.38%
4 × 4	7.56%	12.33%	10.49%	9.62%	8.81%

While the experiments do not restrict the PU size for LSAS, an interesting exploration would be to apply LSAS as a special prediction mode for 16 × 16 PUs, given their significant use in the encoded test sequences. Even with the hefty bitrate requirements for encoding the LSAS polynomials, the bitrate savings due to the fewer residuals of LSAS encoded PUs might offset the encoding penalty, improving coding efficiency.

6.3 Curve-Based Models Evaluation

Unlike the LSAS model, the curve-based models need only a few additional bits to encode the curve displacement value, making these models a much more feasible approach and the main contribution of this Thesis. The experiments shown in this section provide an in-depth evaluation of the curve-based models regarding coding efficiency, the use of curve-based intra modes and PU sizes, the residual energy of the test sequences, and the encoding and decoding times. Additionally, this section presents a combination of both curve-based models for improved coding efficiency, and a comparison of the curve-based models with the state-of-the-art.

6.3.1 Experimental Setup

Figure 6.3 illustrates the experimental setup for the curve-based models. The results explore all encoding configurations with the four QP parameters specified by the AF1100 CTC. The encoder configurations are the input of each encoder implementation. The number of curve displacements and bitstream encoding technique apply only to the Centerline and Radial curve-based models, while the baseline HEVC and the optimized mixed Centerline and Radial encoder implementations have a single configuration each.

The experiments consider the complete set of the AF1100 CTC testing scenarios complemented by the multiple curve-based models configurations. The set of experiments for evaluations employ 46992 distinct test scenarios, encompassing:

- 712 encoder configurations:
  - Both All-Intra encoding configurations, each with their 26 test sequences, for a total of 52 test sequence configurations.
  - Both Random-Access encoding configurations, each with their 23 test sequences, for a total of 46 test sequence configurations.
  - The four Low-delay encoding configurations, each with their 20 test sequences, for a total of 80 test sequence configurations.
  - The four QP parameters, applied to each test sequence configuration.
- 66 encoder implementations:
  - The baseline HEVC implementation used for comparisons.
  - The optimized mixed Centerline and Radial curve-based models encoder implementation.

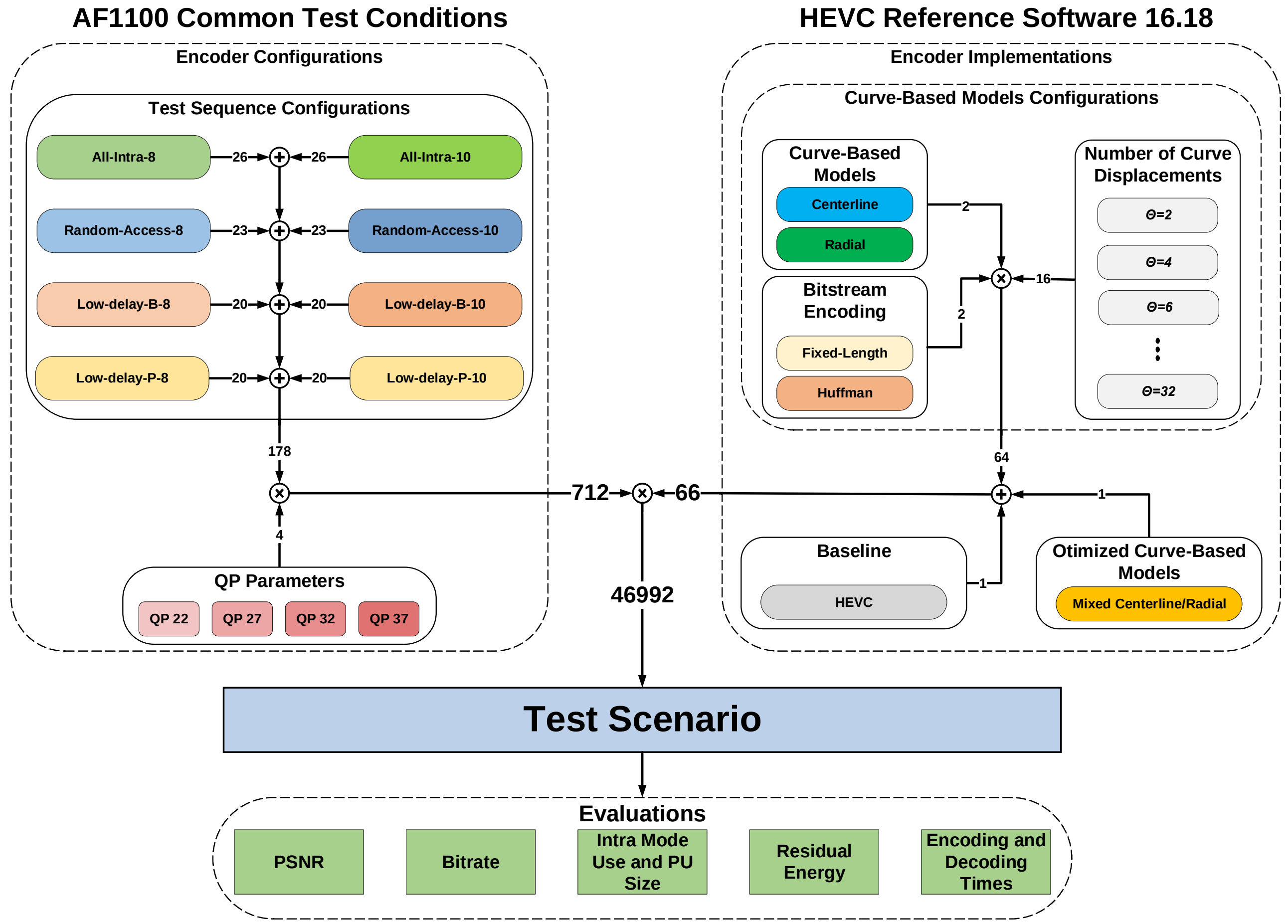


Figure 6.3 – Curve-based models experimental setup. The extensive number of test scenarios enables an in-depth evaluation of the curve-based models.

- 64 curve-based model encoder implementations:
  - ⋆ 16 ranges of curve displacement values, with $2 \leq \theta \leq 32$.
  - ⋆ Two bitstream encoding techniques: fixed-length and Huffman codes.
  - ⋆ Two curve-based models: Centerline and Radial, each using the 16 ranges of θ and the two bitstream encoding techniques.

The evaluations for each test scenario consider the PSNR and bitrate of each encoded sequence, enabling an evaluation of the coding efficiency of each encoder implementation. The experiments with the Centerline and Radial models use a varying number of curve displacements, identified by the variable θ, which is always an even number to retain symmetry on the applied curve displacements, where $\theta(m) = 2m$ and m is in the range $[1, 16]$. Thus, $\theta(1)$ adds two curve displacement values, i.e., $\omega = -1$ and $\omega = 1$. For each value of ω, the HEVC encoder has the default 35 intra-frame prediction modes plus the 33 angular intra-prediction modes with the corresponding ω. For large values of θ, the number of intra-prediction modes P available for the encoder increases quickly, with $P(\theta) = 35 + (33 \times \theta)$, where:

- 35 corresponds to default HEVC intra-prediction modes, i.e., Planar, DC, and the 33 angular modes; and
- $33 \times \theta$ corresponds to each curve displacement applied to an angular intra-prediction mode.

Considering $1 \leq m \leq 16$, this results in $\theta(m)$ values ranging from 2 to 32, corresponding to 101 up to 1091 intra-prediction modes for evaluation. Like MRL [8], the adoption of the curve-based intra modes for the HEVC intra-frame predictor creates a new set of prediction modes that considers the 33 angular prediction modes of HEVC and new modes using curve displacements. Therefore, the encoder must compute more prediction modes for rate-distortion evaluation. Using one additional reference line in MRL duplicates the encoding time since the encoder must evaluate the 33 HEVC intra-prediction modes with both prediction lines as a possibility. Likewise, when MRL uses three reference lines, the encoding time triplicates. Similarly, increasing the number of curve-displacement values for the encoder results in a linear increase in encoding times.

For the Centerline and Radial models, there are two approaches for encoding ω in the HEVC bitstream, as discussed in Section 5.2.4: (i) using fixed-length encoding; and (ii) using static Huffman codes. The coding efficiency results for the Centerline and Radial models present the BD-Rate analysis for both bitstream encoding approaches. However, due to the considerable advantages of Huffman codes, the remaining evaluations for intra-mode use and PU size, residual energy, and encoding and decoding times, present only the Huffman codes results.

All encoder decisions for the baseline and curve-based models consider only the HEVC Rate-Distortion Optimization (RDO) measurements to properly evaluate the curve-based models regarding their prediction quality and bitrate costs; i.e., the encoder computes the RDO for all intra-prediction modes, disregarding the optimizations offered by RMD. While it is understandable that this choice increases encoding times, it also provides a fair comparison of the HEVC intra-predictor with the curve-based models.

This extensive amount of scenarios enables an in-depth evaluation of how the curve-based models influence coding efficiency and the point where additional curve displacements may no longer offer significant benefits.

6.3.2 Coding Efficiency

Figures 6.4 and 6.5 illustrate the BD-Rate results for the Centerline model using fixed-length and Huffman encoding, respectively. For the All-Intra-8 and All-Intra-10 configurations, the adoption of curve-based intra modes shows BD-Rate reductions greater than 1% at $\theta = 2$, i.e., using only two values of curve displacements with $\omega = -1$ and $\omega = 1$. The BD-

Rate reduction is larger when the HEVC encoder has more curve displacements available with greater θ values, achieving $\approx$ 2% BD-Rate reduction for both All-Intra-8 and All-Intra-10 configurations. The Random-Access-8 and Random-Access-10 configurations also show BD-Rate reductions for all θ values, starting at $\approx$ 0.5% with θ = 2.

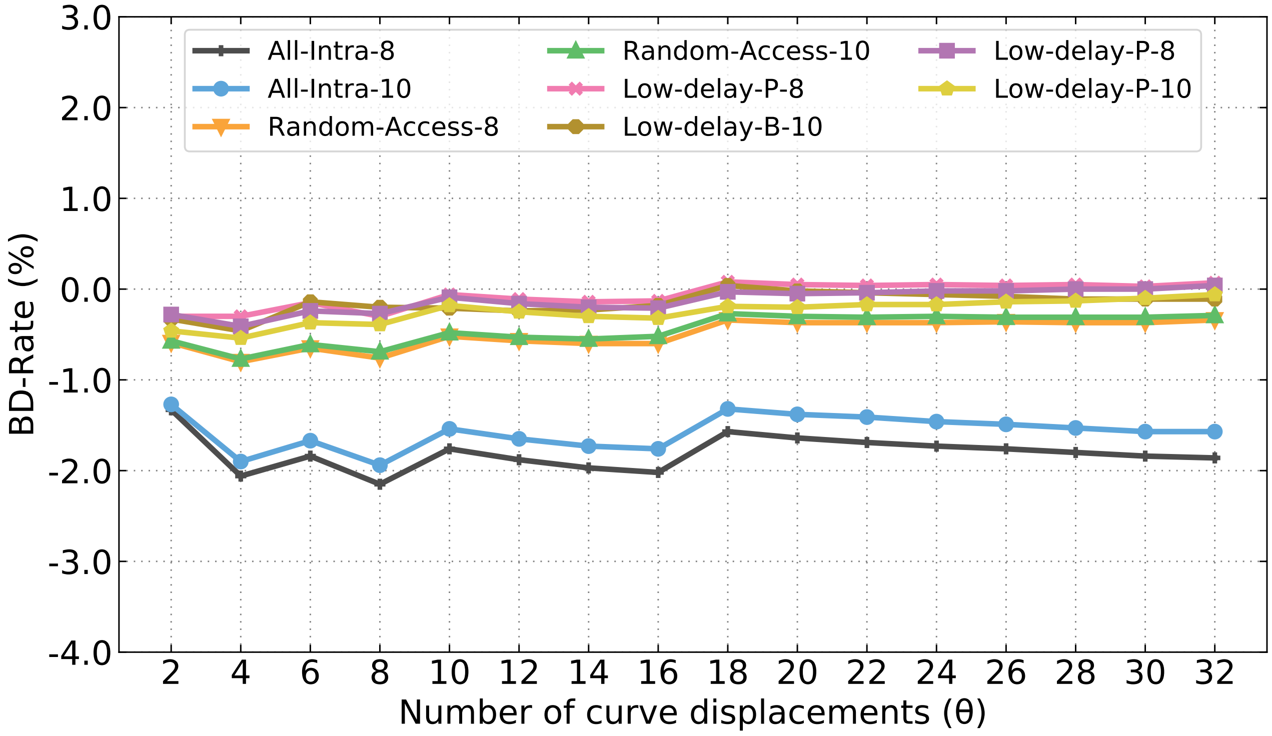


Figure 6.4 – Centerline model BD-Rate analysis with fixed-length encoding for θ ranging from 2 to 32, for all test configurations.

The Low-delay configurations illustrate, in Figure 6.4, the first limitations of fixed-length encoding since the BD-Rate compared to the baseline increases with $\theta > 16$; this occurs because, as θ increases for fixed-length coding, more bits are necessary to encode the curve displacement parameters. Since the Low-delay configurations have only one I-Frame in the entire encoded sequence, the benefits of the curve-based models are not as impactful as in the All-Intra-8, All-Intra-10, Random-Access-8, and Random-Access-10 configurations. The overhead introduced by encoding the curve parameter, in this case, diminishes the coding efficiency, as the Low-delay configurations barely use the curve-based intra modes. The Random-Access-8 and Random-Access-10 configurations also achieve inferior BD-Rate improvements for $\theta > 16$, albeit at a lesser extent than the Low-delay configurations, due to the overhead necessary to encode the curve displacements in the bitstream.

Another negative aspect of fixed-length encoding occurs for θ values that are not a power of 2. As Figure 6.4 illustrates for the All-Intra-8 and All-Intra-10 configurations, certain values of θ achieve inferior coding efficiency than previous θ values, e.g., θ = 6, θ = 10, and θ = 18. This behavior is in line with how fixed-length uses binary strings for encoding ω. For θ = 6, the unused binary strings are an overhead not offset by any encoding gains; however,

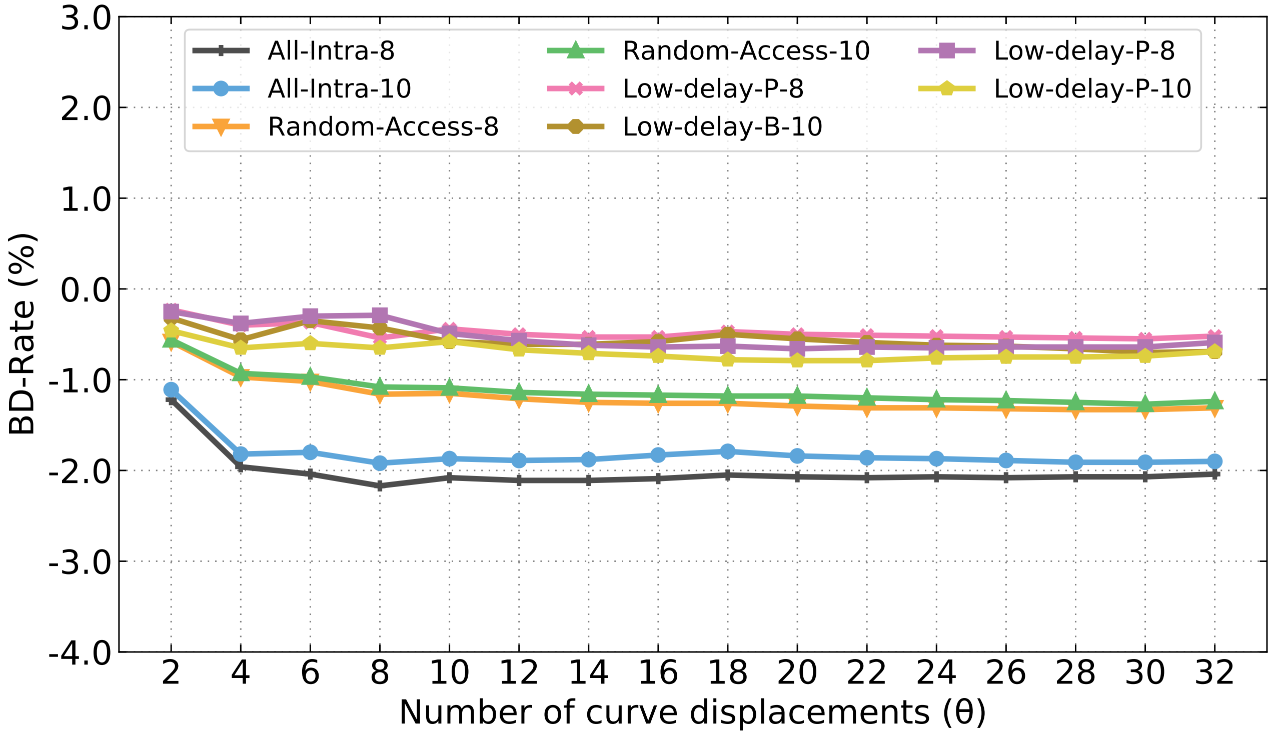


Figure 6.5 – Centerline model BD-Rate analysis with Huffman encoding for θ ranging from 2 to 32, for all test configurations.

increasing the number of available curve displacement parameter for the encoder to $\theta = 8$ introduces two new curve-based intra modes that offset the overhead of using 3 bits for encoding the 8 possible ω values in the range $[-4, 4]$. This fixed-length encoding behavior also affects, albeit to a smaller degree, the Random-Access-8, Random-Access-10, and the Low-delay configurations.

The more efficient Huffman encoding, illustrated in Figure 6.5 for the Centerline model, directly translates into improved video coding efficiency. The All-Intra-8 and All-Intra-10 configurations for the Centerline model, using Huffman coding, show slightly improved BD-Rate compared to the fixed-length encoding. However, Figure 6.5 displays significant benefits of Huffman encoding for the Random-Access-8, Random-Access-10 and Low-delay configurations with BD-Rate reductions for all θ values.

Figures 6.6 and 6.7 illustrate the BD-Rate results for the Radial model with fixed-length and Huffman encoding, respectively. The All-Intra-8 and All-Intra-10 configurations show similar results for the Centerline and Radial models using fixed-length encoding, at $\approx$ 2% BD-Rate reduction compared to the baseline. However, the more significant result occurs for the All-Intra-8 and All-Intra-10 configurations using Huffman codes in the Radial model, reaching a considerable improvement of $\approx$ 3% BD-Rate reduction with $\theta > 16$.

While the Radial model offers considerable improvements with the All-Intra-8 and All-Intra-10 configurations, its performance is somewhat inferior with Random-Access-8 and Random-Access-10 compared to the Centerline model. Results show a BD-Rate reduction

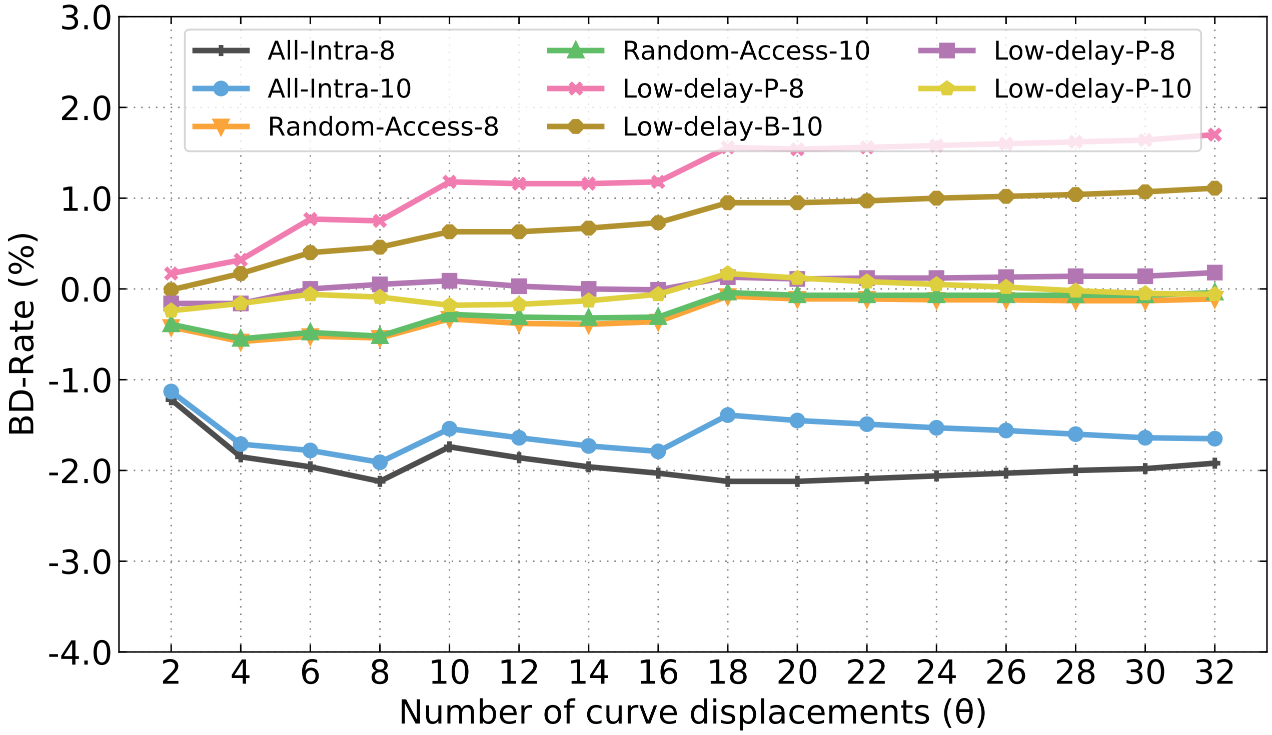


Figure 6.6 – Radial model BD-Rate analysis with fixed-length encoding for θ ranging from 2 to 32, for all test configurations.

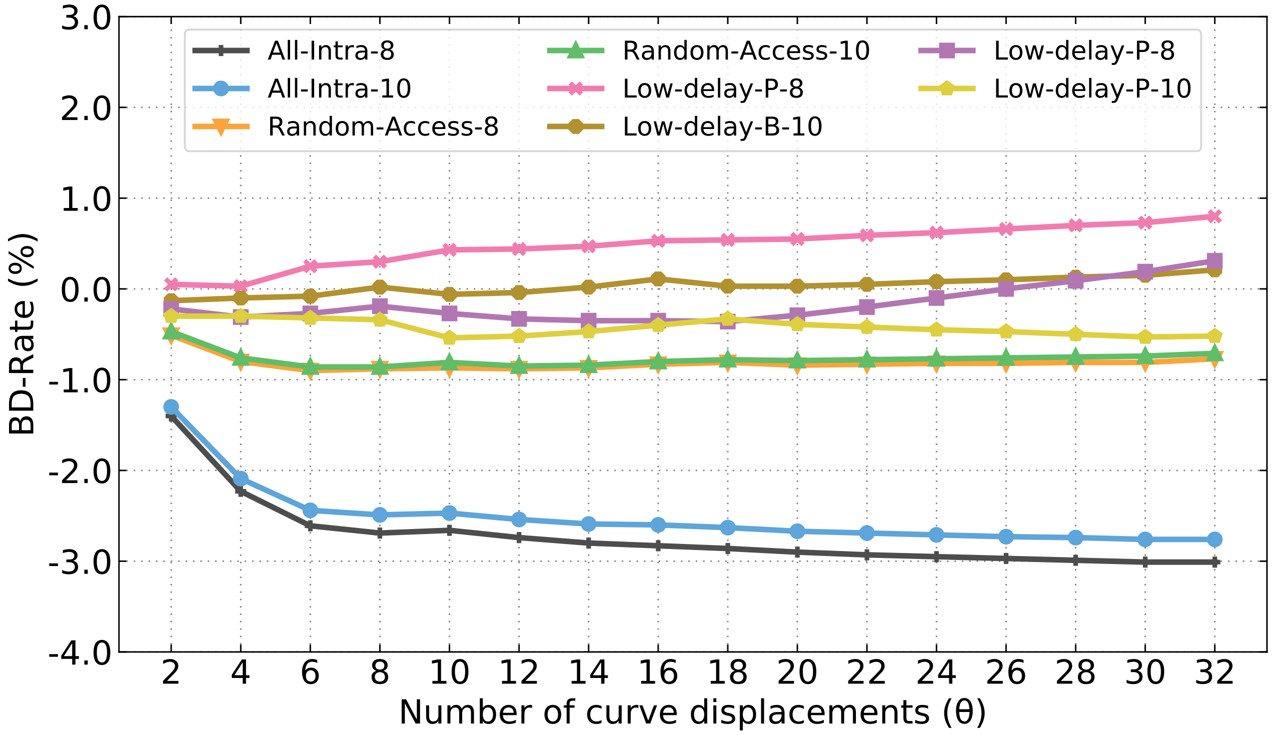


Figure 6.7 – Radial model BD-Rate analysis with Huffman encoding for θ ranging from 2 to 32, for all test configurations.

slightly above 1% and below 1% for the Centerline and Radial models, respectively. Likewise, the Low-delay configurations also show inferior coding efficiency for the Radial model compared to the Centerline model, depending on θ. The results of the Low-delay configurations are inferior to the baseline when the overhead of encoding the curve displacement parameter exceeds the coding efficiency gains of using the curve-based prediction models. Two reasons imply in this efficiency penalty: the curve-based prediction models (i) mainly benefit intra-frame prediction, but the Low-delay configurations benefit mostly from inter-frame prediction and (ii) increase the PU bitstream size. Therefore, the benefits of the curve-based prediction models are much less significant for the Low-delay configurations.

Even though the Centerline model achieves better results for the Random-Access-8, Random-Access-10, and the Low-delay configurations, the Radial model shows better results for the All-Intra-8 and All-Intra-10 configurations. At each PB, the encoder makes a localized decision about which prediction mode to use, considering only the rate-distortion metric. Each prediction model results in an entirely different reconstructed frame, changing the encoder decisions at the spatial level inside a frame during intra-frame prediction and at the temporal level between frames during inter-frame prediction. The encoder cannot analyze the entire coding flow before making a local decision due to computation and memory requirements; thus, some differences can occur among different coding techniques.

Tables 6.5 and 6.6 provide a detailed report of all test configurations, grouped by each video class, for the Centerline and Radial models, respectively. Regardless of the coding efficiency differences between the Centerline and Radial models among the encoding configurations, both show improved coding efficiency in most situations, especially with the All-Intra-8 and All-Intra-10 configurations. Considering the overall better performance with the Huffman codes, Tables 6.5 and 6.6 only illustrate the results for the curve-based models when using Huffman for encoding the bitstream elements. Meanwhile, Figures 6.8 and 6.9 illustrate the rate-distortion curves for some test sequences.

While the All-Intra-8 and All-Intra-10 configurations for both curve-based models show an overall improvement for all video classes, one noticeable set of results occur for the class F videos. Even with Random-Access-8, Random-Access-10, and the Low-delay configurations, the more significant BD-Rate improvements occur with class F videos. These results reveal the good efficiency of the curve-based prediction models for encoding videos with textual content. Section 6.3.3 further discusses and illustrates this efficiency of the curve-based models, explaining the BD-Rate results attained with class F videos.

Figures 6.5 and 6.7 evidence that the coding efficiency results have a saturation point, after which further encoding gains are insignificant, as the BD-Rate improvements with $\theta > 8$ occur at a much-reduced rate than it does for a lower θ values. This saturation point indicates a good optimization strategy for the curve-based prediction models since it should be possible to limit θ while keeping good BD-Rate improvements. Likewise, a combination of the Centerline and Radial models can also offer more possibilities for increasing coding efficiency.

Table 6.5 – Summary of BD-Rate results with the Centerline model for all tested configurations and θ values.

All-Intra-8

Sequence	θ															
	2	**4**	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**	**22**	**24**	**26**	**28**	**30**	**32**
Class A1	-0.93%	-1.83%	-2.11%	-2.32%	-2.29%	-2.37%	-2.43%	-2.47%	-2.46%	-2.51%	-2.55%	-2.57%	-2.60%	-2.63%	-2.65%	-2.65%
Class A2	-1.46%	-2.24%	-2.38%	-2.47%	-2.39%	-2.39%	-2.36%	-2.31%	-2.28%	-2.29%	-2.29%	-2.26%	-2.24%	-2.24%	-2.21%	-2.17%
Class B	-0.97%	-1.53%	-1.67%	-1.77%	-1.69%	-1.70%	-1.68%	-1.64%	-1.61%	-1.62%	-1.61%	-1.59%	-1.58%	-1.56%	-1.53%	-1.49%
Class C	-1.38%	-2.25%	-2.29%	-2.50%	-2.39%	-2.46%	-2.50%	-2.52%	-2.49%	-2.54%	-2.56%	-2.57%	-2.59%	-2.62%	-2.63%	-2.62%
Class D	-1.41%	-2.31%	-2.31%	-2.55%	-2.44%	-2.49%	-2.52%	-2.53%	-2.50%	-2.53%	-2.55%	-2.57%	-2.56%	-2.57%	-2.58%	-2.56%
Class E	-1.54%	-1.86%	-1.90%	-1.69%	-1.59%	-1.49%	-1.36%	-1.21%	-1.18%	-1.16%	-1.09%	-1.03%	-0.97%	-0.91%	-0.85%	-0.74%
Class F	-1.00%	-1.80%	-1.80%	-1.99%	-1.88%	-1.93%	-1.95%	-1.95%	-1.91%	-1.96%	-1.97%	-1.99%	-2.00%	-2.03%	-2.04%	-2.02%
Overall	-1.22%	-1.96%	-2.04%	-2.17%	-2.08%	-2.11%	-2.11%	-2.09%	-2.05%	-2.07%	-2.08%	-2.07%	-2.08%	-2.07%	-2.07%	-2.04%

All-Intra-10

Sequence	θ															
	2	**4**	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**	**22**	**24**	**26**	**28**	**30**	**32**
Class A1	-0.77%	-1.53%	-1.67%	-1.83%	-1.82%	-1.89%	-1.92%	-1.92%	-1.89%	-1.96%	-2.00%	-2.04%	-2.10%	-2.14%	-2.17%	-2.20%
Class A2	-1.29%	-2.01%	-2.04%	-2.10%	-2.06%	-2.05%	-2.01%	-1.93%	-1.89%	-1.92%	-1.91%	-1.92%	-1.91%	-1.90%	-1.90%	-1.87%
Class B	-0.85%	-1.39%	-1.42%	-1.51%	-1.47%	-1.47%	-1.44%	-1.38%	-1.34%	-1.38%	-1.38%	-1.37%	-1.39%	-1.38%	-1.38%	-1.35%
Class C	-1.28%	-2.16%	-2.12%	-2.33%	-2.27%	-2.33%	-2.36%	-2.36%	-2.32%	-2.40%	-2.43%	-2.48%	-2.51%	-2.56%	-2.60%	-2.61%
Class D	-1.33%	-2.23%	-2.16%	-2.41%	-2.34%	-2.40%	-2.42%	-2.41%	-2.37%	-2.43%	-2.46%	-2.48%	-2.52%	-2.54%	-2.57%	-2.57%
Class E	-1.48%	-1.77%	-1.70%	-1.48%	-1.41%	-1.30%	-1.16%	-0.98%	-0.93%	-0.93%	-0.88%	-0.84%	-0.79%	-0.75%	-0.69%	-0.62%
Class F	-0.89%	-1.70%	-1.60%	-1.82%	-1.73%	-1.78%	-1.80%	-1.78%	-1.74%	-1.79%	-1.83%	-1.86%	-1.90%	-1.94%	-1.97%	-1.97%
Overall	-1.11%	-1.82%	-1.80%	-1.92%	-1.87%	-1.89%	-1.88%	-1.83%	-1.79%	-1.84%	-1.86%	-1.87%	-1.89%	-1.91%	-1.91%	-1.90%

Random-Access-8

Sequence	θ															
	2	**4**	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**	**22**	**24**	**26**	**28**	**30**	**32**
Class A1	-0.34%	-0.50%	-0.55%	-0.54%	-0.51%	-0.52%	-0.51%	-0.48%	-0.50%	-0.51%	-0.49%	-0.47%	-0.46%	-0.44%	-0.43%	-0.38%
Class A2	-0.59%	-0.60%	-0.65%	-0.57%	-0.61%	-0.58%	-0.53%	-0.45%	-0.44%	-0.44%	-0.39%	-0.34%	-0.29%	-0.26%	-0.20%	-0.13%
Class B	-0.39%	-0.68%	-0.75%	-0.83%	-0.85%	-0.90%	-0.92%	-0.92%	-0.90%	-0.94%	-0.94%	-0.94%	-0.93%	-0.93%	-0.94%	-0.91%
Class C	-0.74%	-1.28%	-1.30%	-1.50%	-1.49%	-1.58%	-1.65%	-1.69%	-1.67%	-1.72%	-1.74%	-1.77%	-1.81%	-1.84%	-1.85%	-1.85%
Class D	-0.70%	-1.23%	-1.27%	-1.49%	-1.45%	-1.53%	-1.59%	-1.63%	-1.63%	-1.68%	-1.70%	-1.71%	-1.74%	-1.75%	-1.77%	-1.76%
Class F	-0.68%	-1.42%	-1.47%	-1.82%	-1.79%	-1.92%	-2.02%	-2.10%	-2.12%	-2.19%	-2.23%	-2.27%	-2.30%	-2.34%	-2.38%	-2.38%
Overall	-0.58%	-0.97%	-1.02%	-1.16%	-1.15%	-1.21%	-1.25%	-1.26%	-1.26%	-1.29%	-1.31%	-1.31%	-1.32%	-1.33%	-1.33%	-1.31%

Random-Access-10

Sequence	θ															
	2	**4**	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**	**22**	**24**	**26**	**28**	**30**	**32**
Class A1	-0.29%	-0.42%	-0.44%	-0.36%	-0.39%	-0.39%	-0.36%	-0.30%	-0.32%	-0.34%	-0.32%	-0.29%	-0.27%	-0.26%	-0.24%	-0.18%
Class A2	-0.51%	-0.60%	-0.66%	-0.56%	-0.58%	-0.56%	-0.52%	-0.44%	-0.43%	-0.43%	-0.40%	-0.36%	-0.33%	-0.29%	-0.25%	-0.19%
Class B	-0.39%	-0.63%	-0.69%	-0.80%	-0.78%	-0.83%	-0.84%	-0.83%	-0.81%	-0.84%	-0.84%	-0.84%	-0.84%	-0.86%	-0.85%	-0.82%
Class C	-0.74%	-1.20%	-1.28%	-1.44%	-1.45%	-1.53%	-1.58%	-1.60%	-1.63%	-1.69%	-1.70%	-1.71%	-1.74%	-1.76%	-1.78%	-1.77%
Class D	-0.69%	-1.22%	-1.21%	-1.38%	-1.41%	-1.50%	-1.56%	-1.59%	-1.59%	-1.65%	-1.66%	-1.69%	-1.71%	-1.73%	-1.75%	-1.75%
Class F	-0.70%	-1.37%	-1.42%	-1.68%	-1.72%	-1.82%	-1.89%	-1.93%	-1.07%	-1.27%	-1.46%	-1.63%	-1.81%	-1.98%	-2.17%	-2.31%
Overall	-0.56%	-0.93%	-0.97%	-1.08%	-1.09%	-1.15%	-1.17%	-1.17%	-1.02%	-1.08%	-1.11%	-1.14%	-1.17%	-1.21%	-1.23%	-1.24%

Low-delay-B-8

Sequence	θ															
	2	**4**	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**	**22**	**24**	**26**	**28**	**30**	**32**
Class B	0.01%	-0.33%	-0.07%	-0.69%	-0.14%	-0.24%	-0.31%	-0.36%	-0.24%	-0.27%	-0.28%	-0.27%	-0.26%	-0.27%	-0.26%	-0.23%
Class C	-0.19%	-0.31%	-0.35%	-0.37%	-0.39%	-0.43%	-0.45%	-0.44%	-0.43%	-0.45%	-0.47%	-0.47%	-0.47%	-0.47%	-0.47%	-0.44%
Class D	-0.13%	-0.23%	-0.28%	-0.19%	-0.23%	-0.26%	-0.27%	-0.25%	-0.26%	-0.29%	-0.31%	-0.30%	-0.30%	-0.31%	-0.31%	-0.27%
Class E	-0.60%	-0.68%	-0.68%	-0.69%	-0.82%	-0.83%	-0.82%	-0.78%	-0.77%	-0.80%	-0.82%	-0.81%	-0.83%	-0.83%	-0.84%	-0.82%
Class F	-0.38%	-0.55%	-0.62%	-0.74%	-0.78%	-0.86%	-0.91%	-0.93%	-0.80%	-0.86%	-0.88%	-0.93%	-0.96%	-0.99%	-1.00%	-1.01%
Overall	-0.23%	-0.40%	-0.37%	-0.54%	-0.44%	-0.50%	-0.53%	-0.53%	-0.47%	-0.50%	-0.51%	-0.52%	-0.53%	-0.54%	-0.55%	-0.52%

Low-delay-B-10

Sequence	θ															
	2	**4**	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**	**22**	**24**	**26**	**28**	**30**	**32**
Class B	-0.50%	-0.79%	-0.04%	-0.09%	-0.66%	-0.64%	-0.59%	-0.52%	-0.24%	-0.35%	-0.44%	-0.52%	-0.61%	-0.69%	-0.77%	-0.82%
Class C	-0.25%	-0.37%	-0.35%	-0.40%	-0.42%	-0.47%	-0.50%	-0.50%	-0.47%	-0.51%	-0.50%	-0.51%	-0.52%	-0.52%	-0.54%	-0.51%
Class D	-0.21%	-0.28%	-0.39%	-0.32%	-0.33%	-0.37%	-0.38%	-0.37%	-0.33%	-0.36%	-0.38%	-0.39%	-0.40%	-0.40%	-0.41%	-0.39%
Class E	-0.31%	-0.72%	-0.62%	-0.68%	-0.74%	-0.76%	-0.75%	-0.71%	-0.68%	-0.69%	-0.68%	-0.68%	-0.66%	-0.65%	-0.62%	-0.59%
Class F	-0.31%	-0.60%	-0.51%	-0.82%	-0.76%	-0.81%	-0.84%	-0.84%	-0.89%	-0.95%	-0.99%	-1.01%	-1.05%	-1.06%	-1.10%	-1.10%
Overall	-0.32%	-0.56%	-0.35%	-0.43%	-0.58%	-0.61%	-0.61%	-0.58%	-0.50%	-0.55%	-0.59%	-0.62%	-0.63%	-0.66%	-0.70%	-0.69%

Low-delay-P-8

Sequence	θ															
	2	**4**	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**	**22**	**24**	**26**	**28**	**30**	**32**
Class B	0.02%	-0.34%	-0.10%	-0.05%	-0.19%	-0.37%	-0.53%	-0.65%	-0.40%	-0.42%	-0.40%	-0.40%	-0.38%	-0.37%	-0.36%	-0.31%
Class C	-0.29%	-0.29%	-0.23%	-0.22%	-0.47%	-0.52%	-0.53%	-0.52%	-0.53%	-0.56%	-0.57%	-0.58%	-0.56%	-0.57%	-0.57%	-0.55%
Class D	-0.17%	-0.18%	-0.10%	-0.07%	-0.29%	-0.33%	-0.33%	-0.31%	-0.37%	-0.38%	-0.38%	-0.36%	-0.35%	-0.35%	-0.33%	-0.29%
Class E	-0.57%	-0.67%	-0.60%	-0.62%	-0.87%	-0.86%	-0.83%	-0.76%	-0.91%	-0.93%	-0.93%	-0.92%	-0.91%	-0.89%	-0.89%	-0.85%
Class F	-0.37%	-0.50%	-0.57%	-0.61%	-0.82%	-0.90%	-0.96%	-0.98%	-1.06%	-1.10%	-1.10%	-1.10%	-1.12%	-1.11%	-1.14%	-1.11%
Overall	-0.25%	-0.38%	-0.30%	-0.29%	-0.49%	-0.57%	-0.62%	-0.64%	-0.63%	-0.66%	-0.64%	-0.65%	-0.64%	-0.64%	-0.64%	-0.59%

Low-delay-P-10

Sequence	θ															
	2	**4**	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**	**22**	**24**	**26**	**28**	**30**	**32**
Class B	-0.65%	-0.97%	-0.84%	-0.62%	-0.55%	-0.67%	-0.78%	-0.85%	-1.09%	-1.07%	-0.99%	-0.93%	-0.87%	-0.81%	-0.76%	-0.67%
Class C	-0.37%	-0.48%	-0.45%	-0.53%	-0.47%	-0.53%	-0.54%	-0.54%	-0.53%	-0.56%	-0.56%	-0.57%	-0.57%	-0.58%	-0.58%	-0.55%
Class D	-0.23%	-0.31%	-0.33%	-0.38%	-0.35%	-0.40%	-0.41%	-0.41%	-0.38%	-0.42%	-0.42%	-0.42%	-0.43%	-0.43%	-0.42%	-0.40%
Class E	-0.53%	-0.85%	-0.87%	-0.88%	-0.92%	-0.93%	-0.94%	-0.90%	-0.83%	-0.85%	-0.85%	-0.84%	-0.86%	-0.85%	-0.84%	-0.81%
Class F	-0.49%	-0.62%	-0.53%	-0.89%	-0.71%	-0.85%	-0.95%	-1.02%	-0.98%	-1.03%	-1.03%	-1.05%	-1.06%	-1.08%	-1.09%	-1.06%
Overall	-0.46%	-0.65%	-0.60%	-0.65%	-0.58%	-0.67%	-0.71%	-0.74%	-0.78%	-0.79%	-0.79%	-0.76%	-0.75%	-0.75%	-0.74%	-0.69%

Table 6.6 – Summary of BD-Rate results with the Radial model for all tested configurations and θ values.

All-Intra-8

Sequence	θ															
	2	**4**	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**	**22**	**24**	**26**	**28**	**30**	**32**
Class A1	-1.36%	-2.05%	-2.45%	-2.63%	-2.65%	-2.78%	-2.88%	-2.96%	-3.05%	-3.12%	-3.16%	-3.21%	-3.25%	-3.29%	-3.34%	-3.36%
Class A2	-1.50%	-2.20%	-2.54%	-2.59%	-2.59%	-2.64%	-2.66%	-2.65%	-2.65%	-2.67%	-2.67%	-2.66%	-2.66%	-2.66%	-2.66%	-2.63%
Class B	-0.93%	-1.42%	-1.69%	-1.73%	-1.71%	-1.75%	-1.77%	-1.76%	-1.79%	-1.81%	-1.81%	-1.80%	-1.80%	-1.80%	-1.80%	-1.77%
Class C	-1.74%	-2.93%	-3.44%	-3.57%	-3.54%	-3.67%	-3.77%	-3.85%	-3.91%	-3.98%	-4.04%	-4.09%	-4.14%	-4.19%	-4.24%	-4.27%
Class D	-1.88%	-3.00%	-3.45%	-3.66%	-3.59%	-3.70%	-3.79%	-3.85%	-3.92%	-3.98%	-4.01%	-4.05%	-4.09%	-4.12%	-4.16%	-4.17%
Class E	-1.21%	-1.72%	-1.97%	-1.89%	-1.84%	-1.82%	-1.77%	-1.69%	-1.62%	-1.62%	-1.59%	-1.56%	-1.53%	-1.51%	-1.48%	-1.43%
Class F	-1.25%	-2.31%	-2.75%	-2.77%	-2.73%	-2.85%	-2.95%	-3.02%	-3.05%	-3.11%	-3.15%	-3.19%	-3.23%	-3.27%	-3.31%	-3.33%
Overall	-1.40%	-2.23%	-2.61%	-2.69%	-2.66%	-2.74%	-2.80%	-2.83%	-2.86%	-2.90%	-2.93%	-2.95%	-2.97%	-2.99%	-3.01%	-3.01%

All-Intra-10

Sequence	θ															
	2	**4**	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**	**22**	**24**	**26**	**28**	**30**	**32**
Class A1	-1.13%	-1.70%	-2.04%	-2.17%	-2.18%	-2.29%	-2.37%	-2.42%	-2.49%	-2.56%	-2.60%	-2.65%	-2.69%	-2.74%	-2.79%	-2.81%
Class A2	-1.32%	-1.96%	-2.26%	-2.28%	-2.27%	-2.31%	-2.32%	-2.30%	-2.30%	-2.32%	-2.31%	-2.30%	-2.30%	-2.29%	-2.29%	-2.26%
Class B	-0.85%	-1.29%	-1.53%	-1.53%	-1.51%	-1.55%	-1.56%	-1.54%	-1.53%	-1.55%	-1.55%	-1.55%	-1.55%	-1.54%	-1.54%	-1.52%
Class C	-1.68%	-2.84%	-3.34%	-3.45%	-3.43%	-3.55%	-3.65%	-3.71%	-3.79%	-3.86%	-3.91%	-3.96%	-4.01%	-4.05%	-4.10%	-4.13%
Class D	-1.85%	-2.97%	-3.41%	-3.58%	-3.54%	-3.65%	-3.73%	-3.78%	-3.84%	-3.90%	-3.94%	-3.97%	-4.01%	-4.05%	-4.08%	-4.10%
Class E	-1.14%	-1.61%	-1.80%	-1.68%	-1.62%	-1.58%	-1.52%	-1.42%	-1.34%	-1.33%	-1.30%	-1.26%	-1.23%	-1.20%	-1.16%	-1.11%
Class F	-1.17%	-2.23%	-2.64%	-2.67%	-2.61%	-2.73%	-2.81%	-2.87%	-2.91%	-2.96%	-3.00%	-3.03%	-3.06%	-3.10%	-3.13%	-3.14%
Overall	-1.30%	-2.09%	-2.44%	-2.49%	-2.47%	-2.54%	-2.59%	-2.60%	-2.63%	-2.67%	-2.69%	-2.71%	-2.73%	-2.74%	-2.76%	-2.76%

Random-Access-8

Sequence	θ															
	2	**4**	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**	**22**	**24**	**26**	**28**	**30**	**32**
Class A1	-0.11%	-0.15%	-0.09%	-0.01%	0.01%	0.03%	0.08%	0.16%	0.19%	0.20%	0.23%	0.27%	0.31%	0.34%	0.38%	0.45%
Class A2	-0.11%	0.08%	0.17%	0.33%	0.37%	0.43%	0.52%	0.64%	0.79%	0.81%	0.86%	0.92%	0.97%	1.02%	1.08%	1.16%
Class B	-0.27%	-0.35%	-0.38%	-0.34%	-0.34%	-0.34%	-0.31%	-0.25%	-0.22%	-0.23%	-0.20%	-0.18%	-0.15%	-0.13%	-0.11%	-0.05%
Class C	-0.78%	-1.29%	-1.47%	-1.49%	-1.52%	-1.55%	-1.56%	-1.54%	-1.58%	-1.62%	-1.63%	-1.65%	-1.66%	-1.67%	-1.68%	-1.66%
Class D	-0.82%	-1.31%	-1.51%	-1.54%	-1.52%	-1.56%	-1.58%	-1.57%	-1.58%	-1.63%	-1.65%	-1.66%	-1.68%	-1.70%	-1.71%	-1.70%
Class F	-0.84%	-1.49%	-1.78%	-1.81%	-1.85%	-1.92%	-1.96%	-1.97%	-1.98%	-2.04%	-2.08%	-2.11%	-2.14%	-2.18%	-2.21%	-2.21%
Overall	-0.51%	-0.80%	-0.90%	-0.88%	-0.87%	-0.88%	-0.87%	-0.83%	-0.81%	-0.84%	-0.83%	-0.82%	-0.82%	-0.81%	-0.81%	-0.77%

Random-Access-10

Sequence	θ															
	2	**4**	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**	**22**	**24**	**26**	**28**	**30**	**32**
Class A1	-0.08%	-0.03%	0.06%	0.16%	0.22%	0.23%	0.29%	0.38%	0.44%	0.45%	0.48%	0.51%	0.54%	0.57%	0.60%	0.65%
Class A2	-0.03%	0.07%	0.13%	0.26%	0.29%	0.35%	0.44%	0.58%	0.67%	0.71%	0.78%	0.84%	0.91%	0.98%	1.04%	1.13%
Class B	-0.25%	-0.33%	-0.36%	-0.32%	-0.27%	-0.29%	-0.26%	-0.20%	-0.17%	-0.17%	-0.15%	-0.13%	-0.11%	-0.09%	-0.07%	-0.03%
Class C	-0.75%	-1.24%	-1.46%	-1.49%	-1.46%	-1.52%	-1.55%	-1.53%	-1.56%	-1.59%	-1.60%	-1.60%	-1.61%	-1.62%	-1.63%	-1.61%
Class D	-0.80%	-1.32%	-1.51%	-1.55%	-1.51%	-1.58%	-1.62%	-1.61%	-1.57%	-1.60%	-1.61%	-1.62%	-1.63%	-1.64%	-1.65%	-1.64%
Class F	-0.79%	-1.42%	-1.66%	-1.82%	-1.76%	-1.85%	-1.90%	-1.91%	-1.95%	-2.00%	-2.02%	-2.05%	-2.08%	-2.10%	-2.13%	-2.13%
Overall	-0.47%	-0.76%	-0.86%	-0.86%	-0.81%	-0.85%	-0.84%	-0.80%	-0.78%	-0.79%	-0.78%	-0.77%	-0.76%	-0.75%	-0.74%	-0.71%

Low-delay-B-8

Sequence	θ															
	2	**4**	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**	**22**	**24**	**26**	**28**	**30**	**32**
Class B	0.26%	0.05%	0.54%	0.41%	0.80%	0.70%	0.62%	0.56%	0.65%	0.64%	0.67%	0.69%	0.71%	0.74%	0.76%	0.81%
Class C	0.01%	0.09%	0.21%	0.33%	0.43%	0.46%	0.51%	0.58%	0.62%	0.63%	0.67%	0.70%	0.74%	0.78%	0.81%	0.88%
Class D	0.08%	0.23%	0.36%	0.49%	0.54%	0.60%	0.68%	0.79%	0.81%	0.83%	0.87%	0.92%	0.96%	1.01%	1.06%	1.13%
Class E	-0.08%	0.02%	0.06%	0.24%	0.25%	0.34%	0.46%	0.60%	0.47%	0.49%	0.54%	0.59%	0.64%	0.69%	0.74%	0.82%
Class F	-0.13%	-0.24%	-0.05%	-0.04%	0.01%	0.03%	0.08%	0.15%	0.10%	0.11%	0.14%	0.18%	0.21%	0.25%	0.29%	0.35%
Overall	0.05%	0.03%	0.25%	0.30%	0.43%	0.44%	0.47%	0.53%	0.54%	0.55%	0.59%	0.62%	0.66%	0.70%	0.73%	0.80%

Low-delay-B-10

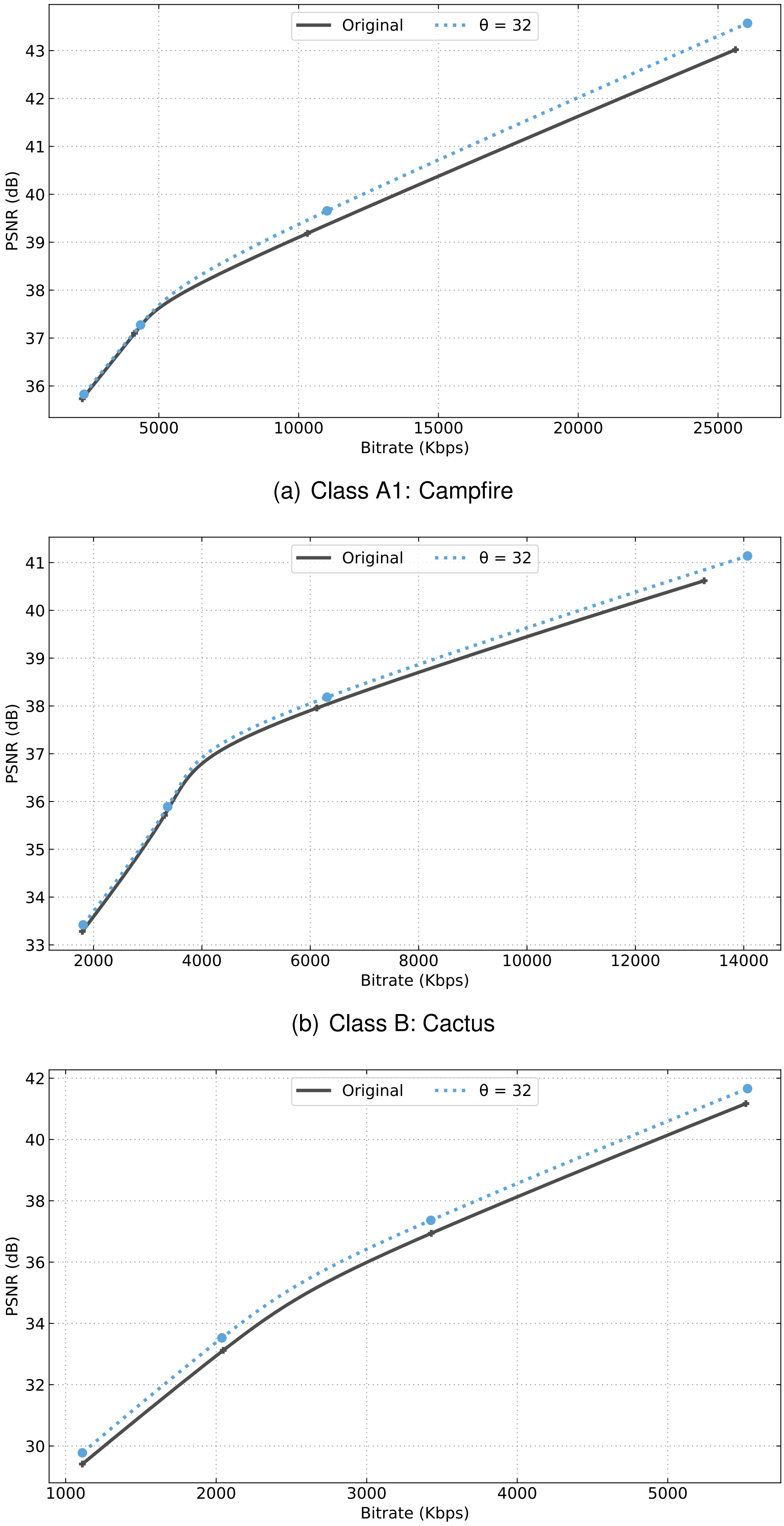
Sequence	θ															
	2	**4**	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**	**22**	**24**	**26**	**28**	**30**	**32**
Class B	-0.14%	-0.03%	0.15%	0.40%	-0.06%	-0.02%	0.05%	0.15%	-0.09%	-0.07%	-0.01%	0.04%	0.09%	0.14%	0.20%	0.28%
Class C	-0.07%	-0.08%	-0.13%	-0.07%	0.06%	0.05%	0.07%	0.13%	0.13%	0.13%	0.16%	0.19%	0.21%	0.24%	0.27%	0.33%
Class D	-0.09%	0.00%	0.04%	0.09%	0.11%	0.13%	0.18%	0.26%	0.30%	0.29%	0.32%	0.34%	0.36%	0.38%	0.41%	0.46%
Class E	-0.11%	-0.01%	-0.16%	-0.08%	-0.05%	0.01%	0.10%	0.23%	0.16%	0.16%	0.19%	0.22%	0.25%	0.28%	0.31%	0.37%
Class F	-0.22%	-0.37%	-0.37%	-0.38%	-0.36%	-0.34%	-0.29%	-0.21%	-0.28%	-0.32%	-0.32%	-0.33%	-0.34%	-0.35%	-0.35%	-0.33%
Overall	-0.13%	-0.10%	-0.08%	0.02%	-0.06%	-0.04%	0.02%	0.11%	0.03%	0.03%	0.05%	0.08%	0.10%	0.13%	0.15%	0.21%

Low-delay-P-8

Sequence	θ															
	2	**4**	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**	**22**	**24**	**26**	**28**	**30**	**32**
Class B	-0.23%	-0.17%	-0.27%	0.10%	0.09%	-0.01%	-0.08%	-0.12%	-0.31%	-0.15%	0.04%	0.23%	0.42%	0.61%	0.79%	1.01%
Class C	-0.20%	-0.29%	-0.28%	-0.23%	-0.34%	-0.37%	-0.38%	-0.35%	-0.35%	-0.31%	-0.25%	-0.19%	-0.13%	-0.06%	7.22E-07	0.09%
Class D	-0.10%	-0.15%	-0.14%	-0.10%	-0.13%	-0.16%	-0.16%	-0.13%	-0.12%	-0.18%	-0.20%	-0.23%	-0.26%	-0.29%	-0.32%	-0.32%
Class E	-0.23%	-0.34%	-0.25%	-0.30%	-0.32%	-0.38%	-0.41%	-0.41%	-0.20%	-0.16%	-0.09%	-0.02%	0.05%	0.12%	0.19%	0.29%
Class F	-0.34%	-0.62%	-0.42%	-0.55%	-0.75%	-0.80%	-0.82%	-0.81%	-0.81%	-0.68%	-0.53%	-0.37%	-0.22%	-0.07%	0.09%	0.27%
Overall	-0.22%	-0.31%	-0.27%	-0.19%	-0.27%	-0.33%	-0.35%	-0.35%	-0.36%	-0.29%	-0.20%	-0.10%	0.00%	0.09%	0.19%	0.31%

Low-delay-P-10

Sequence	θ															
	2	**4**	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**	**22**	**24**	**26**	**28**	**30**	**32**
Class B	-0.39%	-0.23%	-0.21%	-0.31%	-0.86%	-0.75%	-0.62%	-0.46%	-0.18%	-0.30%	-0.38%	-0.46%	-0.54%	-0.62%	-0.70%	-0.75%
Class C	-0.21%	-0.30%	-0.30%	-0.29%	-0.36%	-0.36%	-0.33%	-0.28%	-0.35%	-0.39%	-0.39%	-0.39%	-0.40%	-0.40%	-0.40%	-0.37%
Class D	-0.17%	-0.20%	-0.16%	-0.17%	-0.19%	-0.21%	-0.20%	-0.17%	-0.19%	-0.22%	-0.21%	-0.20%	-0.19%	-0.18%	-0.17%	-0.12%
Class E	-0.23%	-0.20%	-0.34%	-0.19%	-0.42%	-0.37%	-0.30%	-0.20%	-0.14%	-0.19%	-0.21%	-0.23%	-0.24%	-0.26%	-0.28%	-0.26%
Class F	-0.47%	-0.58%	-0.65%	-0.71%	-0.76%	-0.80%	-0.81%	-0.80%	-0.77%	-0.83%	-0.86%	-0.89%	-0.91%	-0.94%	-0.97%	-0.96%
Overall	-0.30%	-0.30%	-0.32%	-0.34%	-0.54%	-0.52%	-0.47%	-0.40%	-0.33%	-0.39%	-0.42%	-0.45%	-0.47%	-0.50%	-0.53%	-0.52%

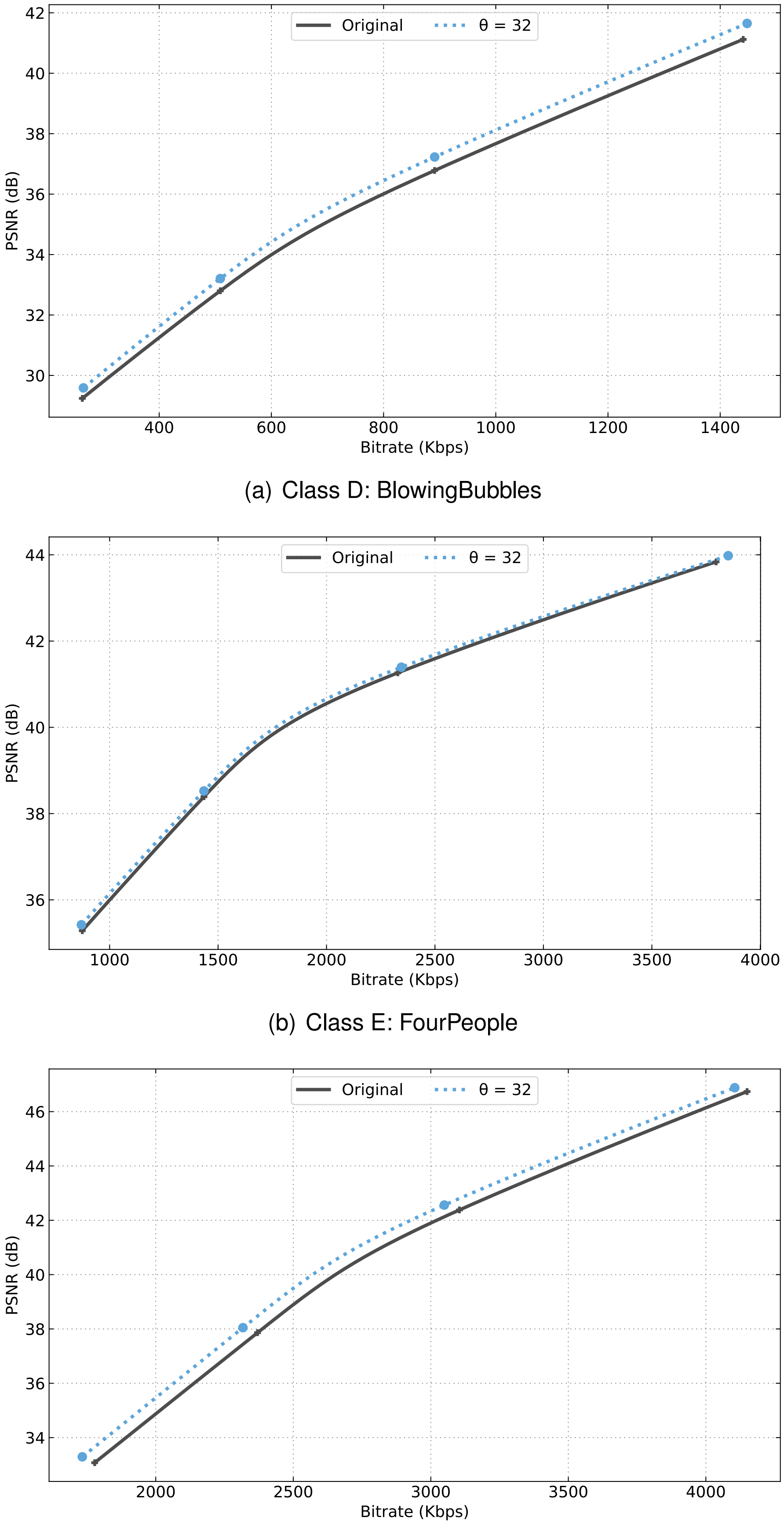


(a) Class A1: Campfire

(b) Class B: Cactus

(c) Class C: PartyScene

Figure 6.8 – Radial model rate-distortion overview for video classes A1, B and C of the AF1100 CTC, using All-Intra-8 configuration.



(a) Class D: BlowingBubbles

(b) Class E: FourPeople

(c) Class F: SlideEditing

Figure 6.9 – Radial model rate-distortion overview for video classes D, E and F of the AF1100 CTC, using All-Intra-8 configuration.

6.3.3 Use of Curve-Based Intra Modes

The intensity of the generated curve depends on ω; the higher its value, the more accentuated is the arc of the curve. Evaluating the use of curve-based intra modes considers the case of θ = 32, where the encoder has the full range of curve displacement values available. The analysis uses the All-Intra-8 configuration to avoid the influence of inter-prediction from the Random-Access and Low-delay configurations, allowing a better evaluation of the intra-prediction behavior when using the curve-based intra modes. While the results illustrate only the All-Intra-8 configuration case, the use of curve-based intra modes follows nearly the same distribution for the All-Intra-10 configuration. Like in the LSAS results, the use of curve-based intra modes consider the block representativeness.

Figures 6.10 and 6.11 illustrate the distribution of block sizes and their associated ω values. Values of $\omega < 0$ correspond to curves displaced to the left of the prediction direction, and values of $\omega > 0$ correspond to curves displaced to the right of the prediction direction. Meanwhile, ω = 0 are the default straight lines used by the HEVC intra-predictor without a curve displacement. The analysis of the curve-based intra modes reveals that the HEVC encoder favors low-accentuated curves, i.e., lower absolute values for ω. This range of ω values for intra-prediction provides an important insight into the behavior of the curve-based models, as it indicates a point after which additional curve displacements offer diminishing gains, i.e., at around θ = 8, or ω in the range $[-4, 4]$. This value for θ, after which the use of curve-based intra modes appears to saturate, follows the same θ value where the BD-Rate gains also reached a saturation point, as shown in Section 6.3.2.

There is one interesting detail in Figures 6.10 and 6.11 regarding larger PBs: the results of both charts show a spike for curve-based intra modes at $\omega = -16$ for block sizes 32×32 and 64×64. This behavior is present for the Centerline and Radial models, with a first increase in the use of curve-based intra modes already visible at $\omega = -15$. An interpretation of this behavior is that larger blocks may achieve better results with larger values of ω. Currently, the curve displacement value does not consider the dimensions of the predicted block, meaning that a displacement value of one, i.e., $\omega = -1$ or $\omega = 1$ applies the same amount of sample displacement for sample prediction in a 64×64 block as it does in a 4×4 block. Although a small curve displacement may affect a small PB significantly, the same displacement might not have as much impact for larger PBs. Although this Thesis does not evaluate a scenario on which curve displacements scale with the PB size, it should be possible to achieve further encoding improvements using $|\omega > 16|$ values for larger PBs, with a scaling logic of the curve displacement tied to the block dimensions.

The lower values of ω used for the curve-based intra modes show that most curves employ a soft arc. Figure 6.12 illustrates this effect for the Kimono, SlideEditing, and BQTerrace test sequences, representing a single encoded frame, with an overlay color map signaling

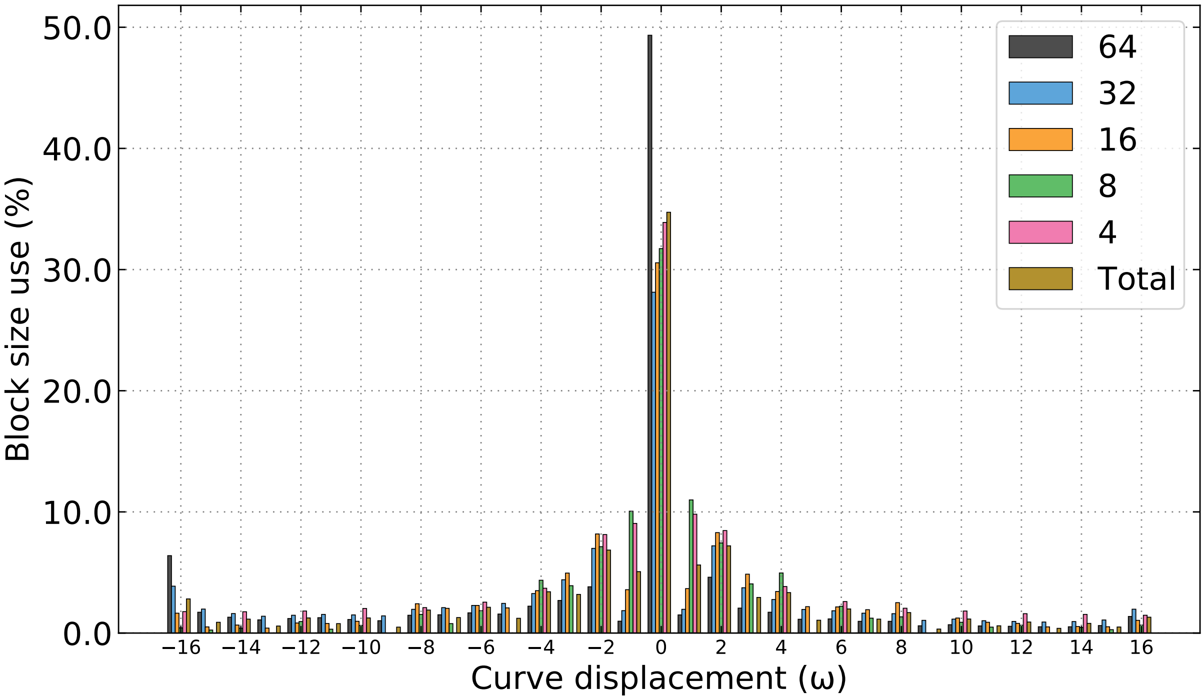


Figure 6.10 – Average use of Centerline intra modes for all test sequences with $\theta = 32$. Low-accentuated curve displacements have higher use, indicating that more curve displacements offer diminishing benefits for coding efficiency after a certain point.

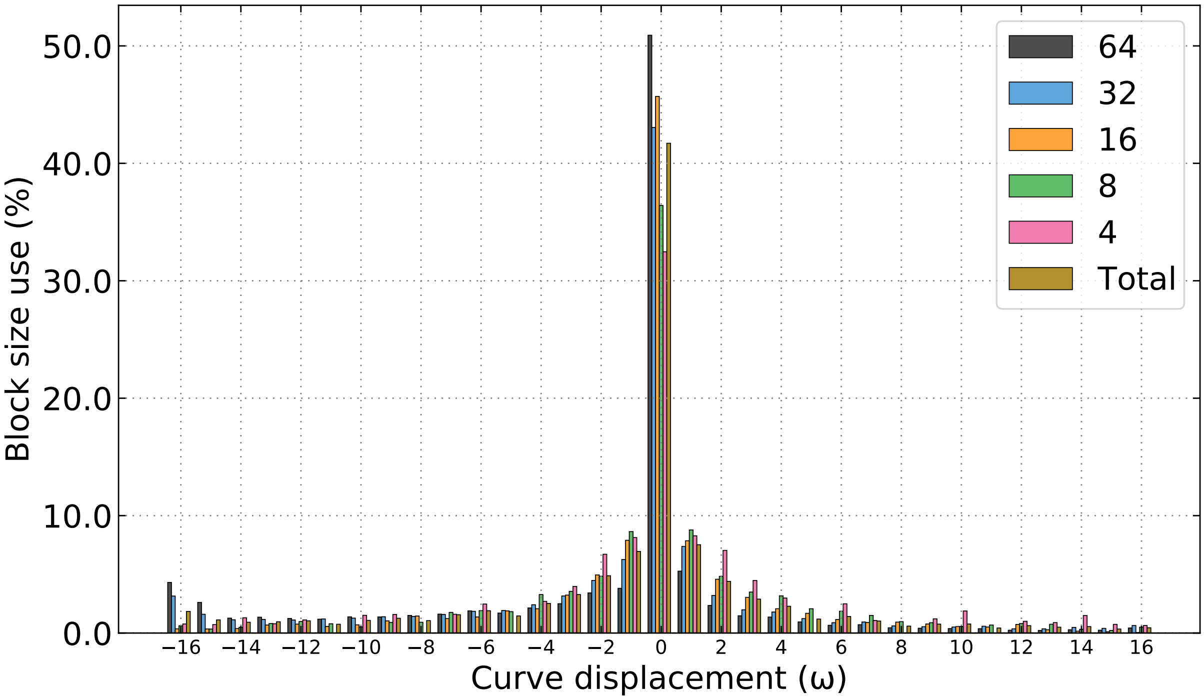
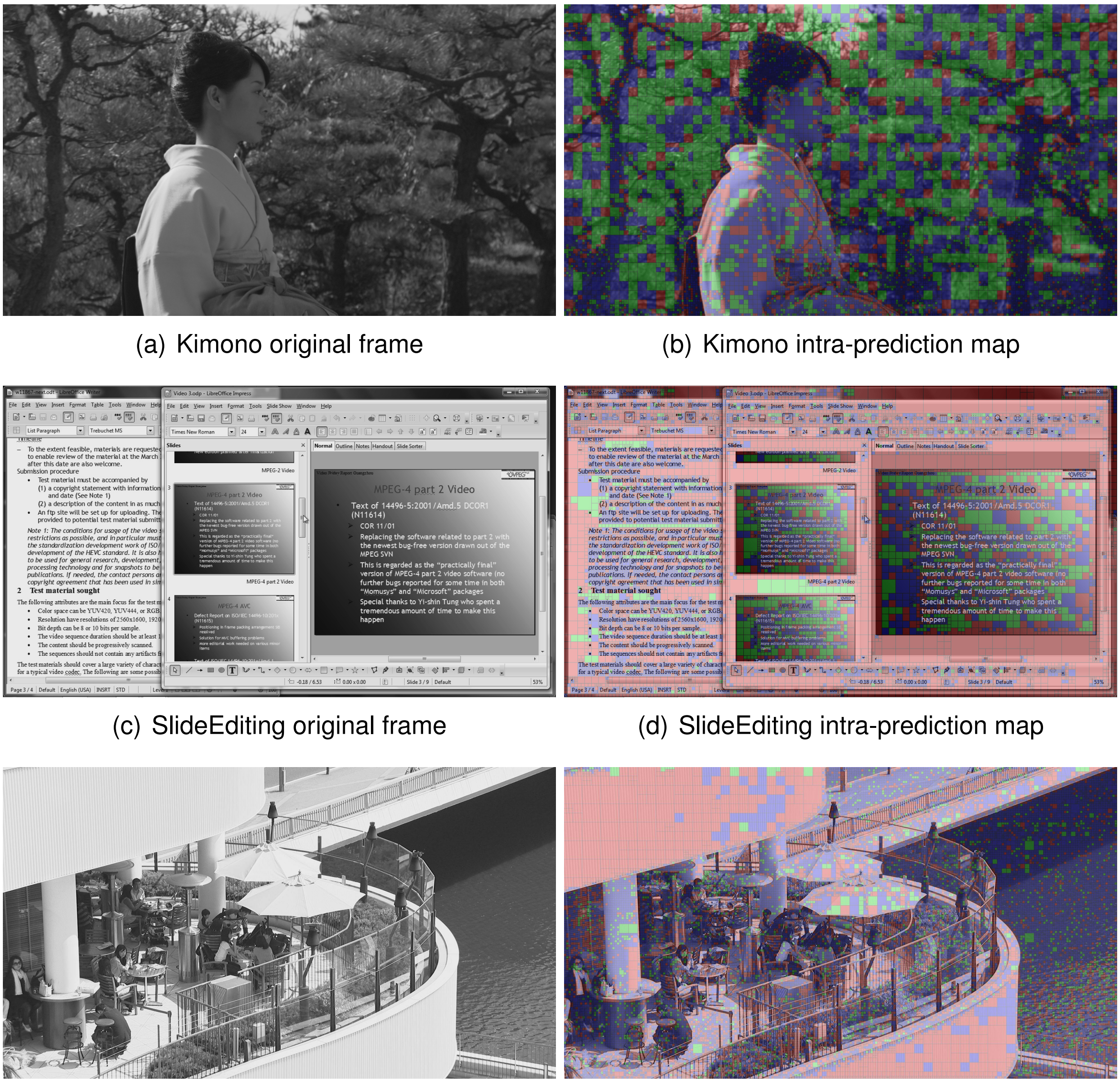


Figure 6.11 – Average use of Radial intra modes for all test sequences with $\theta = 32$. Like the Centerline model, low-accentuated curve displacements have higher use, also indicating the same saturation point of ω values in the range $[-4, 4]$.

the intra-prediction modes used for each PB. The blue blocks correspond to a PB using curve-based intra-prediction with $\omega \neq 0$. Figure 6.12(b) reveals that most of the PBs using curve-


(a) Kimono original frame

(b) Kimono intra-prediction map

(c) SlideEditing original frame

(d) SlideEditing intra-prediction map

(e) BQTerrace original frame

(f) BQTerrace intra-prediction map

🟩 Planar/DC modes. 🟥 HEVC angular modes. 🟦 Curve-based intra modes.

Figure 6.12 – Comparison of HEVC intra-prediction with the curve-based models for the Kimono, SlideEditing, and BQTerrace test sequences. Green blocks correspond to Planar/DC modes; Red blocks to HEVC angular modes; Blue blocks to curve-based intra modes. Blue blocks concentrate on textures like hair, tree branches, plants, water ripples, clothing, and text.

based intra modes concentrate at the texture regions with naturally occurring curves like tree branches, clothing, and hair. Similarly, Figure 6.12(f) displays that curve-based intra modes concentrate around the soft shadows in the pillars and parasols, clothing, plants, and water ripples. Finally, Figure 6.12(d) exposes that text contents comprise most of the curve-based intra modes, which better represent the high-contrasting curve features of text symbols; this is in line with the good BD-Rate results seen in Section 6.3.2 for the class F videos since it appears that curve-based intra modes can improve textual content encoding significantly.

6.3.4 Residual Energy

The accumulated residual energy of video sequences encoded with curve-based intra modes should be smaller than the default HEVC encoding due to a better prediction. Figures 6.13 and 6.14 illustrate the measured residual energy for the Centerline and Radial curve modes, respectively, as the average for each encoding configuration, while Tables 6.7 and 6.8 report the measured residual energy results. In both curve-based models, the results show reduced residual energy for all configurations. A more significant residual energy reduction occurs with the All-Intra configurations since these modes rely purely on intra-frame prediction. Meanwhile, the Random-Access and Low-delay configurations also show reductions in residual energy, albeit less than in All-Intra-8 and All-Intra-10; an expected result since most PUs in these configurations adopt inter-frame prediction.

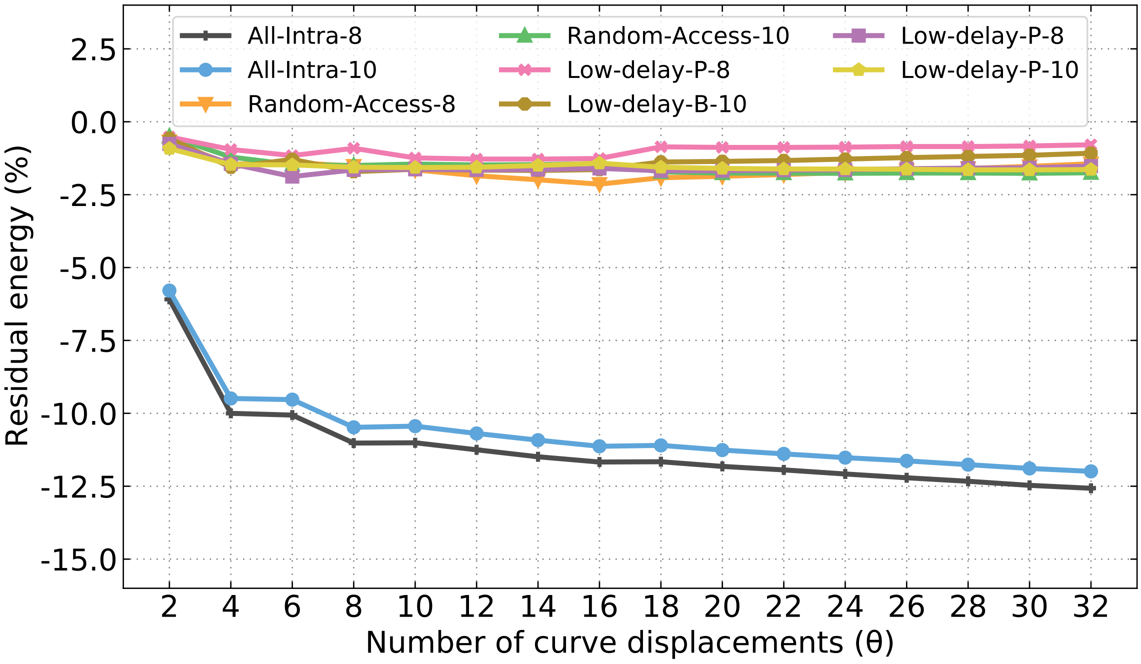


Figure 6.13 – Average residual energy reduction for the Centerline model for all test configurations.

The observed residual energy reductions corroborate the BD-Rate improvements shown in Section 6.3.2. The charts in Figures 6.13 and 6.14 follow nearly the same distribution of the BD-Rate charts in Figures 6.5 and 6.7, respectively, especially for the All-Intra configurations. For the Centerline model, both All-Intra results in Figure 6.5 show a minimal improvement from θ = 4 to θ = 6, with a more expressive BD-Rate reduction from θ = 6 to θ = 8; similarly, the All-Intra results in Figure 6.13 shows the same trend when comparing both θ differences for the residual energy. The same behavior occurs for the other points in the residual energy results compared to their BD-Rate counterpoints, both for the Centerline and Radial models. This behavior shows that the improved coding efficiency seen in Section

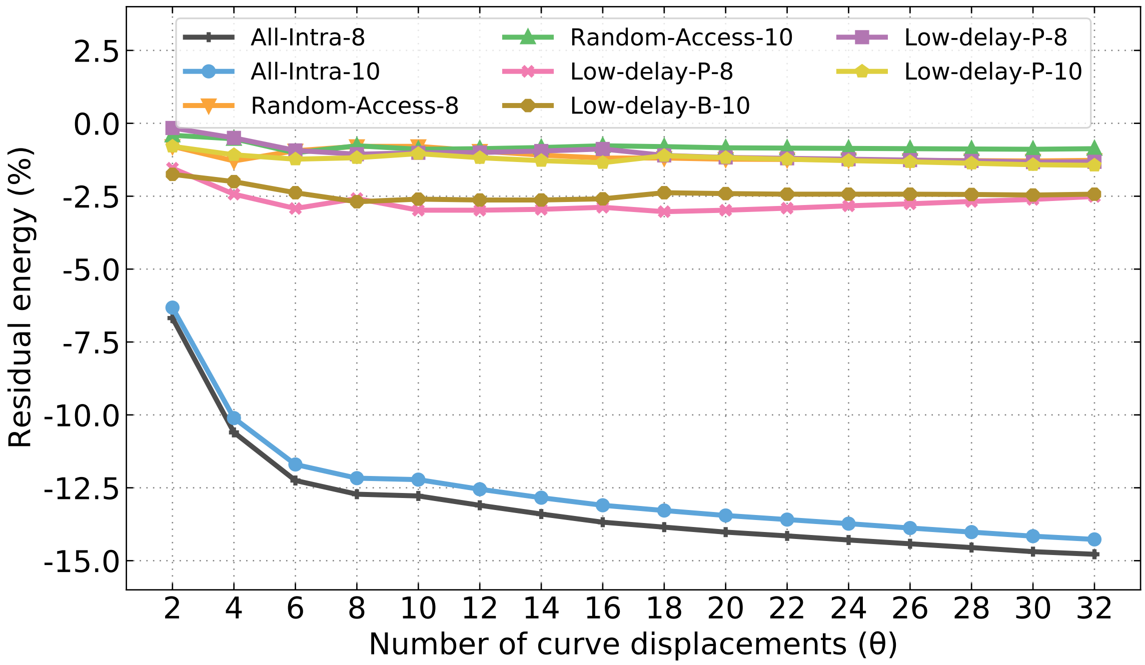


Figure 6.14 – Average residual energy reduction for the Radial model for all test configurations.

Table 6.7 – Centerline model residual energy averaged per configuration for all test sequences and θ values.

θ	All-Intra-8	All-Intra-10	Random-Access-8	Random-Access-10	Low-delay-B-8	Low-delay-B-10	Low-delay-P-8	Low-delay-P-10
2	-6.09%	-5.79%	-0.68%	-0.50%	-0.52%	-0.57%	-0.76%	-0.92%
4	-10.00%	-9.49%	-1.46%	-1.21%	-0.95%	-1.56%	-1.44%	-1.47%
6	-10.06%	-9.53%	-1.40%	-1.47%	-1.16%	-1.29%	-1.88%	-1.48%
8	-11.02%	-10.48%	-1.52%	-1.50%	-0.91%	-1.71%	-1.65%	-1.56%
10	-11.01%	-10.44%	-1.65%	-1.45%	-1.24%	-1.63%	-1.63%	-1.57%
12	-11.25%	-10.69%	-1.85%	-1.48%	-1.28%	-1.66%	-1.65%	-1.56%
14	-11.49%	-10.92%	-1.99%	-1.47%	-1.28%	-1.67%	-1.64%	-1.50%
16	-11.67%	-11.13%	-2.14%	-1.43%	-1.26%	-1.64%	-1.60%	-1.42%
18	-11.66%	-11.10%	-1.92%	-1.72%	-0.86%	-1.38%	-1.68%	-1.56%
20	-11.82%	-11.26%	-1.87%	-1.76%	-0.88%	-1.36%	-1.68%	-1.60%
22	-11.94%	-11.39%	-1.81%	-1.76%	-0.88%	-1.33%	-1.66%	-1.62%
24	-12.08%	-11.52%	-1.74%	-1.77%	-0.87%	-1.28%	-1.65%	-1.62%
26	-12.21%	-11.63%	-1.67%	-1.76%	-0.85%	-1.23%	-1.61%	-1.63%
28	-12.33%	-11.76%	-1.60%	-1.76%	-0.85%	-1.19%	-1.59%	-1.65%
30	-12.47%	-11.89%	-1.53%	-1.77%	-0.83%	-1.15%	-1.58%	-1.65%
32	-12.57%	-11.99%	-1.44%	-1.75%	-0.79%	-1.08%	-1.52%	-1.64%

6.3.2 originates from the reduced residual energy achieved with the curve-based intra modes, reinforcing the notion that the curve-based models improve intra-frame prediction quality.

Another interesting observation about the residual energy results is the saturation point mentioned earlier. Like the results shown in Sections 6.3.2 and 6.3.3, the residual energy reductions follow the same trend of diminished improvements when $\theta > 8$, further reinforcing the notion that the curve-based models offer more significant improvements with $\theta \leq 8$.

Table 6.8 – Radial model residual energy averaged per configuration for all test sequences and θ values.

θ	All-Intra-8	All-Intra-10	Random-Access-8	Random-Access-10	Low-delay-B-8	Low-delay-B-10	Low-delay-P-8	Low-delay-P-10
2	-6.68%	-6.32%	-0.79%	-0.41%	-1.54%	-1.75%	-0.16%	-0.79%
4	-10.60%	-10.11%	-1.29%	-0.53%	-2.43%	-2.00%	-0.50%	-1.08%
6	-12.25%	-11.70%	-0.94%	-1.01%	-2.92%	-2.38%	-0.93%	-1.23%
8	-12.72%	-12.17%	-0.79%	-0.77%	-2.58%	-2.69%	-1.06%	-1.18%
10	-12.78%	-12.22%	-0.79%	-0.89%	-2.98%	-2.60%	-1.01%	-1.04%
12	-13.10%	-12.55%	-0.95%	-0.87%	-2.98%	-2.63%	-1.00%	-1.18%
14	-13.40%	-12.84%	-1.09%	-0.83%	-2.95%	-2.63%	-0.95%	-1.28%
16	-13.68%	-13.10%	-1.19%	-0.77%	-2.88%	-2.59%	-0.88%	-1.35%
18	-13.85%	-13.28%	-1.19%	-0.80%	-3.03%	-2.38%	-1.10%	-1.11%
20	-14.02%	-13.45%	-1.23%	-0.84%	-2.98%	-2.41%	-1.17%	-1.18%
22	-14.15%	-13.59%	-1.24%	-0.85%	-2.91%	-2.43%	-1.20%	-1.23%
24	-14.29%	-13.73%	-1.25%	-0.86%	-2.83%	-2.43%	-1.23%	-1.28%
26	-14.42%	-13.88%	-1.27%	-0.87%	-2.76%	-2.43%	-1.26%	-1.32%
28	-14.55%	-14.02%	-1.28%	-0.88%	-2.68%	-2.44%	-1.29%	-1.37%
30	-14.69%	-14.16%	-1.29%	-0.89%	-2.61%	-2.46%	-1.33%	-1.42%
32	-14.78%	-14.27%	-1.27%	-0.87%	-2.51%	-2.43%	-1.32%	-1.44%

6.3.5 Encoding and Decoding Times

The main drawback of increasing the number of prediction modes is a higher encoding time. The encoder evaluates $P(\theta)$ = 35 + (33 $\times$ θ) prediction modes for each PU, measuring the RDO to select the proper intra-prediction mode and ω. As θ increases, more curve-based intra prediction modes are available for the encoder, increasing the encoding times.

Figures 6.15 and 6.16 illustrate the encoding times comparing the baseline with the Centerline and Radial models, respectively. The results for the lowest available number of curve displacements θ = 2 reveal the encoding times nearly triplicate for both curve-based models, at 265.73% and 266.78% of the baseline encoding time, for the All-Intra-8 configuration in Centerline and Radial models, respectively. Tables 6.9 and 6.10 show a detailed report of the encoding times for all configurations and values of θ for the Centerline and Radial models, respectively. The encoding times increase linearly for the All-Intra configurations for both curve-based models. The exceptions are the Random-Access and Low-delay configurations, as in these cases, the HEVC encoder uses inter-frame prediction prominently.

The increase in encoding time is compatible with the rise in the number of new prediction modes available to the encoder, since: (i) the baseline contains 35 intra-prediction modes; (ii) θ = 2 corresponds to 101 intra-prediction modes or 2.88 times as many intra-prediction modes as the baseline. This analysis of the encoding times and the number of prediction modes shows that the encoding cost for the curve-based models comes entirely from the increase of the available curve-based intra modes; meanwhile, the curve displace-

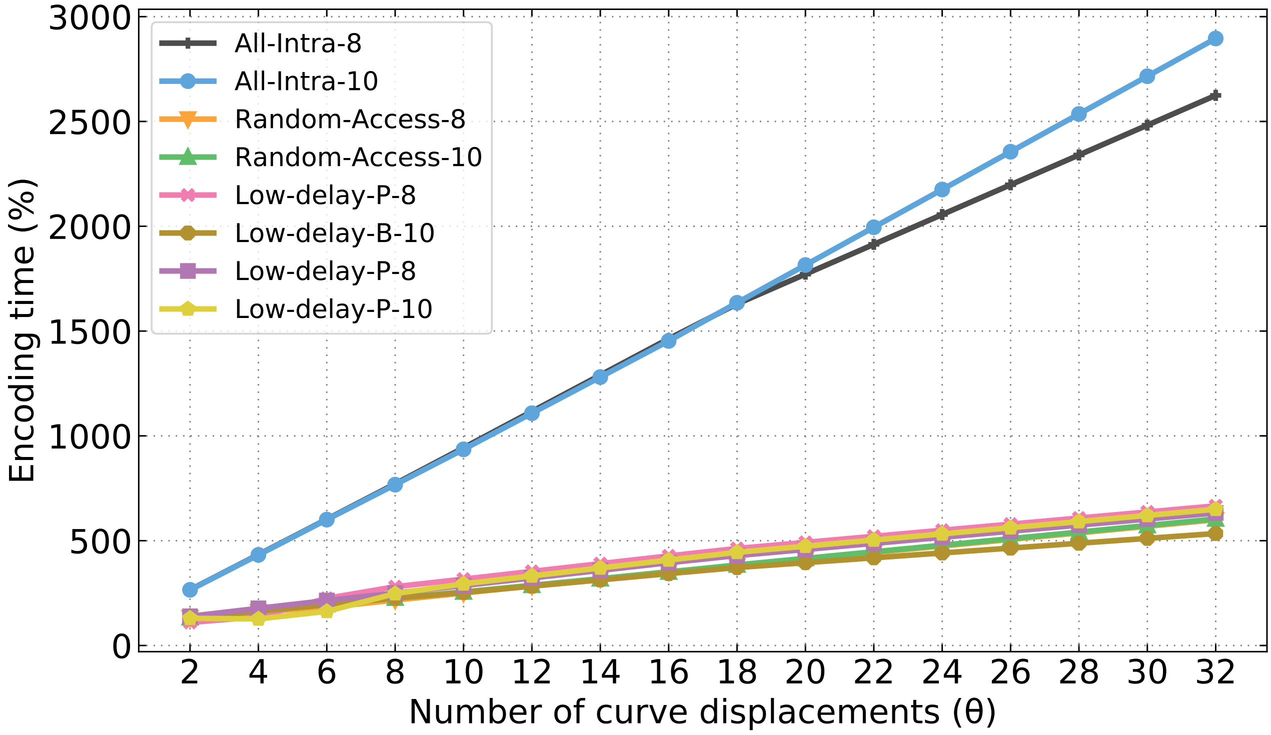


Figure 6.15 – Centerline model encoding time compared to the standard HEVC implementation, for θ ranging from 2 to 32.

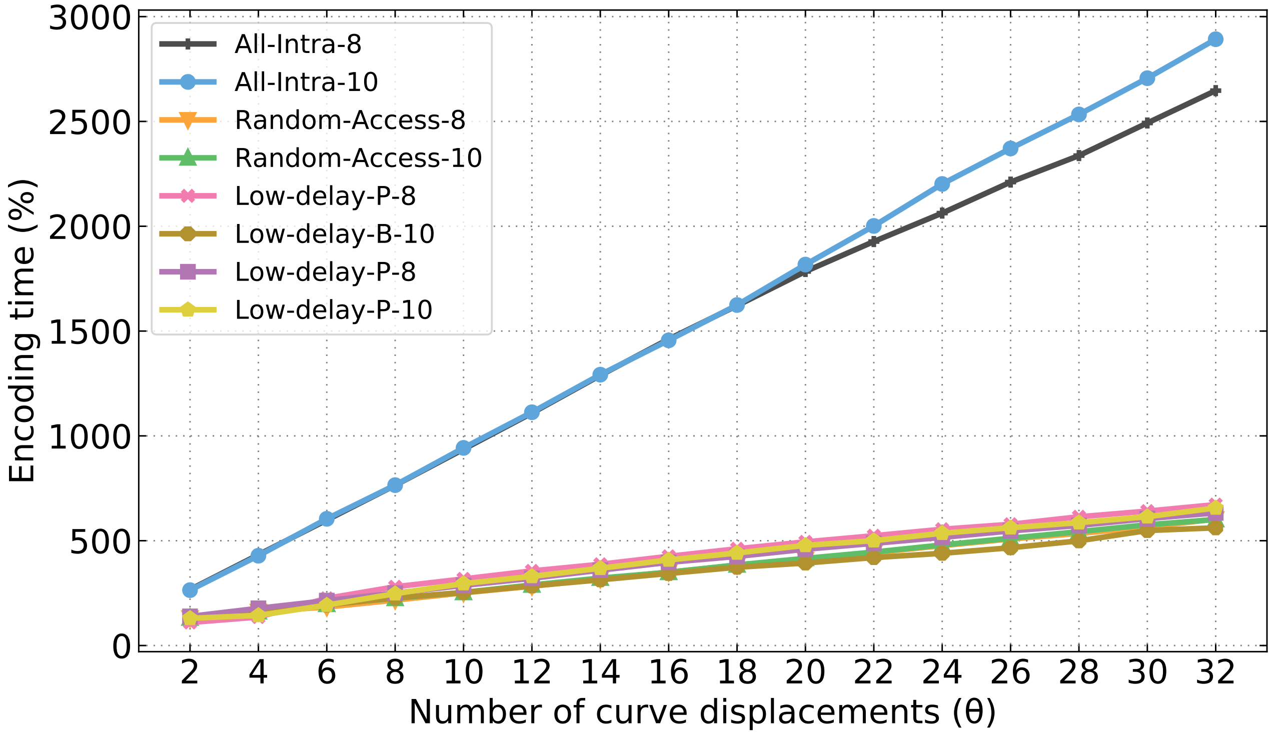


Figure 6.16 – Radial model encoding time compared to the standard HEVC implementation, for θ ranging from 2 to 32.

ment calculations result in a negligible overhead. Likewise, a comparison of both curve-based models shows no differences in computing times between both models, with mostly

Table 6.9 – Centerline model encoding times averaged per configuration for all test sequences and θ values.

θ	All-Intra-8	All-Intra-10	Random-Access-8	Random-Access-10	Low-delay-B-8	Low-delay-B-10	Low-delay-P-8	Low-delay-P-10
2	265.73%	265.60%	132.05%	132.45%	109.93%	132.70%	139.33%	129.67%
4	433.66%	432.25%	161.11%	162.39%	136.61%	163.69%	177.38%	126.84%
6	602.15%	600.50%	181.98%	195.57%	224.89%	194.72%	214.57%	162.81%
8	771.32%	767.30%	216.57%	226.31%	280.52%	225.36%	250.14%	247.38%
10	943.26%	935.78%	250.23%	255.48%	316.68%	252.90%	285.97%	293.95%
12	1116.56%	1108.13%	284.25%	287.13%	353.57%	282.85%	322.19%	331.85%
14	1289.88%	1280.51%	318.29%	318.80%	390.48%	312.83%	358.43%	369.78%
16	1463.22%	1452.93%	352.36%	350.51%	427.42%	342.84%	394.71%	407.74%
18	1629.43%	1635.01%	380.37%	384.32%	462.85%	371.98%	428.95%	444.49%
20	1771.55%	1815.11%	411.70%	415.64%	491.87%	395.12%	457.97%	473.80%
22	1913.68%	1995.25%	443.04%	446.99%	520.92%	418.29%	487.00%	503.15%
24	2055.82%	2175.37%	474.39%	478.34%	549.96%	441.47%	516.03%	532.48%
26	2197.96%	2355.51%	505.74%	509.68%	579.03%	464.63%	545.08%	561.83%
28	2340.09%	2535.65%	537.09%	541.04%	608.07%	487.82%	574.11%	591.16%
30	2482.24%	2715.78%	568.44%	572.38%	637.13%	510.98%	603.16%	620.51%
32	2624.41%	2895.94%	599.83%	603.77%	666.20%	534.18%	632.23%	649.88%

Table 6.10 – Radial model encoding times averaged per configuration for all test sequences and θ values.

θ	All-Intra-8	All-Intra-10	Random-Access-8	Random-Access-10	Low-delay-B-8	Low-delay-B-10	Low-delay-P-8	Low-delay-P-10
2	266.78%	264.18%	130.89%	131.42%	109.88%	133.65%	138.98%	129.71%
4	433.09%	428.36%	161.80%	161.26%	135.83%	163.87%	177.58%	143.78%
6	598.56%	604.44%	183.51%	196.66%	225.18%	193.48%	214.16%	192.71%
8	764.39%	765.05%	216.51%	225.15%	280.27%	227.57%	249.86%	247.45%
10	936.50%	942.68%	251.21%	253.29%	317.73%	252.73%	286.80%	295.35%
12	1108.77%	1112.72%	282.90%	288.95%	356.03%	284.25%	320.79%	329.74%
14	1288.70%	1292.02%	317.88%	321.80%	387.60%	313.43%	359.48%	368.47%
16	1461.60%	1455.72%	350.40%	349.20%	425.48%	343.78%	396.74%	408.27%
18	1621.37%	1624.27%	383.18%	384.59%	461.50%	373.44%	425.09%	441.39%
20	1784.25%	1817.13%	409.83%	415.76%	494.50%	393.64%	459.22%	477.78%
22	1927.02%	2001.50%	442.37%	445.77%	523.81%	419.60%	487.44%	500.15%
24	2062.68%	2201.65%	476.64%	479.85%	554.69%	439.91%	515.77%	535.76%
26	2210.61%	2370.97%	508.24%	511.79%	578.39%	466.36%	546.68%	561.77%
28	2337.02%	2533.35%	534.76%	543.18%	614.15%	499.44%	572.78%	585.96%
30	2492.52%	2706.17%	565.28%	575.92%	641.18%	548.55%	605.42%	614.56%
32	2646.94%	2892.14%	603.16%	601.14%	672.19%	561.43%	632.68%	656.11%

identical encoding times and with differences often attributed to external influences in the computational environment.

The increased computational cost of the curves model, associated with the diminishing return of BD-Rate gains for large θ values, makes the use of many curve displacements less attractive; nevertheless, if the goal is purely encoding efficiency, the additional encoding times might be a worthwhile trade-off. Regarding the decoding times, the curve-based models do not need more computational effort than in standard HEVC. After obtaining the ω value from the encoded video bitstream, the intra-frame predicted block reconstruction works similar to the original HEVC implementation, using Equations 5.7 and 5.8 for the Centerline model and Equations 5.11 and 5.12 for the Radial model.

6.3.6 Mixed Centerline and Radial Models

While the Radial model achieved better BD-Rate results in the All-Intra configurations, the Centerline model performed better with the Random-Access and Low-delay configurations. Thus, the combination of both models can result in a set of new curve-based intra modes for the HEVC encoder with the capacity of improving BD-Rate results.

Considering the previous results showing that $\theta = 8$ offers a good compromise of coding efficiency and encoding times, the tests for the mixed Centerline and Radial models use the curve-based intra modes from both curve-based models with $\theta = 8$, i.e., the encoder has $P(8) = 35 + (2 \times 33 \times 8) = 563$ intra-prediction modes available.

Table 6.11 shows the results of BD-Rate, encoding times, and residual energy for the mixed curve-based models with all encoder configurations. Table 6.11 also compares the Centerline and Radial models results with $\theta = 8$ and $\theta = 32$, the latter corresponding to the most significant BD-Rate reductions in both curve-based models.

The All-Intra configurations achieved even better average BD-Rate reductions with the mixed curve-based models at ≈ 4% and ≈ 3.5%, respectively. These results surpass the Radial model with $\theta = 32$, which achieved the most significant BD-Rate results in the earlier experiments; however, with the benefit of much lower encoding time overhead, since the encoder has fewer intra-prediction modes to evaluate. Like in the previous sections, the residual energy results for the All-Intra configurations follow the same BD-Rate reductions, justifying the coding efficiency gains. In contrast, the Random-Access configurations demonstrated worse overall BD-Rate improvements with the mixed curve-based models, despite the greater residual energy reduction. While video classes C, D, and F showed BD-Rate reductions with the mixed curve-based models similar to the Centerline and Radial models, the encoding efficiency got worse for classes A1, A2, and B, due to the overhead associated with encoding the curve displacement values in these cases, despite the improved prediction indicated by the lowered residual energy. Regarding the Low-delay configurations, except for the Low-delay-B-10 configuration, the results also demonstrate improved BD-Rate accompanied by lowered residual energy, while the encoding times fall between the $\theta = 8$ and $\theta = 32$ evaluations.

Based on these results, another possibility for improving coding efficiency is to restrict the curve-based intra modes to I-Frames. This approach benefits from achieving a higher compression of I-Frames, while the P- and B-Frames rely solely on the default HEVC intra- and inter-frame predictors.

Table 6.11 – Mixed curve-based models results. The Centerline and Radial models use θ = 8 for the displacement values. The Cθ and Rθ configurations correspond to the Centerline and Radial models, respectively.

Configuration		BD-Rate								Encoding time	Residual energy
		A1	A2	B	C	D	E	F	Overall		
All-Intra-8	Cθ = 8	-2.32%	-2.47%	-1.77%	-2.50%	-2.55%	-1.69%	-1.99%	-2.17%	771%	-11.02%
	Rθ = 8	-2.63%	-2.59%	-1.73%	-3.57%	-3.66%	-1.89%	-2.77%	-2.69%	767%	-12.72%
	Cθ = 32	-2.65%	-2.17%	-1.49%	-2.62%	-2.56%	-0.74%	-2.02%	-2.04%	2624%	-12.57%
	Rθ = 32	-3.36%	-2.63%	-1.77%	-4.27%	-4.17%	-1.43%	-3.33%	-3.01%	2623%	-14.78%
	Mixed	-3.83%	-4.02%	-3.22%	-4.84%	-4.60%	-2.99%	-3.86%	-3.92%	1424%	-17.51%
All-Intra-10	Cθ = 8	-1.83%	-2.10%	-1.51%	-2.33%	-2.41%	-1.48%	-1.82%	-1.92%	767%	-10.48%
	Rθ = 8	-2.17%	-2.28%	-1.53%	-3.45%	-3.58%	-1.68%	-2.67%	-2.49%	768%	-12.17%
	Cθ = 32	-2.20%	-1.87%	-1.35%	-2.61%	-2.57%	-0.62%	-1.97%	-1.90%	2896%	-11.99%
	Rθ = 32	-2.81%	-2.26%	-1.52%	-4.13%	-4.10%	-1.11%	-3.14%	-2.76%	2923%	-14.27%
	Mixed	-3.00%	-3.48%	-2.76%	-4.54%	-4.33%	-2.56%	-3.62%	-3.49%	1446%	-16.78%
Random-Access-8	Cθ = 8	-0.54%	-0.57%	-0.83%	-1.50%	-1.49%	—	-1.82%	-1.16%	217%	-1.52%
	Rθ = 8	-0.01%	0.33%	-0.34%	-1.49%	-1.54%	—	-1.81%	-0.88%	216%	-0.79%
	Cθ = 32	-0.38%	-0.13%	-0.91%	-1.85%	-1.76%	—	-2.38%	-1.31%	600%	-1.44%
	Rθ = 32	0.45%	1.16%	-0.05%	-1.66%	-1.70%	—	-2.21%	-0.77%	599%	-1.27%
	Mixed	0.78%	0.95%	0.09%	-1.62%	-1.27%	—	-1.84%	-0.58%	390%	-5.26%
Random-Access-10	Cθ = 8	-0.36%	-0.56%	-0.80%	-1.44%	-1.38%	—	-1.68%	-1.08%	226%	-1.50%
	Rθ = 8	0.16%	0.26%	-0.32%	-1.49%	-1.55%	—	-1.82%	-0.86%	225%	-0.77%
	Cθ = 32	-0.18%	-0.19%	-0.82%	-1.77%	-1.75%	—	-2.31%	-1.24%	604%	-1.75%
	Rθ = 32	0.65%	1.13%	-0.03%	-1.61%	-1.64%	—	-2.13%	-0.71%	605%	-0.87%
	Mixed	0.64%	0.55%	0.05%	-1.49%	-1.16%	—	-1.79%	-0.61%	389%	-5.65%
Low-delay-B-8	Cθ = 8	—	—	-0.69%	-0.37%	-0.19%	-0.69%	-0.74%	-0.54%	281%	-0.91%
	Rθ = 8	—	—	0.41%	0.33%	0.49%	0.24%	-0.04%	0.30%	281%	-2.58%
	Cθ = 32	—	—	-0.23%	-0.44%	-0.27%	-0.82%	-1.01%	-0.52%	666%	-0.79%
	Rθ = 32	—	—	0.81%	0.88%	1.13%	0.82%	0.35%	0.80%	662%	-2.51%
	Mixed	—	—	-0.75%	-1.09%	-0.57%	-1.24%	-1.57%	-1.02%	390%	-8.03%
Low-delay-B-10	Cθ = 8	—	—	-0.09%	-0.40%	-0.32%	-0.68%	-0.82%	-0.43%	225%	-1.71%
	Rθ = 8	—	—	0.40%	-0.07%	0.09%	-0.08%	-0.38%	0.02%	225%	-2.69%
	Cθ = 32	—	—	-0.82%	-0.51%	-0.39%	-0.59%	-1.10%	-0.69%	534%	-1.08%
	Rθ = 32	—	—	0.28%	0.33%	0.46%	0.37%	-0.33%	0.21%	561%	-2.43%
	Mixed	—	—	1.03%	-1.03%	-0.69%	-1.19%	-1.00%	-0.47%	392%	-7.87%
Low-delay-P-8	Cθ = 8	—	—	-0.05%	-0.22%	-0.07%	-0.62%	-0.61%	-0.29%	250%	-1.65%
	Rθ = 8	—	—	0.10%	-0.23%	-0.10%	-0.30%	-0.55%	-0.19%	252%	-1.06%
	Cθ = 32	—	—	-0.31%	-0.55%	-0.29%	-0.85%	-1.11%	-0.59%	632%	-1.52%
	Rθ = 32	—	—	1.01%	0.09%	-0.32%	0.29%	0.27%	0.31%	630%	-1.32%
	Mixed	—	—	0.09%	-1.17%	-0.76%	-1.19%	-1.58%	-0.86%	453%	-6.45%
Low-delay-P-10	Cθ = 8	—	—	-0.62%	-0.53%	-0.38%	-0.88%	-0.89%	-0.65%	247%	-1.56%
	Rθ = 8	—	—	-0.31%	-0.29%	-0.17%	-0.19%	-0.71%	-0.34%	248%	-1.18%
	Cθ = 32	—	—	-0.67%	-0.55%	-0.40%	-0.81%	-1.06%	-0.69%	650%	-1.64%
	Rθ = 32	—	—	-0.75%	-0.37%	-0.12%	-0.26%	-0.96%	-0.52%	645%	-1.44%
	Mixed	—	—	-0.62%	-1.35%	-0.69%	-1.36%	-1.41%	-1.05%	402%	-6.84%

6.3.7 Comparison with State-of-the-art

Considering that the experimental results, shown in the previous Sections, indicate $\omega \leq 8$ as an appropriate trade-off point of coding efficiency and encoding times, this section compares the curve-based models with state-of-the-art. The comparison uses MRL and its Fast variant (MRL Fast) [33], Chen's improved intra-prediction encoding [9], along with PDPC [49] comparing BD-Rate and encoding times.

Although it would be preferable to perform an in-depth comparison of the state-of-the-art works with the curve-based models, not all works use an experimental setup that enables a direct comparison of encoder implementations. SLSP [36] uses only videos from classes B, C, and D, with an average of PSNR and bitrate values for each tested video sequence. There is no indication of the QP parameters used for each test, making it difficult to compare their results with state-of-the-art. The works [33] and [9] also only supply results for the All-Intra-8 configuration. Meanwhile, [49] omits the experimental results for class F videos and the encoding times.

Table 6.12 – Results compared with state-of-the-art regarding BD-Rate reduction and encoding times for the All-Intra-8 configuration.

Work		Video class								Encoding time
		A1	A2	B	C	D	E	F	Overall	
MRL [33]		-2.23%		-2.30%	-2.48%	-1.98%	-2.17%	-2.25%	-2.23%	463%
MRL (Fast) [33]		-1.85%		-1.92%	-2.05%	-1.68%	-1.90%	-1.83%	-1.87%	212%
Chen 2017 [9]		-1.84%		-1.85%	-1.94%	-1.70%	-2.17%	-1.94%	-1.89%	230%
PDPC [49]		-2.24%		-2.26%	-1.59%	-1.54%	-2.48%	—	-2.01%	—
Centerline	θ = 2	-0.93%	-1.46%	-0.97%	-1.38%	-1.41%	-1.54%	-1.00%	-1.22%	265%
	θ = 4	-1.83%	-2.24%	-1.53%	-2.25%	-2.31%	-1.86%	-1.80%	-1.96%	433%
	θ = 6	-2.11%	-2.38%	-1.67%	-2.29%	-2.31%	-1.90%	-1.80%	-2.04%	602%
	θ = 8	-2.32%	-2.47%	-1.77%	-2.50%	-2.55%	-1.69%	-1.99%	-2.17%	771%
Radial	θ = 2	-1.36%	-1.50%	-0.93%	-1.74%	-1.88%	-1.21%	-1.25%	-1.40%	266%
	θ = 4	-2.05%	-2.20%	-1.42%	-2.93%	-3.00%	-1.72%	-2.31%	-2.23%	433%
	θ = 6	-2.45%	-2.54%	-1.69%	-3.44%	-3.45%	-1.97%	-2.75%	-2.61%	598%
	θ = 8	-2.63%	-2.59%	-1.73%	-3.57%	-3.66%	-1.89%	-2.77%	-2.69%	764%
Mixed models		-3.83%	-4.02%	-3.22%	-4.84%	-4.60%	-2.99%	-3.86%	-3.92%	1424%

Like the results shown in this Thesis, the related work considers an HM HEVC encoder implementation as the baseline. The BD-Rate results compare the respective works with the baseline rate-distortion metrics obtained from the HM, and the encoding times are relative to the HM baseline encoding times. Table 6.12 provides a full comparison of related work with the Centerline and Radial models, up to θ = 8, and the mixed curve-based models, using the All-Intra-8 configuration. Compared with MRL, the Radial model with θ = 4 offers similar overall BD-Rate improvements with similar encoding times. The optimizations of MRL Fast significantly reduce the encoding time overhead of MRL, however, reducing the BD-

Rate gains seen in the non-optimized variant. The encoding times are comparable to the curve-based models using $\theta = 2$, albeit with improved BD-Rate results for MRL Fast. Chen's improved intra-prediction offers a middle ground of BD-Rate results compared to the Radial model using $\theta = 2$ and $\theta = 4$; however, at much lower encoding times.

Analyzing the related work, the curve-based models present results comparable to the state-of-the-art on BD-Rate and encoding times. While the curve-based models result in higher computational costs, they can also offer more significant BD-Rate improvements for higher values of θ, as shown in Section 6.3.2. Meanwhile, the mixed Centerline and Radial models offer a good trade-off of encoding times and coding efficiency. Unfortunately, the results for any of the Random-Access or Low-delay configurations are absent for the evaluated works. Considering that only All-Intra-8 results are available, the Radial model is the best proposal due to more significant BD-Rate improvements than the Centerline model. However, the Centerline model offered better results for the Random-Access and Low-delay configurations, which would be interesting to compare with the state-of-the-art.

7. CONCLUSIONS, CONTRIBUTIONS AND FUTURE WORK

There is ample space for improvements in video coding standards, with many works in state-of-the-art addressing this research topic. For the past three decades, progress in video coding standards has shown a constant improvement in coding efficiency, with the most recent releases of AV1 in mid-2018 and VVC in mid-2020 achieving further improvements. Usually, a new video coding standard is the culmination of gradual improvements in several aspects of previous standards, such as the ideas discussed in this Thesis.

The LSAS and curve-based models showed qualities for improving current and future video coding standards. While LSAS in its current state is by no means a feasible approach for improving coding efficiency, improvements to its hefty bitrate requirements could turn LSAS into a practical approach. Additionally, LSAS showed its best results for 16×16 block sizes, indicating its possible applicability as a special prediction mode for this PB size, perhaps even compensating the high bitrate requirements with its improved intra-prediction capabilities.

The most relevant improvements occurred with the curve-based models. The Centerline and Radial models achieved improved BD-Rate results in all tested configurations, even with only a few curve displacement values. However, ideally, new methods should not change the bitstream of coding standards. A desirable improvement of the curve-based models would be to infer the curve displacement at decoding time so that the transmission of more information becomes unnecessary; consequently, further improving coding efficiency since there would be no bitstream overhead associated with the new prediction modes.

The curve-based models showed promising results, with an average BD-Rate reduction in the All-Intra configurations of $\approx$ 2% and $\approx$ 3% for the Centerline and Radial models, respectively. The experiments unveiled that a low number of curve displacement values offers a good trade-off in BD-Rate gains versus the encoding times. The $\theta \leq 8$ values provide the best trade-off in BD-Rate gains versus encoding time. Mixing both curve-based models resulted in further encoding efficiency gains, with $\approx$ 4% BD-Rate reductions compared with the baseline, while reducing the encoding times for the maximum tested value of $\theta = 32$ for the Radial model. This set of results satisfies the original hypothesis of this Thesis that it is possible to propose intra-frame prediction techniques capable of reducing the bitstream of encoded videos while keeping the same objective video quality.

This Thesis addressed the problem of the intra-frame prediction, targeting HEVC as the baseline for experiments, answering the hypothesis: “it is possible to propose new intra-frame prediction models that reduce the bitstream of the encoded videos while keeping the same objective video quality.” The results for the curve-based models demonstrate higher coding efficiency, achieved by reducing the residual energy of the encoded test sequences through better intra-frame prediction. A reduction in residual energy also translates into lower bitstream requirements for storing the quantized residuals of PBs. The video coding flow transforms and quantizes the prediction residuals, incurring in information loss and quality degradation upon decoding; thus, by reducing the residual energy, it is possible to keep the same objective quality of encoded videos with a lower bitstream.

The HEVC intra-predictor is very similar to the intra-predictors of other video coding standards, and the idea of curve-based models target intra-prediction in a broad sense. Intra-predictors of other video coding standards still use angular prediction as one of the most suitable methods due to its capability for modeling textures in video frames. Adapting this work to other video coding standards must consider only standard-specific characteristics like the prediction block size and partitioning schemes.

7.1 Contributions of this Work

The Author of this Thesis has a direct collaboration with the Grupo de Sistemas Embarcados (GSE) from the Pontifical Catholic University of Rio Grande do Sul (PUCRS). The works developed with GSE encompassed improvements to video coding technologies, primarily using HEVC as the baseline. Besides LSAS and curve-based models presented in this Thesis, the Author and the research group also explored improvements targeting the intra-frame prediction of 3D-HEVC [51–54]. These research projects contributed to the state-of-the-art in video coding technologies, with a full list of publications during this Ph.D. shown in Appendix A, and the major contributions of this Thesis being:

(i) **The creation of two novel intra-frame prediction models for video coding**—LSAS paved the way for creating the curve-based models. Early research for improving intra-frame prediction looked back at early state-of-the-art attempts at image coding technologies. The use of polynomial surfaces for encoding image blocks, inspired by Kunt et al.'s work in 1985 [28], unveiled the characteristic of naturally occurring curves in image blocks. This observation led to the extension of traditional angular intra-frame prediction tools to encompass a curve displacement parameter that uses samples from the reference arrays in a novel manner. The implementation is simple and computationally efficient since the required calculations only change the frame-indexing step of the intra-predictor. The encoding time overhead noticed in the experiments originated from the necessity of the encoder to exhaustively evaluate each new intra-prediction mode for rate-distortion estimation; this is not an exclusive drawback of the work shown in this Thesis, with other works, like MRL, showing similar encoding time penalties for the same reasons; however, optimizations like the combined Centerline and Radial models can achieve a compromise in coding efficiency and encoding times;

(ii) **Improvement of coding efficiency for an existing video coding standard**—The improvements achieved with HEVC demonstrate the efficiency of the curve-based models. Experimental results also illustrate that the curve-based intra modes concentrated around naturally occurring textures in block contents, like human hair, tree branches, water ripples, and text content of screen recordings. The initial assumption of using curve-based prediction was to reduce residuals due to improved intra-frame prediction, enabling the better representation of textures of video block contents. The experiments show this assumption as the reason for the improved BD-Rate improvements, with the coding efficiency results closely matching the residual energy reduction; and

(iii) **The creation of Centerline and Radial models**—The Centerline and Radial models follow a similar simple implementation idea, improving the existing intra-prediction tools. A simple modification of the existing curve-based models, e.g., changing the curve displacement weights shown for the Radial model in Figure 5.9, can yield different results while requiring minimal modifications of the existing curve-based models. Other curve-based intra-prediction models can use the Centerline and Radial models as the starting point for future research into improving intra-frame prediction.

7.1.1 Other Contributions

Although not directly related to video coding technologies, the Author of this Thesis also took part in other academic research projects at GSE. The Author published works related to simulators for memory technologies, communication algorithms, and security for Network-on-Chip systems [6, 7, 17, 55]. These works employ complex computational models for full system simulation for different target platforms and environment constraints.

Of particular note is the work [6] and its later improvement in [7], which uses a full system simulator capable of executing real-world applications in a simulated environment. The proposal of [6] and [7] is a novel hardware/software solution for data synchronization in parallel systems, capable of accelerating parallel applications without modifying the existing source code. The results showed an average speedup of 57% for all tested applications. One of the several experiments consists of a parallel implementation of the X.264 video encoder, used for encoding a limited set of test sequences in a low-powered embedded system. This platform offers an interesting possibility for testing the X.265 encoder, which incorporates multiple parallel techniques for accelerating the HEVC encoder.

7.2 Discussion and Future Work

Considering the work discussed in Chapter 4 and the results of Section 6.3.3, the curve-based models are prone to encoder optimizations, intending to mitigate the computational overhead it introduces. One possibility is identifying the most likely curve displacement value based on characteristics of the predicted block or, ideally, characteristics of neighboring blocks; the latter a preferred approach since it removes the necessity of embedding more information in the HEVC bitstream. The results also showed that larger PBs benefit from large ω values, where finding the ranges of curve displacements for each PB to create constraints on the ω values is another research path. Future work could further improve the results of the curve-based models, achieving more significant BD-Rate improvements while keeping the same, but preferably lower, computational overhead. This Thesis already showed some effort towards optimizing the curve-based prediction model by mixing the Centerline and Radial models, providing more possibilities for the HEVC intra-frame predictor.

Another research venue for the curve-based models is their use for encoding specific image content. As shown in the experimental results, the class F videos consistently benefit from the curve-based models. HEVC SCC [34] offers intra-prediction tools specific for encoding text content in videos. SCC introduces the intra block copy mode, which uses the idea of inter-frame prediction for detecting high-correlated frame regions. Similar blocks, e.g., frame regions with repeating textual patterns, can reference a single previously encoded block as its prediction source, akin to MCP used in inter-frame prediction. SCC makes many other improvements in the HEVC intra-predictor for enhancing coding efficiency for screen content in videos, showing the necessity of content-specific prediction models. Using the curve prediction models as an especial prediction mode for SCC in HEVC or other video coding standards is one possibility since there was a high use of prediction modes, shown in Figure 6.12(d), where curve-based intra modes concentrated around textual screen content.

7.3 Considerations for VVC Integration

The most pressing evaluation needed for the curve-based prediction models is to incorporate it with the new video coding standards. When the research of this Thesis began, neither the AV1 nor the VVC standards were complete. Both new coding standards offer improved intra-prediction tools, among many other improvements, which might benefit from the curve-based models. These new coding standards also feature larger block sizes and non-square block segmentation, the latter a challenge for adapting the curve-based models. Considering that VVC is the direct successor of HEVC, adapting the curve-based models to VVC is the likely future research path.

As Figure 7.1 illustrates, besides the thirty-five intra-frame prediction modes of HEVC, VVC includes thirty-three new intra-prediction modes, for a total of sixty-five. This extension of prediction modes enables a more refined control for modeling block contents, thus improving coding efficiency. Like in HEVC, the VVC intra-predictor performs the two-point interpolation of two adjacent reference samples from one of the two reference arrays, according to the angular prediction direction. To minimize the encoding time penalties of the thirty-three new intra-prediction modes, VVC initially uses the same thirty-five HEVC prediction modes with RMD, generating a list with the MPM. According to the MPM modes chosen by RMD, VVC only evaluates the new angular intra-prediction modes closest to the MPM for rate-distortion.

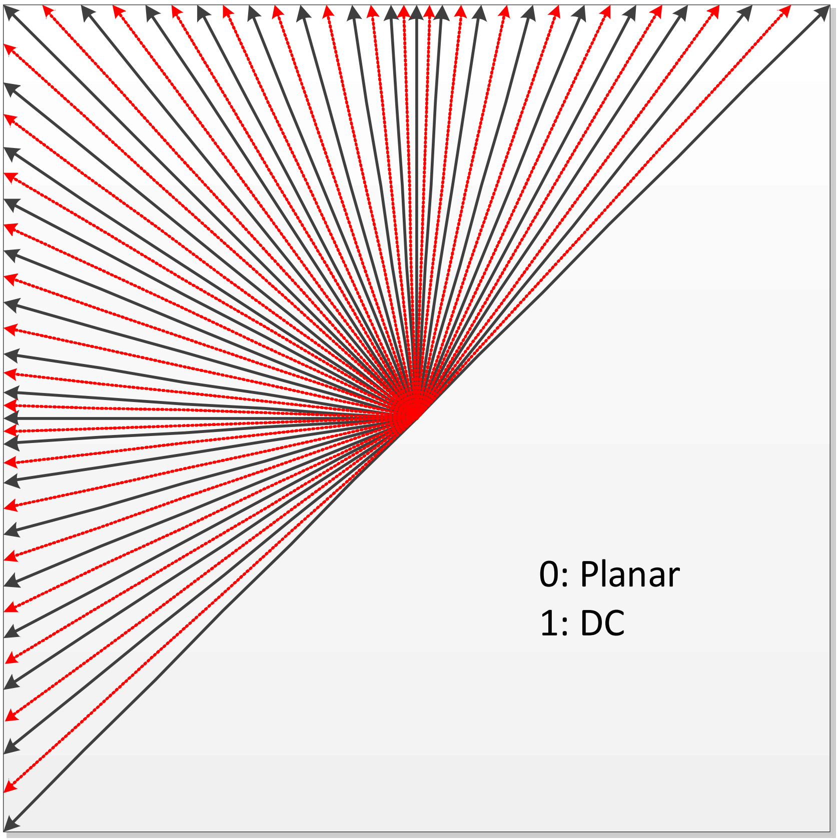
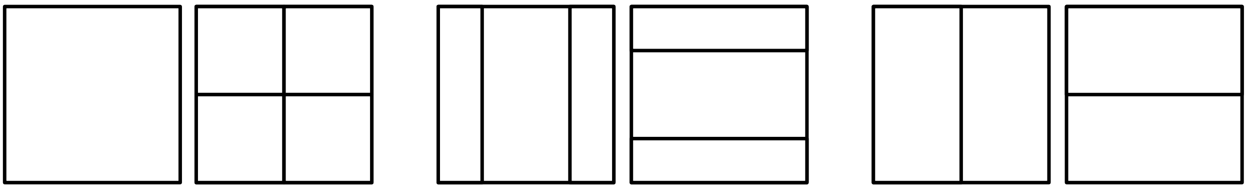


Figure 7.1 – VVC intra-prediction modes. The dotted red lines correspond to the thirty-three new modes, while the black lines correspond to the same modes of HEVC (captured from [76]).

Another significant difference of VVC is the use of a Multi-Type Tree (MTT) instead of a CTU for frame partitioning. MTT allows a quaternary partitioning scheme, like HEVC, and the ternary and binary partitioning schemes, as Figure 7.2 illustrates. MTT adds flexibility to the partitioning scheme, grouping frame regions with higher similarity and improving intra-prediction. In HEVC, the symmetric partitioning of the CTU for intra-prediction enables the encoder to use any of the thirty-three angular modes since the reference arrays are symmetrical around the PB. Since MTT creates rectangular blocks in binary and ternary partitioning, it is necessary to address the asymmetry of the reference arrays for these cases.

VVC addresses angular prediction with block asymmetry using reference arrays of size $2H+1$ and $2W+1$, for the left and above reference arrays, respectively, where H is the block height, and W is the block width.



(a) Quaternary (b) Ternary (c) Binary

Figure 7.2 – Versatile Video Coding MTT partitioning schemes. HEVC only allows quaternary partitioning (a) for I-Frames, only creating square block subdivisions.

As mentioned in this Thesis, MRL is one of the improvements that VVC uses for intra-prediction, enabling more reference lines as candidates for intra-frame prediction. Unlike the work [33] shown earlier in this Thesis, the VVC adopts optimizations for implementing MRL: (i) MRL uses only three candidate reference lines, instead of the four possibilities of the original proposal; (ii) VVC uses MRL only for the MPM modes; (iii) Planar intra-prediction does not use MRL, adopting only the first reference line, like in HEVC. These optimizations reduce the computational overhead of VVC since the encoder does not need to evaluate all intra-prediction modes for each reference line in MRL.

Although there are other intra-prediction improvements in VVC, the ones presented are the most relevant for incorporating the curve-based prediction models. Using the MPM optimizations of VVC is a promising idea for incorporating the curve-based models with the new VVC intra-frame predictor: for each MPM candidate, also test the MPM and the adjacent modes with the curve displacement values. Likewise, the encoder could test the MPM modes with MRL and the curve displacements. Still, the most challenging aspect for implementing the curve-based models in VVC is the asymmetrical partitions of MTT. The Centerline and Radial models assume the PB to be symmetrical, with its center equidistant from all sides of the PB. A simple approach uses curves in symmetrical partitions of the MTT, like how it works currently in HEVC. The more elaborate idea considers the MTT as a single block. Depending on its partitioning layout, each block contains restrictions to use only specific curve displacement values according to the coordinates of each sub partition; thus, generating curve patterns spanning the entire MTT.

REFERENCES

[1] Akyazi, P.; Ebrahimi, T. “Comparison of Compression Efficiency between HEVC/H.265, VP9 and AV1 Based on Subjective Quality Assessments”. In: International Conference on Quality of Multimedia Experience (QoMEX), 2018, pp. 6.

[2] Bankoski, J.; Wilkins, P.; Xu, Y. “Technical Overview of VP8, an Open Source Video Codec for the Web”. In: IEEE International Conference on Multimedia and Expo (ICME), 2011, pp. 6.

[3] Bjøntegarrd, G. “VCEG-AI11, Improvements of the BD-PSNR Model”, Technical Report, Video Coding Experts Group (VCEG), 2008, 16p.

[4] Bossen, F. “JCTVC-L1100, Common Test Conditions and Software Reference Configurations for HM”, Technical Report, Joint Collaborative Team on Video Coding (JCT-VC), 2013, 4p.

[5] Bross, B.; Chen, J.; Liu, S. “Working Draft 5 of Versatile Video Coding”. Source: https://mpeg.chiariglione.org/standards/mpeg-i/versatile-video-coding/working-draft-5-versatile-video-coding, Nov 2020.

[6] Cataldo, R.; Fernandes, R.; Martin, K.; Sepúlveda, J.; Susin, A.; Marcon, C. “Subutai: Distributed Synchronization Primitives in NoC Interfaces for Legacy Parallel-Applications”. In: Design Automation Conference (DAC), 2018, pp. 6.

[7] Cataldo, R.; Fernandes, R.; Sanchez, G.; Silveira, J.; Martin, K.; Diguet, J.-P.; Marcon, C.; Sepúlveda, J. “Subutai: Speeding up Legacy Parallel Applications through Data Synchronization”, *IEEE Transactions of Parallel and Distributed Systems*, vol. 1–1, Nov 2020, pp. 14.

[8] Chang, Y.-J.; Jhu, H.-J.; Jian, H.-Y.; Zhao, L.; Zhao, X.; Li, X.; Liu, S.; Bross, B.; Keydel, P.; Schwarz, H.; Marpe, D.; Wiegand, T. “Intra Prediction Using Multiple Reference Lines for the Versatile Video Coding Standard”. In: Applications of Digital Image Processing (ADIP), 2019, pp. 6.

[9] Chen, C.; Zhu, S.; Zeng, B.; Gabbouj, M. “A New Block-Based Method for HEVC Intra Coding”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27–8, Aug 2017, pp. 1727–1736.

[10] Chen, J.; Alshina, E.; Sullivan, G.; others. “N17055, Algorithm Description of Joint Exploration Test Model 7 (JEM7)”, Technical Report, Joint Video Exploration Team (JVET), 2017, 48p.

[11] Cisco Systems, Inc. “Cisco Visual Networking Index: Forecast and Methodology, 2016-2021”. Source: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html#_Toc484813989, Aug 2020.

[12] Cuff, P. “Frequency Domain and Fourier Transforms”. Source: https://www.princeton.edu/~cuff/ele201/kulkarni.html, Oct 2018.

[13] da Silva, T. L.; Agostini, L. V.; da Silva Cruz, L. A. “Fast HEVC Intra Prediction Mode Decision Based on EDGE Direction Information”. In: European Signal Processing Conference (EUSIPCO), 2012, pp. 1214–1218.

[14] Daede, T.; Norkin, A.; Brailovskiy, I. “Video Codec Testing and Quality Measurement”. Source: https://tools.ietf.org/id/draft-ietf-netvc-testing-06.html, Jul 2020.

[15] Feng, L.; Dai, M.; Zhao, C.-l.; Xiong, J.-y. “Fast Prediction Unit Selection Method for HEVC Intra Prediction Based on Salient Regions”, *Optoelectronics Letters*, vol. 12–4, Jul 2016, pp. 316–320.

[16] Fernandes, P.; Bernardo, M. V.; G. Pinheiro, A. M.; Fiadeiro, P. T.; Pereira, M. “Quality Comparison of the HEVC and VP9 Encoders Performance”, *Multimedia Tools and Applications*, vol. 76–11, Jun 2017, pp. 13633–13649.

[17] Fernandes, R.; Marcon, C.; Cataldo, R.; Sepúlveda, J. “Using Smart Routing for Secure and Dependable NoC-based MPSoCs”, *IEEE Transactions on Network*, vol. 28–3, Jun 2020, pp. 1158 – 1171.

[18] Fernandes, R.; Sanchez, G.; Cataldo, R.; Agostini, L.; Marcon, C. “Least-Squares Approximation Surfaces for High Quality Intra-Frame Prediction in Future Video Standards”. In: International Conference on Electronics, Circuits and Systems (ICECS), 2018, pp. 205–208.

[19] Fernandes, R.; Sanchez, G.; Cataldo, R.; Webber, T.; Marcon, C. “Efficient HEVC Intra-Frame Prediction Using Curved Angular Modes”, *Electronics Letters*, vol. 54–21, Sep 2018, pp. 1214–1216.

[20] Garcia, D. C.; de Queiroz, R. L. “Least-Squares Directional Intra Prediction in H.264/AVC”, *IEEE Signal Processing Letters*, vol. 17–10, Oct 2010, pp. 831–834.

[21] Grange, A.; Alvestrand, H. “A VP9 Bitstream Overview”. Source: https://tools.ietf.org/id/draft-grange-vp9-bitstream-00.html, Sep 2018.

[22] Grois, D.; Marpe, D.; Mulayoff, A.; Itzhaky, B.; Hadar, O. “Performance Comparison of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC Encoders”. In: Picture Coding Symposium (PCS), 2013, pp. 394–397.

[23] Grois, D.; Nguyen, T.; Marpe, D. “Coding Efficiency Comparison of AV1/VP9, H.265/MPEG-HEVC, and H.264/MPEG-AVC Encoders”. In: Picture Coding Symposium (PCS), 2016, pp. 1–5.

[24] Hu, N.; Yang, E.-H. “Fast Mode Selection for HEVC Intra-Frame Coding With Entropy Coding Refinement Based on a Transparent Composite Model”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25–9, Sep 2015, pp. 1521–1532.

[25] Jacobs, M.; Probell, J. “A Brief History of Video Coding”, *ARC International*, vol. 1–1, Jan 2007, pp. 6.

[26] Karczewicz, M. “VCEG-AF15, Improved Intra Coding”, Technical Report, Video Coding Experts Group (VCEG), 2007, 9p.

[27] Kim, I.-K.; Min, J.; Lee, T.; Han, W.-J.; Park, J. “Block Partitioning Structure in the HEVC Standard”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22–12, Dec 2012, pp. 1697–1706.

[28] Kunt, M.; Ikonomopoulos, A.; Kocher, M. “Second-Generation Image-Coding Techniques”, *Proceedings of the IEEE*, vol. 73–4, Apr 1985, pp. 549–574.

[29] Lainema, J.; Bossen, F.; Han, W.-J.; Min, J.; Ugur, K. “Intra Coding of the HEVC Standard”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22–12, Dec 2012, pp. 1792–1801.

[30] Lainema, J.; Ugur, K. “Angular Intra Prediction in High Efficiency Video Coding (HEVC)”. In: IEEE International Workshop on Multimedia Signal Processing (MMSP), 2011, pp. 1–5.

[31] Lawson, C.; Hanson, R. “Solving Least Squares Problems”. New Jersey: Society for Industrial Mathematics, 1987, 352p.

[32] Layek, M. A.; Thai, N. Q.; Hossain, M. A.; Thu, N. T.; Tuyen, L. P.; Talukder, A.; Chung, T.; Huh, E.-N. “Performance Analysis of H.264, H.265, VP9 and AV1 Video Encoders”. In: Asia-Pacific Network Operations and Management Symposium (APNOMS), 2017, pp. 322–325.

[33] Li, J.; Li, B.; Xu, J.; Xiong, R. “Efficient Multiple-Line-Based Intra Prediction for HEVC”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28–4, Apr 2018, pp. 947–957.

[34] Liu, S.; Xu, X.; Lei, S.; Jou, K. “Overview of HEVC Extensions on Screen Content Coding”, *APSIPA Transactions on Signal and Information Processing*, vol. 26–1, Jan 2015, pp. 50 – 62.

[35] Liu, X.; Liu, Y.; Wang, P.; Lai, C.-F.; Chao, H.-C. “An Adaptive Mode Decision Algorithm Based on Video Texture Characteristics for HEVC Intra Prediction”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27–8, Aug 2017, pp. 1737–1748.

[36] Lucas, L. F. R.; Rodrigues, N. M. M.; Pagliari, C. L.; da Silva, E. A. B.; de Faria, S. M. M. “Sparse Least-Squares Prediction for Intra Image Coding”. In: IEEE International Conference on Image Processing (ICIP), 2015, pp. 1115–1119.

[37] Marpe, D.; Schwarz, H.; Wiegand, T. “Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13–7, Jul 2003, pp. 620–636.

[38] Matsuo, S.; Takamura, S.; Shimizu, A. “Modification of Intra Angular Prediction in HEVC”. In: Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012, pp. 1–4.

[39] Microsoft Coorporation. “About YUV Video”. Source: https://docs.microsoft.com/en-us/windows/desktop/medfound/about-yuv-video, Sep 2018.

[40] Microsoft Coorporation. “Recommended 8-Bit YUV Formats for Video Rendering”. Source: https://docs.microsoft.com/en-us/windows/desktop/medfound/recommended-8-bit-yuv-formats-for-video-rendering, Sep 2018.

[41] Mukherjee, D.; Bankoski, J.; Grange, A.; Han, J.; Koleszar, J.; Wilkins, P.; Xu, Y.; Bultje, R. “The Latest Open-Source Video Codec VP9 - An Overview and Preliminary Results”. In: Picture Coding Symposium (PCS), 2013, pp. 390–393.

[42] Radiocommunication Sector of ITU. “Recommendation BT.709, Parameter Values for the HDTV Standards for Production and International Programme Exchange”, Technical Report, International Communication Union (ITU-T), 2015, 19p.

[43] Rao, K.; Yip, P. “Discrete Cosine Transform: Algorithms, Advantages, Applications”. San Diego: Academic Press, 1990, 512p.

[44] Reitmeier, G.; Sullivan, G. J. “Video Compression and Its Role in the History of Television”, *SMPTE Motion Imaging Journal*, vol. 125–6, Aug 2016, pp. 60–74.

[45] Rhee, C. E.; Lee, K.; Kim, T. S.; Lee, H.-J. “A Survey of Fast Mode Decision Algorithms for Inter-Prediction and Their Applications to High Efficiency Video Coding”, *IEEE Transactions on Consumer Electronics*, vol. 58–4, Nov 2012, pp. 1375–1383.

[46] Richardson, I. “Video Codec Design: Developing Image and Video Compression Systems”. Chichester: John Wiley and Sons, 2002, 304p.

[47] Rivaz, P.; Haughton, J. “AV1 Bitstream & Decoding Process Specification”, Technical Report, Argon Design LTD, 2018, 678p.

[48] Rosewarne, C.; Bross, B.; Naccari, M.; et al. “High Efficiency Video Coding (HEVC) Test Model 16 (HM 16) Improved Encoder Description Update 9”, Technical Report, The Joint Collaborative Team on Video Coding (JCT-VC), 2017, 62p.

[49] Said, A.; Zhao, X.; Karczewicz, M.; Chen, J.; Zou, F. “Position Dependent Prediction Combination for Intra-Frame Video Coding”. In: IEEE International Conference on Image Processing (ICIP), 2016, pp. 534–538.

[50] Samet, H. “The Quadtree and Related Hierarchical Data Structures”, *ACM Computing Surveys*, vol. 16–2, Jun 1984, pp. 187–260.

[51] Sanchez, G.; Fernandes, R.; Agostini, L.; Marcon, C. “DCDM-Intra: Dynamically Configurable 3D-HEVC Depth Maps Intra-Frame Prediction Algorithm”. In: IEEE International Conference on Image Processing (ICIP), 2018, pp. 1782–1786.

[52] Sanchez, G.; Fernandes, R.; Cataldo, R.; Agostini, L.; Marcon, C. “Low Area Reconfigurable Architecture for 3D-HEVC DMMs Decoder Targeting 1080p Videos”. In: IEEE International Conference on Electronics, Circuits and Systems (ICECS), 2018, pp. 201–204.

[53] Sanchez, G.; Fernandes, R.; Cataldo, R.; Agostini, L.; Sousa, L.; Marcon, C. “Multicore Parallelism Exploration Targeting 3D-HEVC Intra-Frame Prediction”, *IEEE Design Test*, vol. 37–3, 2020, pp. 15–21.

[54] Sanchez, G.; Saldanha, M.; Fernandes, R.; Cataldo, R.; Agostini, L.; Marcon, C. “3D-HEVC Bipartition Modes Encoder and Decoder Design Targeting High-Resolution Videos”, *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67–2, 2020, pp. 415–427.

[55] Sepúlveda, J.; Fernandes, R.; Marcon, C.; Florez, D.; Sigl, G. “A Security-Aware Routing Implementation for Dynamic Data Protection in Zone-Based MPSoC”. In: Symposium on Integrated Circuits and Systems Design (SBCCI), 2017, pp. 59–64.

[56] Shannon, C. E. “A Mathematical Theory of Communication”, *The Bell System Technical Journal*, vol. 27–3, Jul 1948, pp. 379–423.

[57] Sharman, K.; Sühring, K. “JCTVC-AF1100 - Common Test Conditions for HM Video Coding Experiments”. Source: https://jvet.hhi.fraunhofer.de/, Mar 2019.

[58] Shen, L.; Zhang, Z.; An, P. “Fast CU Size Decision and Mode Decision Algorithm for HEVC Intra Coding”, *IEEE Transactions on Consumer Electronics*, vol. 59–1, Feb 2013, pp. 207–213.

[59] Sullivan, G.; Wiegand, T. “Rate-Distortion Optimization for Video Compression”, *IEEE Signal Processing Magazine*, vol. 15–6, Nov 1998, pp. 74–90.

[60] Sullivan, G. J.; Ohm, J.-R.; Han, W.-J.; Wiegand, T. “Overview of the High Efficiency Video Coding (HEVC) Standard”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22–12, Dec 2012, pp. 1649–1668.

[61] Sze, V.; Budagavi, M.; Sullivan, G.-J. “High Efficiency Video Coding (HEVC): Algorithms and Architectures”. New York: Springer Publishing Company, 2014, 375p.

[62] Tech, G.; Chen, Y.; Muller, K.; Ohm, J.-R.; Vetro, A.; Wang, Y.-K. “Overview of the Multiview and 3D Extensions of High Efficiency Video Coding”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26–1, Jan 2015, pp. 1–1.

[63] Telecommunication Standardization Sector of ITU. “Recommendation H.120, Codecs for Videoconferencing Using Primary Digital Group Transmission”, Technical Report, International Communication Union (ITU-T), 1993, 66p.

[64] Telecommunication Standardization Sector of ITU. “Recommendation H.261, Video Codec for Audiovisual Services at p x 64 kbit/s”, Technical Report, International Communication Union (ITU-T), 1993, 29p.

[65] Telecommunication Standardization Sector of ITU. “Recommendation ITU-T H.265, High Efficiency Video Coding”, Technical Report, International Communication Union (ITU-T), 2016, 664p.

[66] Topiwala, P.; Dai, W.; Krishnan, M.; Abbas, A.; Doshi, S.; Newman, D. “Performance Comparison of AV1, HEVC, and JVET Video Codecs on 360 (Spherical) Video”. In: Applications of Digital Image Processing (ADIP), 2017, pp. 6.

[67] Uhrina, M.; Bienik, J.; Vaculik, M. “Coding Efficiency of HEVC/H.265 and VP9 Compression Standards for High Resolutions”. In: International Conference Radioelektronika (RADIOELEKTRONIKA), 2016, pp. 419–423.

[68] Urban, J. “How Chroma Subsampling Works”. Source: http://blog.biamp.com/how-chroma-subsampling-works, Sep 2018.

[69] Watson, A. B. “Image Compression Using the Discrete Cosine Transform”, *Mathematica Journal*, vol. 4–1, Aug 1994, pp. 17.

[70] Wiegand, T.; Schwarz, H. “Source Coding: Part I of Fundamentals of Source and Video Coding”. Boston: Now Publishers Inc, 2011, 222p.

[71] Wiegand, T.; Schwarz, H. “Video Coding: Part II of Fundamentals of Source and Video Coding”, *Foundations and Trends in Signal Processing*, vol. 10–1–3, Dec 2016, pp. 1–346.

[72] Wiegand, T.; Sullivan, G.; Bjontegaard, G.; Luthra, A. “Overview of the H.264/AVC Video Coding Standard”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13–7, Jul 2003, pp. 560–576.

[73] Winken, M.; Helle, P.; Marpe, D.; Schwarz, H.; Wiegand, T. “Transform Coding in the HEVC Test Model”. In: IEEE International Conference on Image Processing (ICIP), 2011, pp. 3693–3696.

[74] Winkler, S. “Digital Video Quality: Vision Models and Metrics”. Chichester: John Wiley & Sons, 2005, 192p.

[75] Xiph.Org Foundation. “Xiph.org Video Test Media”. Source: https://media.xiph.org/video/derf/, Aug 2020.

[76] Yang, H.; Shen, L.; Dong, X.; Ding, Q.; An, P.; Jiang, G. “Low-Complexity CTU Partition Structure Decision and Fast Intra Mode Decision for Versatile Video Coding”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30–6, 2020, pp. 1668–1682.

[77] Yoon, S.; Jameson, A. “Lower-Upper Symmetric-Gauss-Seidel Method for the Euler and Navier-Stokes Equations”, *AIAA Journal*, vol. 26–9, 1988, pp. 1025–1026.

[78] Zhao, L.; Zhang, L.; Ma, S.; Zhao, D. “Fast Mode Decision Algorithm for Intra Prediction in HEVC”. In: Visual Communications and Image Processing (VCIP), 2011, pp. 1–4.

[79] Zhao, L.; Zhou, K.; Guo, J.; Wang, S.; Lin, T. “A Universal String Matching Approach to Screen Content Coding”, *IEEE Transactions on Multimedia*, vol. 20–4, Apr 2018, pp. 796–809.

[80] Zhu, S.; Zeng, B. “Constrained Quantization in the Transform Domain with Applications in Arbitrarily-Shaped Object Coding”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20–11, Nov 2010, pp. 1385–1394.

[81] Zhu, W.; Yang, K. H.; Beacken, M. J. “CIF-to-QCIF Video Bitstream Down-Conversion in the DCT Domain”, *Bell Labs Technical Journal*, vol. 3–3, Jul 1998, pp. 21–29.

APPENDIX A – PUBLICATION LIST

Table A.1 – Published articles and papers on video coding.

Title: 3D-HEVC Bipartition Modes Encoder and Decoder Design Targeting High-Resolution Videos **Authors:** G. Sanchez, M. Saldanha; R. Fernandes, R. Cataldo, L. Agostini, C. Marcon **Journal:** IEEE Transactions on Circuits and Systems **Qualis:** A2, **Year:** 2020
Title: Multicore Parallelism Exploration Targeting 3D-HEVC Intra-Frame Prediction **Authors:** G. Sanchez, R. Fernandes, R. Cataldo, L. Agostini, L. Sousa, C. Marcon **Journal:** IEEE Design & Test **Qualis:** A1, **Year:** 2020
Title: Efficient HEVC intra-frame prediction using curved angular modes **Authors:** R. Fernandes, G. Sanchez, R. Cataldo, T. Webber, C. Marcon **Journal:** Electronics Letters **Qualis:** A3, **Year:** 2018
Title: Least-Squares Approximation Surfaces for High Quality Intra-Frame Prediction in Future Video Standards **Authors:** R. Fernandes, G. Sanchez, R. Cataldo, L. Agostini, C. Marcon **Conference:** International Conference on Electronics Circuits and Systems (ICECS) **Qualis:** B1, **Year:** 2018
Title: Low Area Reconfigurable Architecture for 3D-HEVC DMMs Decoder Targeting 1080p Videos **Authors:** G. Sanchez, R. Fernandes, R. Cataldo, L. Agostini, C. Marcon **Conference:** International Conference on Electronics Circuits and Systems (ICECS) **Qualis:** B1, **Year:** 2018
Title: DCDM-Intra: Dynamically Configurable 3d-Hevc Depth Maps Intra-Frame Prediction Algorithm **Authors:** G. Sanchez, R. Fernandes, L. Agostini, C. Marcon **Conference:** International Conference on Image Processing (ICIP) **Qualis:** A1, **Year:** 2018

Table A.2 – Published articles and papers on other areas.

Title: Subutai: Speeding up Legacy Parallel Applications through Data Synchronization **Authors:** R. Cataldo, R. Fernandes, G. Sanchez, J. Silveira, K. Martin, J-P. Diguet, C. Marcon, J. Sepúlveda **Journal:** IEEE Transactions on Parallel and Distributed Systems **Qualis:** A1, **Year:** 2020
Title: Using Smart Routing for Secure and Dependable NoC-based MPSoCs **Authors:** R. Fernandes, C. Marcon, R. Cataldo, J. Sepúlveda **Journal:** IEEE Transactions on Networking **Qualis:** A1, **Year:** 2020
Title: Subutai: Distributed Synchronization Primitives in NoC Interfaces for Legacy Parallel-Applications **Authors:** R. Cataldo, R. Fernandes, K. Martin, J. Sepulveda, A. Susin, C. Marcon, J-P. Diguet **Conference:** Design Automation Conference (DAC) **Qualis:** A1, **Year:** 2018
Title: A security-aware routing implementation for dynamic data protection in zone-based MPSoC **Authors:** J. Sepulveda, R. Fernandes, C. Marcon, D. Florez, G. Sigl **Conference:** Symposium on Integrated Circuits and Systems Design (SBCCI) **Qualis:** A4, **Year:** 2017

Table A.3 – Articles and papers on video coding under review.

Title: Using curved angular intra-frame prediction to improve video coding efficiency **Authors:** R. Fernandes, G. Sanchez, R. Cataldo, L. Agostini, C. Marcon **Journal:** Journal of Visual Communication and Image Representation **Qualis:** A1, **Year:** 2020

MARISTA

PUCRS

Pontifícia Universidade Católica do Rio Grande do Sul
Pró-Reitoria de Graduação
Av. Ipiranga, 6681 - Prédio 1 - 3º. andar
Porto Alegre - RS - Brasil
Fone: (51) 3320-3500 - Fax: (51) 3339-1564
E-mail: prograd@pucrs.br
Site: www.pucrs.br